Computer Graphics

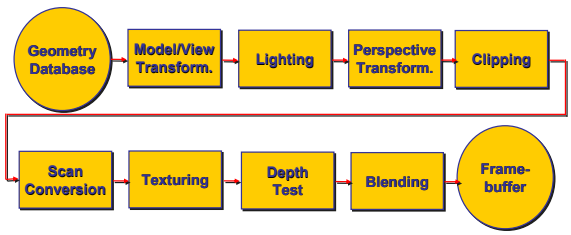
Rendering Pipeline

University of British Columbia

Chapter 2.5

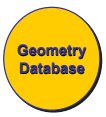
Rendering Pipeline

The Rendering Pipeline



University of British Columbia

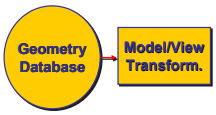
Geometry Database



- Geometry database:
  - Application-specific data structure for holding geometric information
  - Depends on specific needs of application
    - Independent triangles, connectivity information etc.

University of British Columbia

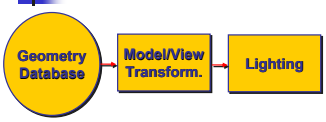
Model/View Transformation



- Modeling transformation:
  - Map all geometric objects from a local coordinate system into a world coordinate system
- Viewing transformation:
  - Map all geometry from world coordinates into camera coordinates

University of British Columbia

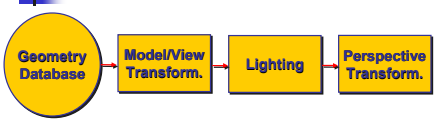
Lighting



- Lighting:
  - Compute the brightness of every point based on its material properties (e.g. Lambertian diffuse) and the light position(s)
  - Computation is performed *per-vertex*

University of British Columbia

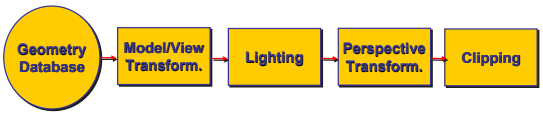
Perspective Transformation



- Perspective transformation
  - Projecting the geometry onto the image plane
  - Projective transformations and model/view transformations can all be expressed with 4x4 matrix operations

University of British Columbia

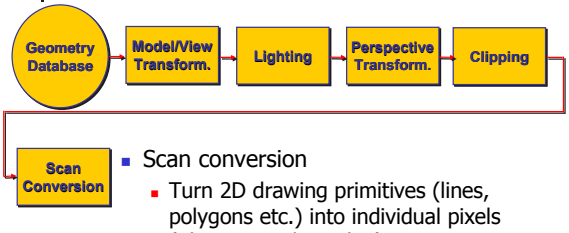
Clipping



- Clipping
  - Removal of parts of the geometry that fall outside the visible screen or window region
  - May require *re-tessellation* of geometry

University of British Columbia

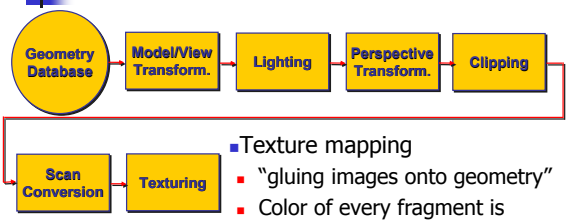
Scan Conversion



- Scan conversion
  - Turn 2D drawing primitives (lines, polygons etc.) into individual pixels (*discretizing/sampling*)
  - Interpolate color across primitive
  - Generate discrete *fragments*

University of British Columbia

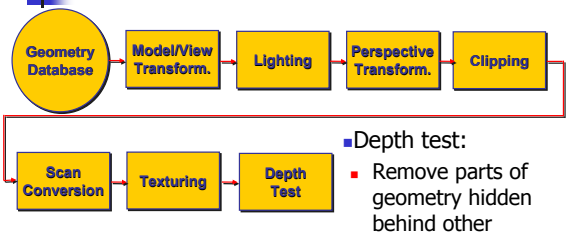
Texture Mapping



- Texture mapping
  - "gluing images onto geometry"
  - Color of every fragment is altered by looking up a new color value from an image

University of British Columbia

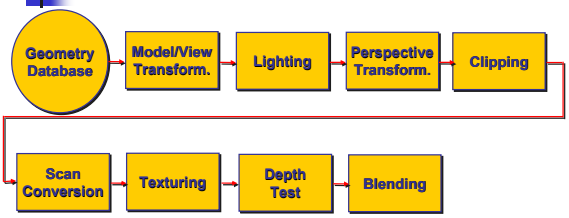
Depth Test



- Depth test:
  - Remove parts of geometry hidden behind other geometry
  - Perform on every individual fragment
    - other approaches (later)

University of British Columbia

Blending



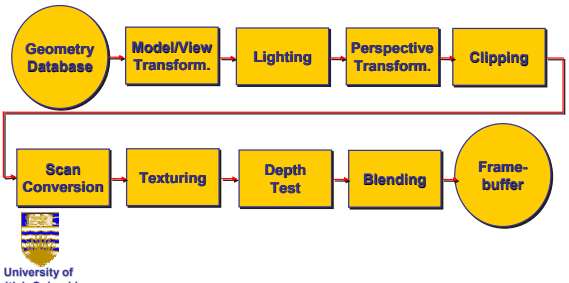
University of British Columbia

Blending

- Blending:
  - Final image: write fragments to pixels
  - Draw from farthest to nearest
  - No blending – replace previous color
  - Blending: combine new & old values with some arithmetic operations
  - *Framebuffer :* video memory on graphics board that holds resulting image & used to display it

University of British Columbia

The Rendering Pipeline



University of British Columbia

Discussion

- Advantages of a pipeline structure
  - Logical separation of the different components, modularity
  - Easy to parallelize:
    - Earlier stages can already work on new data while later stages still work with previous data
    - Similar to pipelining in modern CPUs
    - But much more aggressive parallelization possible (special purpose hardware!)
    - Important for hardware implementations!
  - Only local knowledge of the scene is necessary

University of British Columbia

Discussion

- Disadvantages:
  - Limited flexibility
  - Some algorithms would require different ordering of pipeline stages
    - Hard to achieve while still preserving compatibility
  - Only local knowledge of scene is available
    - Shadows
    - Global illumination

University of British Columbia