11/17/2020

There will be a quiz on Tuesday 11/24 during class.
It will include material covered after the midterm and before
The Texture mapping. Mainly related to E1 and shading
Including S3 material understanding concepts and the assignments material.
Solution to S3 will be provided.

The display programs are generally scaling the gray levels of PGM/PPM files
Not always – you might need to scale the pixels when you read them into
The GL Frame buffer (ppmread.c includes scaling).

Intensity and color information.
Color
Most LCD screens can display RGB information at 8 bits/pixel

[0,...,255] intensity for the R, G, and the B components.

R=G=B ➔ 256 **intensities** of gray level.

YUV is used for some TV/Video formats.
Y represents intensity in a way that is closer to **Brightness.** That is, the perception of Intensity.

Going from RGB ➔ YUV through a transform (U, V relates to perception of the Blue and Red chromas)
We can use the Y for gray levels.

We pretend infinite gray levels from [0,…,1].
Mapping [0,...,1] to [0,...,255]

For assignment S4, you might need to map from [n,...,k] into [0,…,255]

For example, mapping the Feep image from [0,...,15] to [0,…,255]
Photoshop, GIMP, and Image-magic are automatically mapping during the display.

How many different colors can an LCD monitor display?

Resolution pixels / area (dot/inch)

R = 2^8, G=2^8, B=2^8 ➔ 2^{24} = 16*1024 ~= 16 million

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.c
You will have to change it for PGM files

Slides from ppmread.c

24:21, 24:22 reading the header etc.

24:23 reading the data using integers. One color component into a single integer
Into one dimensional array of n*m*3.

Malloc is a system call used in C in lieu of New(). In this case 3*[the size of unsigned int * n *m] integers.

May want to consider unsigned chars.

Scaling in 24:24

The Display callback uses glDrawPixels (24:25)

The first step in texture mapping ➔ place the image in FB.
Flush() ➔ displays the image.
OGL has a few built-in image processing functions. Such as, filtering, histogram, etc.

Several transformations are required to map an image from FB to geometry.
25: 12-16

Analog to digital conversion (ADC)
Assume that a film and the eye are analog (continuous and not discrete) representation of the image.

In ADC we sample the image pixel by pixel (discrete coordinate system)
Next, we assign a discrete value (gray level) for each pixel (quantization.)

ADC ➔ sampling and quantization (lossy representation).
DAC ➔ (lossy) reconstruction

Sampling relates to pixels/area

Example: monitor of 36" by 18". Assume, that it includes 1924*1080 pixels.
Implies density of pixels, pixels /area (resolution).
1924*1080 / (36*18) (DPI)

Lower resolution screen
640*480 / (36*18) (DPI)

Aliasing is an artifact of the transformation from analog to digital.
Visible (noise) if the sampling rate is not high enough low- resolution low density.

Lines looks like “stairs” or zigzag. To avoid aliasing we can average gray levels on the area
Or better yet use other filters

26-9
OGL expects that your image dimensions are 2^k by 2^l.
Otherwise use glu-scale-image to get the right dimensions.
Uses the ideal procedure that includes filtering.

For example for converting an image of 580*220 ➔ 512*256

26-10, 11
Mapping pixels to vertices. The vertex inherits the pixels gray level
(like shading). Between vertices OGL is using interpolation.

The Image is n by m considered as [s, t] [(0,…,1), (0,…,1)]

Assume that the image is 512x512

I want to map Pixel (Texel) 255x384 to glVertex(1, 2, 3)
This vertex will get the color of that pixel.

Set s0 ← 0.5 (256/512) set t0 ← 384/512 use glTexCoordinate(s0, t0)

Interpolation and boundary problems.
What should OGL do if for some reason it gets a value
Of s >1 or t>1

Can use different wrapping options 26:14

During the process of mapping (not one to one and onto), we may get more than one screen pixel mapped into one Texel, more than one Texel mapped into one pixel, and holes.

To mitigate – use the nearest Texel, or linear filtering. In S4 choose one of two (or both) modes filtering is preferred

Slides 24:6 and 24:7
Writing buffers

Consider
An instruction ‘LB dst (pointer)’ load byte using a register as a pointer
‘SB src (pointer)’ storing a byte using a pointer.

Example ‘lb R2(R3)’ use R3 as a pointer to memory and load the byte pointed to by
R3 into R2 R2 ← M[R3]
In ‘SB R2 (R3)’ M[R2] ← R3

Swapping a chunk of memory (n – bytes) from address K1 into memory staring from address K2.

R2 points to address K1, R4 points to address K2
LB R1 (R2)
LB R3 (R4)
SB R3(R2)
SB R1(R4)

Swap with XOR

y	x	x XOR y		
0	0	0		
0	1	1		
1	0	1		
1	1	0		

Curves in GL.