

11/12/2020

There will be a quiz next Thursday.

Fixing n - for each vertex P find the plane where p is located

Find 3 points on the plane perform cross product $\rightarrow n$

More than one plane \rightarrow average all the normal

Transformations on vertices (objects) might transform light sources.

In some case (you want only to translate light source) use push and pop

Along with the translation.

Assignment S3 is due on 11/13.

How to set the light source at infinite with an angle of Theta.

According to the Phong model you define a point light source using a

Finite location (defined by a point $(a, b, c, 1)$)

Clearly the vector $(a, b, c, 0)$ points to the light source this is I in fig cg18-13.

Infinite location (defined by a vector $(a, b, c, 0)$)

Clearly the vector $(a, b, c, 0)$ points to the light source this is I in fig cg18-13.

"Placing" the sun at $(0, 1, 0, 0)$ 90 degrees – above my head

We need to find the angle between I for now we call it (Betha) and N for the model

You define N .

Betha is \cos^{-1} of $I \cdot N$ slide 18-15

$\cos(\phi) = V \cdot R$ R is

GL-SPOT Place the spot at finite location and define any $(0,0,0,1)$ point as the direction

Correction:

You can only use GL functions and CANNOT use any GLU or GLUT functions for the objects e.g., sphere, cube (you can use the sphere snippet posted on TRACS).

Use Glu for the gluLookat and Glu perspective and GLUT for interacting with you program.

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

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 front 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 $d' = (s \text{ XOR } d^*)$

23-9 Use swap with no temporary via XOR you can copy from FB to FB without temporary memory.
XOR is one of the 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 (rather than glVertex).

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.

11-12-2020

Image formats 24-19

For glDrawPixels Open GL is expecting a list of pixels Placed in an array to be placed on the frame buffer an array of pixels

For glReadPixels OpenGL returns an array of pixels from the FB.

A monochrome pixel is 8-bit unsigned value from 0 to 255

Denotes a gray level.

A chromatic pixel have RGB (3 values of grayscale) RGBK (K opacity)

You can use unsigned chars maybe in binary or in ASCII

You can read into integers write with casting from integers

The pixel value is assumed to denote intensity (physical value) of the pixel.

Might denote brightness (perceptual)

The dictionary definition of Shadow is:

“A color, especially with regard to how light or dark it is or as distinguished from one nearly like it.”

24-19

We need to provide the pixels for glDrawPixels.

jpeg has a relatively complex headers. You cannot readily access individual image pixels.

PGM, PPM, denotes a format with minimal header and pixels arrange in raw fashion (no processing)

In C scanf() printf() to read/write ASCII fscanf() from a file
Putc() getc() to read write.

In C++ '<<' '>>' for ASCII specific library for binary

	ASCII (<<)		Binary			
	Header		Header			
B/W (1 bit/pixel)	P1		P4			
Monochrome	P2		P5	PGM PGMA		
Color	P3		P6	PPM		
	Formatted		Unformatted			

Slide CG24-20 PPM ASCII format

The program ppmread.c by Angel expects this format

e.g.,
P3
comment
256 256 255
Pixels

But some images might have a slightly different arrangement

e.g., P3 256 256 255
comment

Check the one on the assignments.
Change it or edit the image

Used for Raw images

Recognized by Linux (use same “tools” as for jpeg)
Windows does not.

Photoshop can display, GIMP (free) can easily display

Image-magic can display
PGMA and many other formats are archived in an FSU site

The display programs are generally scaling the gray levels
Not always

In FEEP the max is defined to be 15.
So the intensity of $[0, 1]$ is mapped into $[0, 15]$ by GIMP.

If it is not mapped the entire image will look dark you may need to
Scale from $[0-15]$ to $[0-255]$

Slides 24-21
24-25 snippets of ppmread
Change it for PGM files

21 reads the header

To shadow a sphere place a circle somewhere where you would expect the shadow effect
And color it in some shade of black

To shadow a cube place a polygon (quad)

One of your light sources is ambient
You should specify all the 9 intensity parameters for every light source

Position a polygon (of your choice)
With a gray level (of your choice)
In a place of (your choice)

Try to make it realistic trial and error

My camera must have raw format

16-Millions of colors