
CS380 Introduction to Computer Graphics

Homework #6

20160042 Inyong Koo

Describe the following terms with respect to computer graphics.

Interactions between light and materials can be classified into three groups: specular, diffuse, and translucent surface.

1. Geometry processing

Geometry processing works with vertices. The goals of the geometry processor are to determine which geometric objects can appear on the display and to assign shades or colors to the vertices of these objects. Geometry processing is composed of 4 process: projection, primitive assembly, clipping, and shading.

2. Primitive assembly

When vertices reach the rasterizer, they can no longer be processed individually and first must be assembled into primitives. **Primitive assembly** is the process that groups vertices into objects. It takes place before clipping.

3. Hidden-surface removal

Even though an object lies inside the view volume, it will not be visible if it is obscured by other objects. **Hidden-surface removal** is normally carried out as part of fragment processing, through algorithms based on on the three-dimensional spatial relationships among objects.

4. Perspective division

After clipping takes place, the remaining vertices are still in four-dimensional homogenous coordinates. **Perspective division** converts them to three-dimensional representation in normalized device coordinates.

5. Scan conversion

Scan conversion, or **rasterization**, is a process that determines which fragments or pixels to visualize objects.

6. Fragment processing

Fragment processing is the process of taking the image from the frame buffer and displaying it on a monitor. The hidden-surface removal, interpolation of per-vertex colors and texture coordinates takes place in fragment processing. Texture parameters determine how to combine texture colors and fragment colors to determine final colors in the color buffer.

7. Aliasing

Quality of display causes numerous problems, such as the jaggedness associated with images on raster displays. Reducing such jaggedness is called **aliasing**.

8. Normalized device coordinates

Clipping is done before the perspective division that is necessary if the w component of a clipped vertex is not equal to 1. The portion of all primitives that can possibly be displayed lie within the cube $w \geq x \geq -w, w \geq z \geq -w, w \geq y \geq -w$. This coordinate system is called **normalized device coordinates** because it depends on neither the original application units nor the particulars on the display device.

9. Clipping

Clipping is the process of determining which primitives, or parts of primitives, fit within the clipping or view volume defined by the application program.

10. Winding test

Winding test considers the polygon as a knot being wrapped around a point or a line. It is used in fill algorithm to determine whether a point is inside or outside of the polygon.

11. Object-space approach

There are two approaches for hidden-surface-removal algorithms; object-space approach and image-space approach. In **object-space approach**, we consider the objects pairwise, and determine if both are completely visible, partially obscure each other, or one completely obscure the other. When there are k objects, we should compare an object to the rest ($k-1$ objects), thus the time complexity for this algorithm is $O(k^2)$.

12. Image-space approach

Image-space approach follows our viewing and ray-casting model, considering a ray that leaves the center of projection and passes through a pixel. The intersection of this ray with each of the planes determined by our k objects, determine for which object is the closest to the camera.

13. Back-face culling

In OpenGL, we often choose to render only front-facing polygons. The elimination of all back-facing polygons, or **back-face culling**, can be done once we determine whether each polygon is fronting back or forward. If θ is the angle between the normal and the viewer, then the polygon is facing forward if and only if $-90^\circ \leq \theta \leq 90^\circ$ or, equivalently, $\cos \theta \geq 0$. The second condition is much easier to test, because we can use dot product $\mathbf{n} \cdot \mathbf{v} \geq 0$.

14. z-buffer algorithm

The **z-buffer algorithm** is the most widely used hidden-surface-removal algorithm. It is based on image-space approach. We keep depth information with each fragment to determine which part should be removed. The depth information is stored and updated during fragment processing.

15. Painter's algorithm

Although image-space methods are dominant in hardware due to efficiency and ease of implementation of the z-buffer algorithm, often object-space methods are used within the application to lower the polygon count. Depth sort, a variant of the **painter's algorithm**, is a direct implementation of the object-space approach to hidden-surface removal. We just render the objects in back-to-front order.

16. Antialiasing by area averaging

We can get a smoother appearing image if we shade each pixel by the percentage of the area that crosses it. This technique is known as **antialiasing by area averaging**.

17. Spatial-domain aliasing

Spatial-domain aliasing is the spatial approximation of object to limited resolution, with effort to make smooth image.

18. Time-domain aliasing

When we generate sequences of images, such as for animations, we must be also concerned with **time-domain aliasing** in order to smoothen flickering.

19. Gamma correction

Due to hardware differences of CRTs, Two monitors may generate different brightnesses for the same values in the frame buffer. We can have a lookup table in the display whose values can be adjusted for the particular characteristics of the monitor. This is called **gamma correction**.

20. Halftoning

A color buffer can be specified by its spatial resolution and by its precision. In real world, we cannot satisfy these separate fixed numbers. In such circumstances, we can trade spatial resolution for grayscale or color precision. **Halftoning** techniques in the printing industry use photographic means to simulate gray levels by creating patterns of black dots of varying size.