Lecture 02. Graphics Pipeline

3D Graphics Rendering Pipeline

3D Scene: objects, lights, camera.

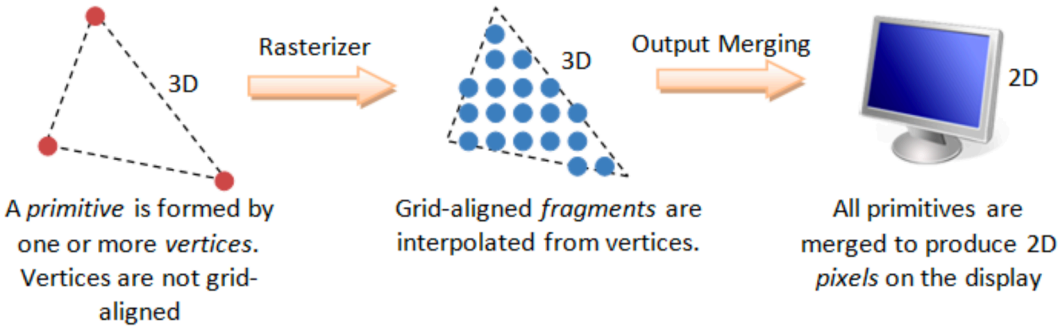
Three main steps of the pipeline

- Application;
- Camera & model transformation (in GPU);
- Rasterization (in GPU)（光栅化）: sampling and dicretization.

Primitives

- The inputs to the Graphics Rendering Pipeline are geometrice primitives (such as triangle, point, line or quad), which is formed by *one or more* vertices.
- Each vertex is associated with its attributes such as the position, color (for vertices) (in Red-Green-Blue-Alpha, RGB+透明度), normal（法向量） and texture（材质）.

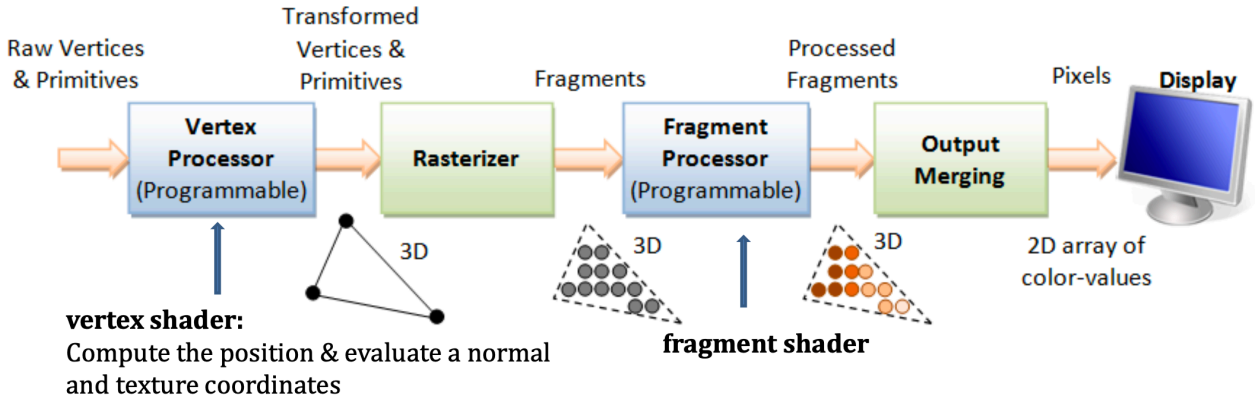
Pixels vs Fragment



Vertex, Primitives, Fragment and Pixel

- Pixels refers to the dots on the display. A pixel is 2D, with a (x, y) position and a RGB color value;
- A fragment is 3D, with a (x, y, z) position. The (x, y) are aligned with the 2D pixel-grid. The z value (not grid-aligned) denotes its depth.
- Fragments are produced via *interpolation* of the vertices. Hence, a fragment has all the vertex's attributes such as color, fragment-normal and texture coordinates.

3D Graphics Rendering Pipeline (Hardware Architecture)



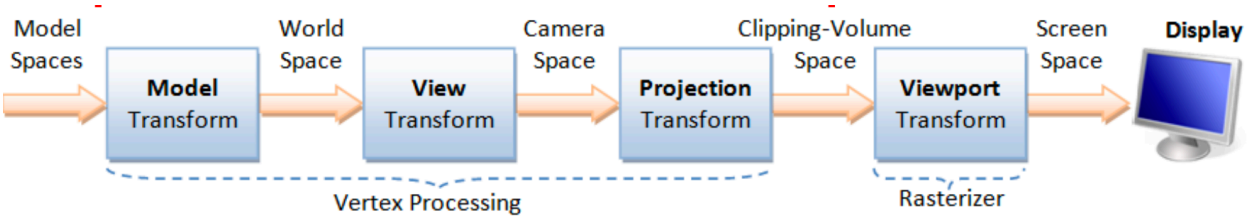
3D Graphics Rendering Pipeline: Output of one stage is fed as input of the next stage. A vertex has attributes such as (x, y, z) position, color (RGB or RGBA), vertex-normal (n_x, n_y, n_z), and texture. A primitive is made up of one or more vertices. The rasterizer raster-scans each primitive to produce a set of grid-aligned fragments, by interpolating the vertices.

- **Vertex Processing**: process and transform individual vertices and normals.
- **Rasterization**: convert each primitive (connected vertices) into a set of fragments. A fragment can be treated as a pixel in 3D spaces, which is aligned with the pixel grid, with attributes such as position, color, normal and texture.
- **Fragment Processing**: process individual fragments.
- **Output Merging**: combine the fragments of all primitives (in 3D space) into 2D color-pixel display.

Coordinate System

- Right hand coordinate system (RHS);
- Object coordinates (a.k.a. local, model coordinates);
- World coordinates (absolute coordinates);
- Viewing coordinates;
- Also clip, normalized device, and window coordinates.

Coordinate Transform Pipeline



Coordinates Transform Pipeline

Visibility

Painter's Algorithm: paint from back to front, overwrite in the framebuffer.

Z-Buffer Algorithm

Initialize depth buffer to ∞

During rasterization:

```
for (each triangle T)
  for (each sample (x,y,z) in T)
    if (z < zbuffer[x,y])          // closest sample so far
      framebuffer[x,y] = rgb;      // update color
      zbuffer[x,y]     = z;        // update z
    else
      ;     // do nothing, this sample is not closest
```