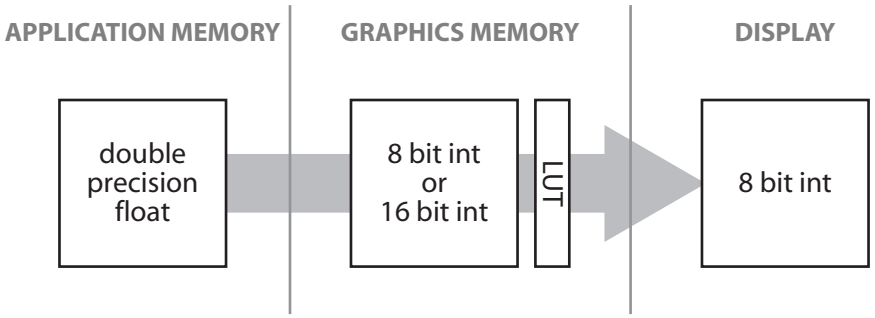
CS 465 Homework 1: Pixel formats

out: Friday 25 August 2006
due: Friday 1 September 2006

This homework is about the data that is used to represent images and exactly what it means. Follow tradition by giving data sizes in kilobytes, megabytes, or gigabytes, and data rates in kilobits, megabits, or gigabits per second. Don't forget that these units refer to multiples of 1024, not multiples of 1000. Round your answers to three significant digits—for example, 3.52 MB.

The following questions are based on this situation: An application (maybe it's the Ray I ray tracer from this class) computes a full-screen RGB color image and stores it in an array in main memory. The application draws the image on the screen by transferring it to a framebuffer in the graphics system, which causes it to be stored in the memory on the graphics card. The graphics hardware reads out this image and transmits it via a digital link to an LCD display.



The application stores each pixel in three double-precision floating point numbers, using the convention that pixel values are linearly related to display intensity, with the value 1.0 representing the maximum displayable intensity.

We'll consider three different formats in which the data might be stored in the framebuffer: (a) 8 bit integers with linear quantization, (b) 8 bit integers with nonlinear quantization at gamma 2.2, and (c) 16 bit integers with linear quantization. The graphics hardware has to convert these formats to the display's pixel format, and we'll assume this is done using a simple lookup table.

The display has a resolution of 1680x1050 pixels and is operating at 60 Hz. Its display gamma is 2.2, and it always takes 8 bits per color channel per pixel. The monitor has a maximum displayable intensity of $I_{\max}$ and is in an environment where viewing flare is 1%—that is, the pixel value zero results in an intensity of $0.01 I_{\max}$, and the maximum brightness is $1.01 I_{\max}$.

1. How much application memory is consumed by the image?
2. How much graphics memory is consumed by the framebuffer for each of the three framebuffer formats?
3. What data rate is flowing across the cable to the display?

4. What must the application do to convert its pixels into framebuffer pixels for each of the three framebuffer formats? Give a mathematical expression that you could type into the code that does this processing. Assume the application does not worry about viewing flare, preferring to believe that pixel value zero is exactly black. Assume the standard functions *round*, *pow*, *min*, and *max* are available.

5. What are the first 5 entries of the graphics card's lookup table for each of the three framebuffer formats? Round carefully to ensure that each pixel value gets mapped to the displayable value that comes closest to the intensity it's supposed to represent. Assume the graphics hardware does not correct for viewing flare either.

6. For each of the pixel formats, what are the relative differences in observed intensities corresponding to the framebuffer pixel values 0, 1, 2, 3, and 4? (Express your answer as four percentages: e.g. 0 to 1 is a 1% increase, 1 to 2 is a 3% increase, etc.) Which of these first four steps are visually noticeable, assuming a just noticeable difference of 2%?

7. Suppose the application believes the card is using pixel format (a) when it is really using (b). Would the displayed image look darker or lighter than it should?