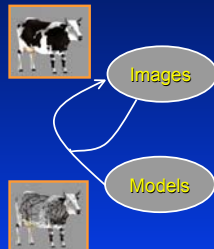


Geometry Graphics Pipeline

- Conventional Computer Graphics



Copyright © Manuel M. Oliveira

Rendering Images

Geometric model



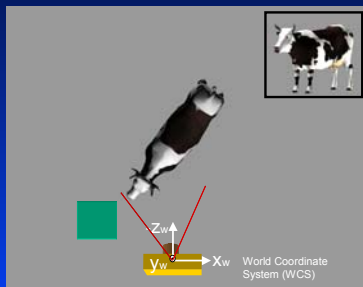
rendering
algorithm

Picture



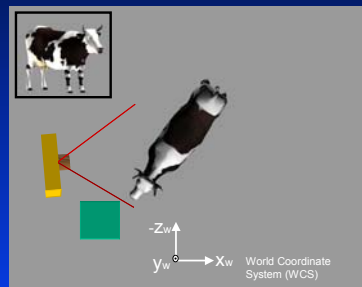
Copyright © Manuel M. Oliveira

The Visualization Problem



Copyright © Manuel M. Oliveira

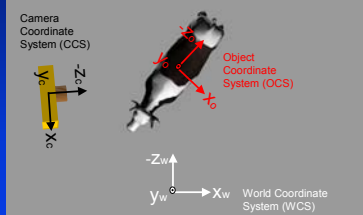
The Visualization Problem



Copyright © Manuel M. Oliveira

The Visualization Problem

Need to change coordinate systems!



Copyright © Manuel M. Oliveira

Geometric Models

- Polygonal Meshes (triangles - most common)
- Curved surfaces (e.g., Bezier)
- Implicit (e.g., sphere equation):

$$(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 - r^2 = 0$$

Copyright © Manuel M. Oliveira

Triangular Meshes

Why Triangles?

– Planar polygon

- Simplifies hardware design and implementation
- Bilinear interpolation produces reasonable results for several attributes (e.g., color and depth)

– Curved surfaces can be arbitrarily approximated by polygonal representations



Copyright © Manuel M. Oliveira

Rendering Pipeline

Z-buffer and Gouraud shading



Copyright © Manuel M. Oliveira

Rendering Pipeline



- Map from WCS to CCS
- Culling (vf, bf)
- Lighting

Copyright © Manuel M. Oliveira

Rendering Pipeline

Z-buffer and Gouraud shading



- Map from WCS to CCS
- Culling (vf, bf)
- Lighting
- Map from CCS to Screen CS
- Clipping
- Persp. division

Copyright © Manuel M. Oliveira

Rendering Pipeline



- Map from WCS to CCS
- Culling (vf, bf)
- Lighting
- Map from CCS to Screen CS
- Clipping
- Persp. division
- Color
- Visibility

Copyright © Manuel M. Oliveira

Rendering Pipeline



- Map from WCS to CCS
- Culling (vf, bf)
- Lighting
- Map from CCS to Screen CS
- Clipping
- Persp. division
- Color
- Visibility
- FB display

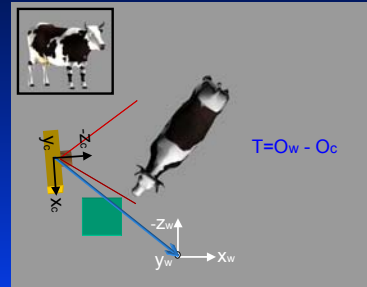
Copyright © Manuel M. Oliveira

Geometry Processing

- Map from WCS to CCS
- Culling
- Lighting

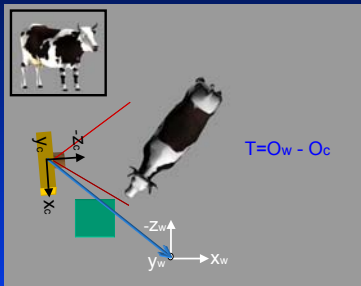
Copyright © Manuel M. Oliveira

Geometry Processing: Mapping from the WCS to the CCS



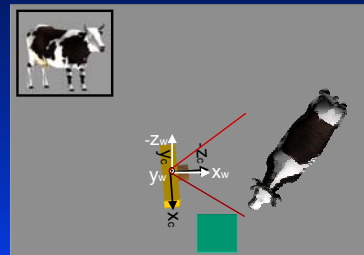
Copyright © Manuel M. Oliveira

Mapping from the WCS to the CCS: Equivalent Views: View 1



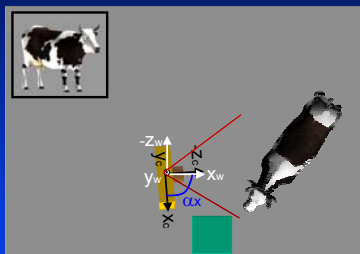
Copyright © Manuel M. Oliveira

Mapping from the WCS to the CCS: Equivalent Views: View 2



Copyright © Manuel M. Oliveira

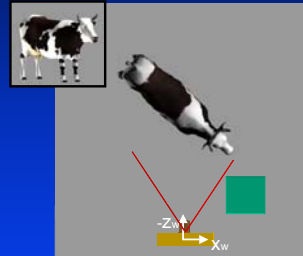
Mapping from the WCS to the CCS: Angle between Frames



Copyright © Manuel M. Oliveira

Mapping from the WCS to the CCS: Equivalent Views: View 3

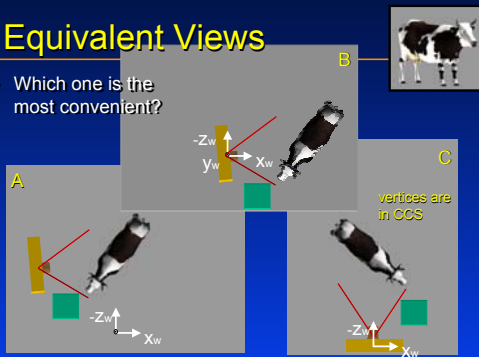
- Objects and camera rotated



Copyright © Manuel M. Oliveira

Equivalent Views

- Which one is the most convenient?



Copyright © Manuel M. Oliveira

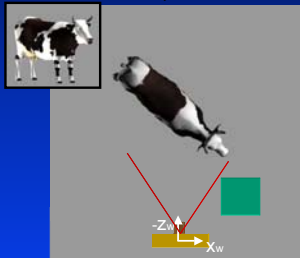
OpenGL Conventions

- Right-hand coordinate systems (world and camera)
- World and Camera coordinate systems aligned (camera at the origin)
- Translating and rotating the camera is equivalent to translating and rotating the scene (objects) in the opposite sense.

Copyright © Manuel M. Oliveira

Geometry Processing: Culling

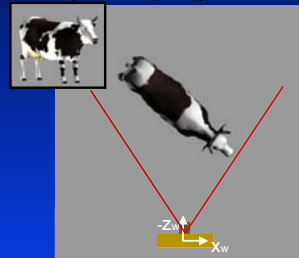
- Reduces the amount of primitives to be rendered



Copyright © Manuel M. Oliveira

Culling: View Frustum Culling

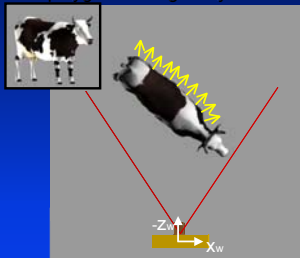
- Removes objects completely outside of the field of view



Copyright © Manuel M. Oliveira

Culling: Back Face Culling

- Removes polygons facing away from the viewer



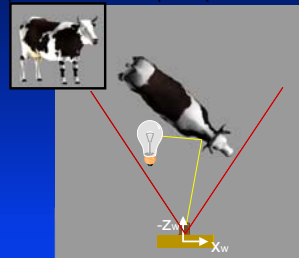
Copyright © Manuel M. Oliveira

Obs: Not all culling is done as geometry processing (normal cones). Some is done after projection and mapping to viewport.

Geometry Processing: Lighting

- Computed in camera space (or in an isomorphic space)

Why?



Copyright © Manuel M. Oliveira