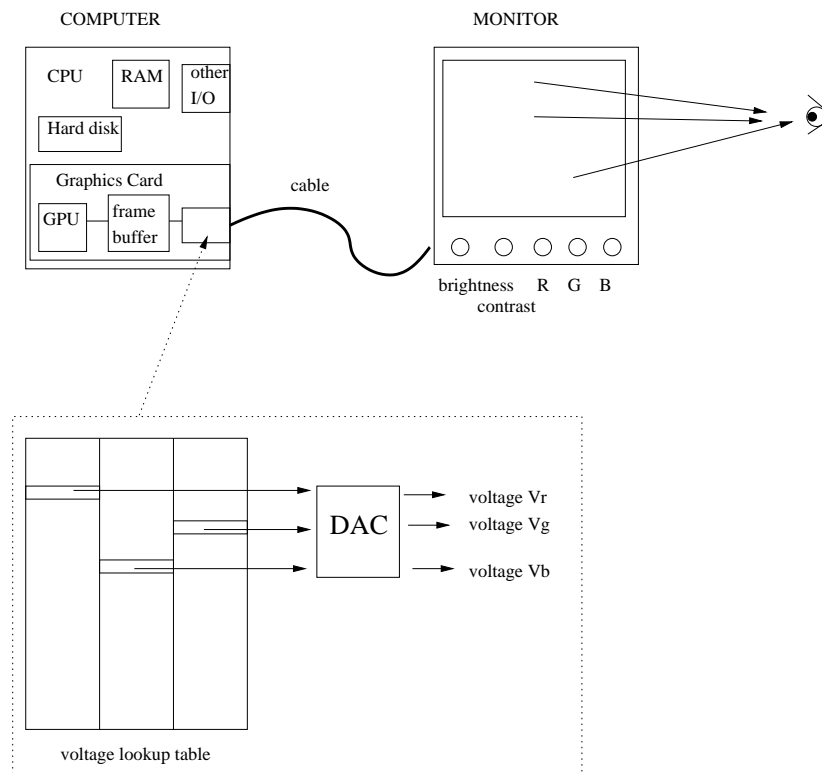


Displays

Today we are going to consider the basics of how images are displayed. The typical display is a monitor, but the same discussion applies to projection displays.

Consider the sketch below. On the left is shown the computer, which consists (among other things!) of the CPU/RAM/disk/etc and the graphics card. The graphics card contains a processing unit called the GPU (graphics processing unit) which is devoted to performing operations that are specific to graphics. The graphics card also contains the frame buffer, the depth buffer, texture buffers.



The graphics card is connected to the monitor via a cable. (In the case of a laptop, you would not see the cable.) The signal that is sent to the monitor along the cable specifies RGB levels for each monitor pixel, but these are typically *NOT* the same as the intensity values that are written in the frame buffer. Rather, the cable typically carries the RGB intensities as an analog signal (voltage level) rather than a digital signal (bits). e.g. Those of you who have plugged your laptop into a projector may know that you typically use a VGA cable (VGA is analog). CRT displays expect analog inputs. LCD can expect either analog or digital.

Where does the analog signal specified? Each RGB triplet in the frame buffer is used to index into a lookup table (a readable/writable memory) which sits on the graphics card. The table contains a triplet of RGB *voltage* values V_{rgb} which typically have precision greater than the intensities in the frame buffer. e.g. the voltages values in the LUT might have 10 or 12 bit precision, rather than 8 bits. The voltage values are converted to an analog signal (see DAC in the figure, which stands for “digital to analog converter”) and the analog signals are sent to the monitor via the cable

mentioned above. For simplicity, let's say that the RGB values and voltage values are in the range 0 to 1.

We will return to the question of how the voltages are chosen later. For now, let's go to the monitor and see what happens when the voltage value arrives there.

Brightness, contrast, and gamma

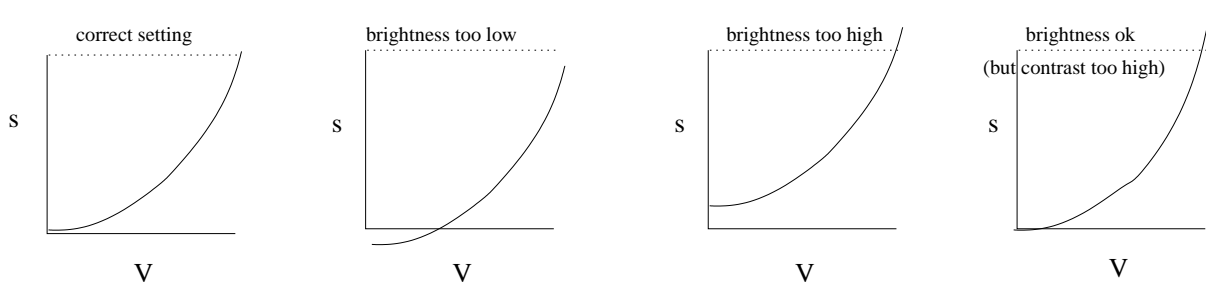
In the lecture on color, we considered the spectra of light that is *emitted* from a monitor (or projector). In particular, we examined the RGB spectra of the light emitted from the display and wrote these spectra as three columns of a display matrix \mathbf{D} . We used \mathbf{s}_{rgb} to scale these spectra, namely by considering the spectrum $I(\lambda) = \mathbf{D}\mathbf{s}_{rgb}$ leaving the display at a pixel.

The voltages V_{rgb} that arrive on the cable to the monitor are not the same physical thing as the values \mathbf{s} that weight the physical spectrum of light emitted from the monitor. Rather, the \mathbf{s} values at each pixel *depend on* the voltage values V . The \mathbf{s} values also depend on the “brightness,” “contrast” and “color” settings on the monitor which the user can adjust. (These adjustments help control the electronics of the monitor itself – they do *not* affect the voltage signal that arrives via cable to the monitor.)

Most monitors are designed so that, if the brightness and contrast are set properly, then the \mathbf{s} values should depend on the voltage V values via a power law:

$$\mathbf{s}_{rgb} = V_{rgb}^\gamma$$

where $\gamma \approx 2.5$ (see sketch on left). Crudely speaking, the brightness setting controls the height of the curve at $V = 0$. If brightness is too low, then the curve shifts downward – in practice, you cannot have negative \mathbf{s} values and so there is a cutoff whereby a range of values near $V = 0$ all give $\mathbf{s} = 0$ (see second fig below). If brightness is set too high, then the curve is shifted up and there are no voltages V that give the minimum emitted intensity that can be displayed by the monitor (see third figure).



The contrast should be set as large as possible to span (but not exceed) the range of what the monitor can display. If contrast is set too high, then the highest voltages (V near 1) will all map to the same (max) intensity of the display which is also not what we want (see figure on right). The color settings are similar to the contrast settings. They can raise or lower the maximum values, but they do so only by operating on one of the RGB channels of the monitor. Assuming the basic gamma model (with same gamma for RGB), we can treat the color settings as if we are multiplying by a constant κ_{rgb} .

In the end, the spectra that comes off a pixel on the monitor, if the monitor's brightness and contrast is set correctly, should be roughly:

$$I(\lambda) = \mathbf{D}\mathbf{s} = \mathbf{D} \begin{bmatrix} \kappa_r V_r^\gamma \\ \kappa_g V_g^\gamma \\ \kappa_b V_b^\gamma \end{bmatrix}$$

Gamma correction

I mentioned earlier the voltage lookup table which converts the I_{rgb} values in the frame buffer into voltages V_{rgb} . If we were to let the lookup tables contain the identity mapping i.e. $I = V$, then the scaling factors \mathbf{s} for light spectra intensities emitting from the monitor would be non-linearly related to the rendered values I_{rgb} in the frame buffer, because of the gamma built into the monitor.

The standard way to correct the monitor gamma is to let the voltage table perform the mapping

$$V_{rgb} = (I_{rgb})^{\frac{1}{\gamma}}.$$

i.e.

$$(V_{rgb})^\gamma = ((I_{rgb})^{\frac{1}{\gamma}})^\gamma = I_{rgb}$$

This is known as *gamma correction*. The result is that

$$I(\lambda) = \mathbf{D} \begin{bmatrix} \kappa_r V_r^\gamma \\ \kappa_g V_g^\gamma \\ \kappa_b V_b^\gamma \end{bmatrix} = \mathbf{D} \begin{bmatrix} \kappa_r I_r \\ \kappa_g I_g \\ \kappa_b I_b \end{bmatrix}$$

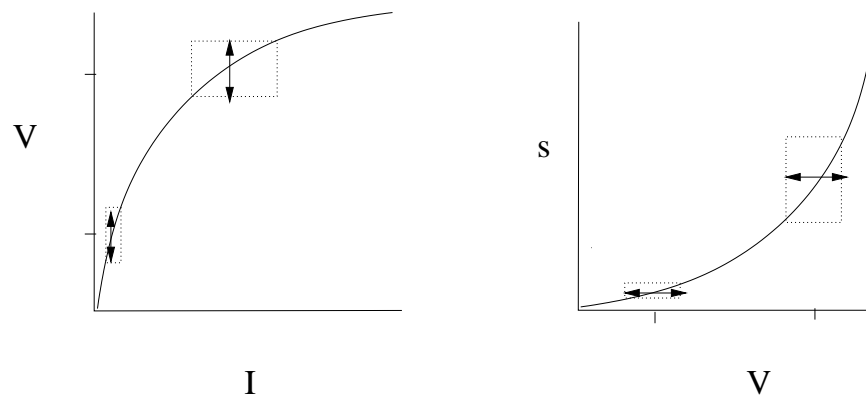
Gamma correct causes the spectrum $I(\lambda)$ that is displayed to be linearly related to the original I_{rgb} values.

Note that the above discussion did not address *why* monitors have a gamma behavior $\mathbf{s} = V^\gamma$, in the first place. Let's now turn to this question. The voltage signal that is carried on the cable is an analog (rather than digital) signal and is susceptible to small amounts of noise. So, although the voltage levels are specified with relatively high precision (say 10 bits), the *accuracy* will be less than this and the monitor will receive $V + n$ where n is some small random number (with mean 0).

Last class, I discussed how the visual system is more sensitive to changes ΔI in light intensity at lower intensities I (i.e. Weber's Law). Thus, if the monitor scaling factors s were proportional to V any given amount of noise n that is added to the voltage signal would be relatively more noticeable at low intensities than at high intensities.

In order to counter this perceptual effect, one effectively needs to encode the low light levels more accurately. To do so, one encodes the image intensities as voltages using a compressive non-linearity. When uniform noise is added to the voltages on the cable, this noise perturbs the low intensities by a small amount and perturbs the large intensities by a relatively large amount. (The perturbations shown in the figure below are magnified to exaggerate the effect.) Such a non-uniform perturbation is appropriate for reducing the effect of noise on human visual perception, since human vision is more sensitive to perturbations of intensities at low intensities. The monitor then inverts the non-linearity so that the resulting signal d_{rgb} is proportional to the original value of I_{rgb} .

Many video cameras *record* intensities using this gamma compression scheme. For example, the NTSC video standard which has been used in broadcast television for decades assumes that the



video is to be displayed on a monitor that has a gamma of about 2.5 and so the signals that are stored on tape have been already compressed relative to the intensities in the actual physical image using a $1/\gamma = 1/2.5 = .4$ compressive non-linearity power law. (Broadcast television has always worked this way, i.e. the NTSC signal is sent via electromagnetic waves through the air and picked up by an antenna.)

Thus, the main reason that monitors have this V^γ behavior and that we perform a non-linear transformation prior to sending the voltage signal to the monitor to protect the voltage signal against noise in the transmission and this non-linear transformation needs to be inverted.

Does OpenGL apply an inverse gamma encoding prior to writing images in the frame buffer? No, it doesn't. Graphics systems typically ignore gamma at the rendering stage and instead use the voltage lookup table to handle the gamma correction. Note: if you build in the gamma compression into the graphics system, you need to be careful since many operations such as interpolation of image intensities (Gouraud shading) and compositing require linear operations on the RGB values. This would be awkward to do if the values had been gamma corrected.

Summary

Let's briefly summarize the sequence of transformations by which rendered image intensities I_{rgb} in the frame buffer are presented on a display, and then observed by a human.

$$I_{rgb} \rightarrow V_{rgb} \rightarrow V_{rgb} + n \rightarrow s_{rgb} \rightarrow I(\lambda) \rightarrow I_{LMS}$$

The transformations are determined, in order, by the LUT, the cable, the brightness/contrast/gamma of the monitor, the light emitter(s) in the monitor, and the spectral sensitivities of the photoreceptors.

ASIDE

I finished the lecture with an informal discussion of various other display systems, in particular, what happens when you use a projector to display an image on a screen. I also discussed an example of an interesting application which has arisen recently, in which people are attempting to display images onto non-uniform surfaces. For more information on *many* such applications, see the Spatial Augmented Reality website: www.uni-weimar.de/medien/ar/SpatialAR/.