Realtime Rendering

CS 481/681 Computer Graphics Rendering

University of Alaska Fairbanks

Overview

- Reyes Architecture
- Shade Trees

Reyes Architecture

- Cook, Carpenter, Catmull 1987
- Pixar Renderman
- An Architecture
- Micropolygons
- Visibility
- Programmable Shaders
- Motivation: Speed
  - Rendering a 2 hour movie in one year
  - 3 minutes per frame

Reyes Flowchart

Shade Trees

- Cook 1984

Questions

- What are advantages / disadvantages?
- What are micropolygons?
- Why use shaders?

Comparison

MODEL
↓
read model
↓
bound
↓
on screen? —N→ cull
↓ Y
split ←N— diceable?
↓ Y
dice
↓
TEXTURES → shade
↓
sample
↓
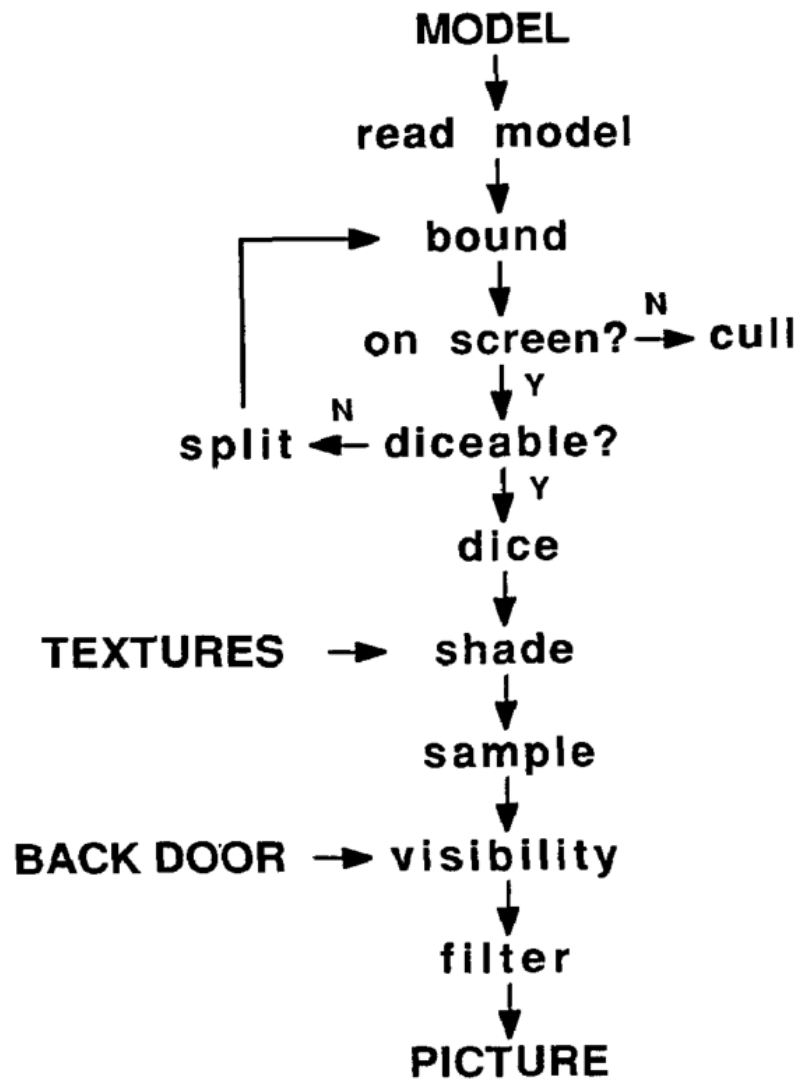BACK DOOR → visibility
↓
filter
↓
PICTURE

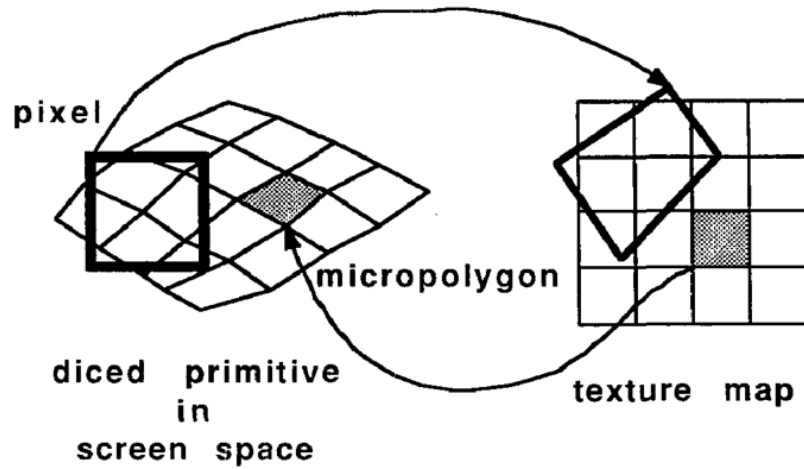*Figure 1. Overview of the algorithm.*

Figure 1: Reyes Architecture

Figure 2. With CATs, micropolygons map exactly to texture map pixels. With the inverse pixel method, pixels map to quadrilateral areas of texture that require filtering.
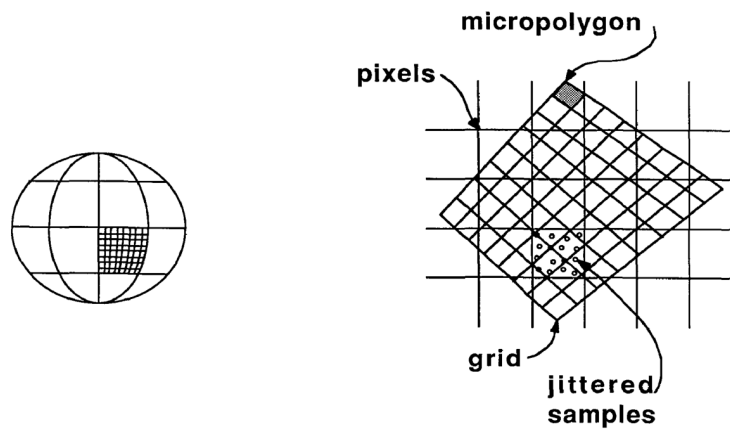
Figure 2: Reyes Texture Maps

Figure 4a. A sphere is split into patches, and one of the patches is diced into a 8×8 grid of micropolygons.

Figure 4b. The micropolygons in the grid are transformed to screen space, where they are stochastically sampled.

Figure 3: Reyes Micropolygons

```
Initialize the z buffer.
For each geometric primitive in the model,
        Read the primitive from the model file
        If the primitive can be bounded,
                Bound the primitive in eye space.
                If the primitive is completely outside of the hither-yon z range, cull it.
                If the primitive spans the ε plane and can be split,
                        Mark the primitive undiceable.
                Else
                        Convert the bounds to screen space.
                        If the bounds are completely outside the viewing frustum, cull the primitive.
        If the primitive can be diced,
                Dice the primitive into a grid of micropolygons.
                Compute normals and tangent vectors for the micropolygons in the grid.
                Shade the micropolygons in the grid.
                Break the grid into micropolygons.
                For each micropolygon,
                        Bound the micropolygon in eye space.
                        If the micropolygon is outside the hither-yon range, cull it.
                        Convert the micropolygon to screen space.
                        Bound the micropolygon in screen space.
                        For each sample point inside the screen space bound,
                                If the sample point is inside the micropolygon,
                                        Calculate the z of the micropolygon at the sample point by interpolation.
                                        If the z at the sample point is less than the z in the buffer,
                                                Replace the sample in the buffer with this sample.
        Else
                Split the primitive into other geometric primitives.
                Put the new primitives at the head of the unread portion of the model file.
Filter the visible sample hits to produce pixels.
Output the pixels.
```

Figure 3. Summary of the algorithm.
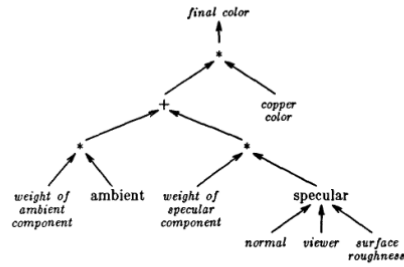
Figure 4: Reyes Algorithm
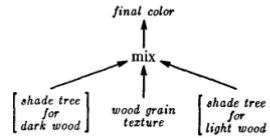
4

Figure 1a. Shade tree for copper.

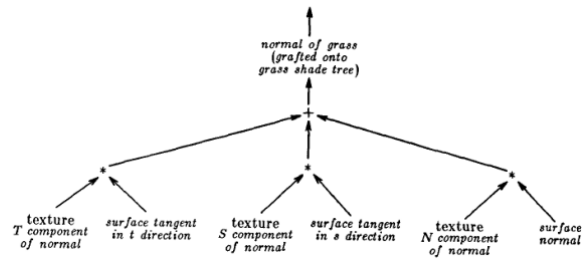Figure 1b. The mix node in a shade tree for wood.
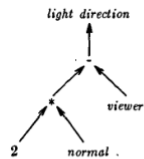
Figure 1c. Textured grass normal.
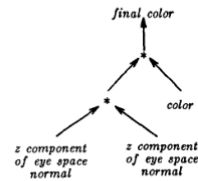
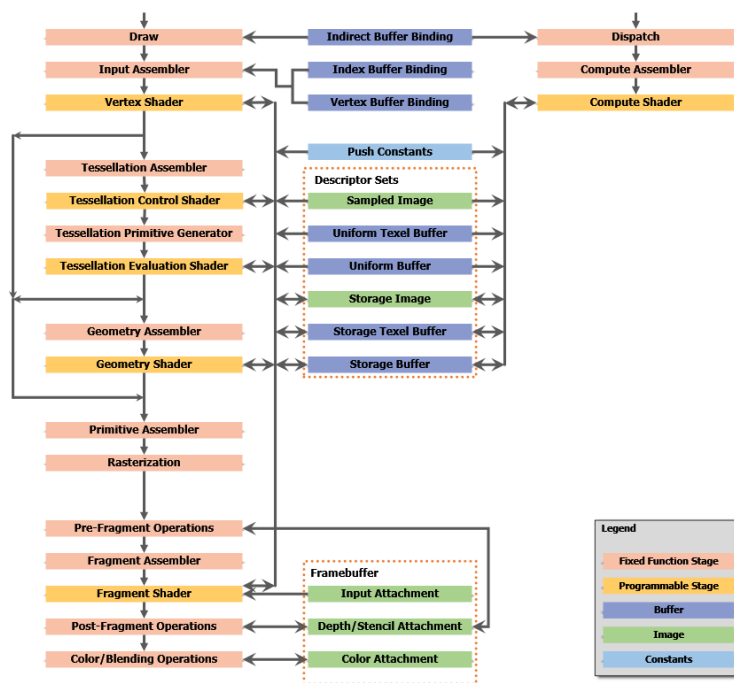Figure 1d. "Highlight at" branch of a light tree.

Figure 1e. Simple shade tree.

Figure 5: Shade Trees

Figure 6: Vulkan Pipeline