11/10/2020

Buffers and Texture Mapping

Relation to shading - using images rather than light sources to determine
the intensity of vertices, followed by interpolation.

Note that in shading and texture mapping we need to specify the normal to surface elements and details
Too many normal vectors ? ➔ Bump mapping

Texture Mapping

How can I render a complex structure

1) Geometry Primitive (Polygons)
   Micky Mouse? – Many Polygons (triangles) from wire frame
   Human face.
   Orange – dull or many polygons might require extremely large number of polygons
2) Texture mapping - map an image to geometry (e.g. polygons)
3) Curves and surfaces – OGL approximates curves and surfaces – renders (plot)

Texture mapping requires several transformation

We need to instruct OGL to perform these transformations by mapping image pixels to vertices.

Each vertex inherits the pixel brightness. The picture is interpolated by mapping
internal pixels to internal points in the polygon
It is not a one to one mapping

Texture coordinates i.e., image coordinates, identify pixels

Specified using coordinates s and t $0 <= s, t <= 1$

You supply s, t

OGL has to map into V = (x, y, z, 1) (i.e., from the image to the vertex.)

Has to find these functions
x = x(s, t), y=y(s, t), z=z(s, t) 25-14 but it might produce holes.
The inverse mapping has the advantage of avoiding holes.

For every vertex V, OGL has to determine the inverse function s=s(x, y, z) t = t(x, y, z) 25-15

25-17 parametric representation of cylinder
25-18 sphere, note that this is a surface ➔ use glPoints? glLines? Better use patches -- quads

Buffers

Read an image from a file into a frame-buffer (FB) ➔ render to screen using glFlush(), glSwap()
Write an image from a FB to a file
Copy image slices from the frame buffer onto the from buffer

An $m \times n$ image has pixels, it is addressed through integer coordinates (e.g., i, j)
K is the number of bits per pixels.

Monitor - 8 bits/pixel (GIMP, Photo-Shop have internal representation of 16 b/p converted into 8 b/p on display)

24-3 k bits / pixel ➔ K binary plans
24-4 OGL frames
Front, Back, Auxiliary and Overlay can function as frame buffers

The Depth (z - buffer) contains information about distance of vertices from the camera
Can be used for hidden surface removal

The Stencil buffer is a one-bit buffer (masking, fonts, shape representation)

The frame buffer is a part of the Main memory or on a card
mapped to the memory space

23-6 ➔ the source can be a file, or the frame-buffer
The destination can be a file or the frame-buffer

Writing pixels 23-7 d' (written value) is a function of s, and d* (d* is the previous value of the pixel).

Output pixel is a function of the input pixel and previous output (value of the pixel).
16 modes of writing 23-8. Replace is like regular write.

In the XOR mode d' = (s XOR d*)
23-9 Use swap with no temporary via XOR you can copy from FB to FB without temporary memory.
XOR is one of the basis basic operations of encryption.

Raster position – 23-11
The next primitive will be drawn from x, y, z.

23-12 buffer selection.

Nuisance – need to take some computer parameters into account
May require Packing un-packing
From bits and bytes to words.
From words to bytes, to bits
23-13 14, 15, Stencil skipped
23-14 relates to glColor using glrasterpositing (raher than gal vertex).

Pixels 8 b/p is the common.
RGB chromatic (color) images
Luminance (gray-scale) relates to our perception of gray level
Can be approximated by setting the intensities R=G=B

23-17

- Three functions
  - Draw pixels: processor memory to frame buffer
  - Read pixels: frame buffer to processor memory
  - Copy pixels: frame buffer to frame buffer

Draw (23-18) From an image stored in an array (2D or 1D) to FB

Slides 23:21 – 25 show snippets of the “original” ppmread.c by Angel.
Available on TRACS and in the assignment but might require a minor modification for some of the images.

We use PPM / PGM (PGMA) very primitive, raw a version that can be opened by text editors.

My camera must have raw format

16-Millions of colors