**Charles A. Poynton**
56A Lawrence Avenue E
Toronto, ON M4N 1S3
CANADA
tel +1 416 486 3271
fax +1 416 486 3657
poynton@poynton.com
www.inforamp.net/~poynton

# Gamma correction on the Apple Macintosh

*Gamma correction* is the process of precompensating for the nonlinear voltage-to-light intensity function of the CRT of a video monitor or a computer monitor, in order to obtain correct reproduction of intensity. Gamma correction is important if you want to obtain good picture quality and accurate color reproduction.

Charles A. Poynton, *A Technical Introduction To Digital Video*. New York: John WIley & Sons, 1996.

This note details the treatment of gamma correction on Apple Macintosh computers. If you want a more general and complete description of gamma that is not specific to the Macintosh, consult Chapter 6, *Gamma*, of my book. You can also consult my *Frequently Asked Questions about Gamma* (Gamma FAQ), available online through my home page on the World Wide Web (at the URL indicated above).

In most computer systems, RGB values are coded as unsigned integers using code zero to represent black and the maximum code – code 255 in an 8-bit system – to represent maximum intensity. What differs is the intensity interpretation of code values that lie between these extremes – for example, what is the physical intensity associated with the code value halfway between black and white, code 128 in an 8-bit system? That is the subject of this note.

When you adjust colors on your screen by manipulating sliders in an application, or copy a color from some object on your screen, or use the color picker, the process of gamma correction is going on behind your back. In these cases its technical details are of no importance to you, provided you obtain the colors you want. The issues in this note are important if you want to achieve good reproduction of continuous-tone imagery, get accurate colors from a video signal, exchange image data with other disciplines such as prepress, or to relate calibrated colors to numerical values.

## Gamma in CRTs

A CRT monitor is driven by analog *R'G'B'* voltage signals. In this note I will use the term *signal* to refer either to voltage, or to the digital code associated with voltage.

The physics of a CRT causes a voltage input to produce a physical intensity at the face of the tube according to the following relation:

$$intensity = voltage^{\gamma}, \quad \gamma \approx \frac{5}{2}$$

*Gamma* is defined as the exponent of the power function. The physics of a CRT dictates that gamma has a value of about $^{5}/_{2}$, that is, about 2.5, and the values found in practice are very close to this. The CRT transfer function has a parabolic graph, similar to the function $y = x^2$ that squares its argument. It is a *power function*, not an exponential of the form $y = e^x$. Incidentally, the nonlinearity is associated with the electron gun of the CRT: the phosphors themselves are quite linear, at least until the onset of saturation, at a voltage of about 75 percent.

To transform an intensity value into a voltage to drive a monitor, it is necessary to perform gamma correction using a power function having an exponent that is the inverse of the gamma value at the CRT. The gamma-correction function has a shape roughly similar to a square root function. The gamma correction function is approximately this:

$$voltage = intensity^{\left(\frac{1}{\gamma}\right)}$$

## Gamma in computer graphics

Computer graphics systems – such as rendering software packages – conventionally use intensity representations of image data, sometimes known as *tristimulus* or *linear-light*. These systems compute RGB values that are directly proportional to light intensity, and typically load 8-bit red, green and blue intensity values into the graphics system hardware (framebuffer). Gamma correction is accomplished by loading a gamma correction function into the framebuffer's lookup table. Computer graphics practitioners know *gamma* as the inverse of the exponent required for correction, that is, a value between about 2.2 and 2.8.

## Gamma in video

Video systems conventionally perform gamma-correction at the camera, using analog circuits. Video gamma correction corrects for a gamma value somewhat smaller than the CRT's actual power of 2.5, in order to take into account the typical dim surround viewing conditions of television. This obtains pictures that are subjectively more pleasing than if intensity were reproduced exactly. NTSC historically specified *gamma* as the value $^{1}/_{2.2}$ or about 0.4545. More modern video standards, including Rec. 709 for HDTV, call for the value 0.45 or about $^{1}/_{2.2222}$:

$$voltage = intensity^{0.45}$$

A video signal is mapped back into the intensity domain at the receiver by the CRT itself, with no need of explicit processing. This is obviously very economical: no components are required! But even more important than the easy mapping back to intensity domain is that gamma-corrected video signal coding has exceptionally good performance with respect to human visual perception. That is the main theme of the *Gamma* chapter of my book, referenced earlier. A video technologist sees little value in utilizing a coding space that has neither the perceptual benefits of video signal coding nor the physical "correctness" of intensity space.

## QuickDraw RGB codes

The main issue addressed by this note is the relationship between QuickDraw RGB values and physical intensity. The QuickDraw graphics system by convention employs neither the gamma-corrected signal coding of video, nor the intensity coding of computer graphics.

In the Macintosh, a color image is stored in a data structure called a *pixmap*. A pixmap can contain either indexed pixels or direct pixels. The indexed technique is known outside the Mac world as *pseudocolor*. An indexed pixel is usually represented in eight bits, so this mode is sometimes rather loosely referred to as 8-bit color. The direct pixel technique is known outside the Mac world as *direct color* or *truecolor*. A direct pixel is usually represented in 16 or 24 bits, and is loosely called 16-bit or 24-bit color. The hardware of a Mac graphics card also implements either indexed pixels or direct pixels. QuickDraw performs conversion of image data between these two forms where necessary. Both of these techniques use RGB values to specify color.

When an application presents a direct pixmap to QuickDraw to be displayed, the RGB values passed by the application are simply copied into the framebuffer. Every Macintosh graphics card is associated with a lookup table that maps from code to RGB voltage at the time of display. This lookup table is initialized by a gamma table that is stored in a 'gama' ROM resource on every graphics card.

By Apple's convention, the gamma lookup table contains a ${}^{1}/_{1.45}$ power function. It is possible to change the function but this is rarely done. The CRT in an Apple monitor has the same 2.5-power function as the CRT in a monitor from other manufacturers . The combination of the gamma lookup table's correction using a power of ${}^{1}/_{1.45}$, and a 2.5-power function at the monitor, causes RGB values presented to QuickDraw to be related to intensity by a power function having an exponent that is the quotient of these two values, ${}^{1.4}/_{2.5}$, or about ${}^{1}/_{1.8}$:

$$R'G'B'_{QuickDraw} = intensity^{\left(\frac{1}{1.8}\right)}$$

This relationship establishes Apple's unwritten rule of Macintosh gamma correction:

QuickDraw RGB values are interpreted as intensity values that have been subjected to a power function having an exponent of $^{1}/_{1.8}$.

Intensity is proportional to QuickDraw *R'G'B'* raised to the 1.8 power.

In terms of the relationship between QuickDraw *R'G'B'* values and voltage, the *R'G'B'* values can be thought of as being partially gamma-corrected. The function conventionally loaded into the lookup table of a Macintosh graphics board corrects the rest of the way:

$$voltage = R'G'B'_{QuickDraw}{}^{\left(\frac{1}{1.45}\right)}$$

The product of 1.8 and 1.45 is about 2.5, appropriate to precompensate the nonlinearity of the CRT and obtain correct intensity reproduction.

In QuickDraw, an indexed pixmap is associated with a *color lookup table* (CLUT), also known as a palette. RGB values associated with color lookup tables are given the same gamma interpretation as RGB values of a direct pixmap. When your application calls QuickDraw to load a color table entry with an RGB triple, the RGB values that you pass cannot directly be loaded into the hardware lookup table because the nonlinearity of the monitor has only been partially corrected. QuickDraw maps your RGB values through the gamma lookup table associated with the target device, to perform the remainder of the correction, then loads those values into the hardware lookup table.

If your application deals with physical intensity values, you must map from intensity to QuickDraw codes before calling QuickDraw. As an example, the CCD imaging device of a desktop scanner produces intensity values. If image data from a CCD scanner is to be passed to QuickDraw – to be displayed on the screen or written to a PICT file – the power function that maps intensities to QuickDraw codes must be applied by the scanner software.

Provision is made in QuickDraw and in the graphics card hardware architecture for cards from different manufacturers and cards for different monitors to have different gamma tables. In practice, though, all of the gamma tables in use are exact copies of the table in Apple's first "TFB" color card for the Mac. Color calibration systems for the Mac "trim" the gamma table to bring the response of the monitor into conformance with the ideal model represented by the standard TFB table, without altering the fundamental rule that relates QuickDraw RGB values to intensity.

There are various methods to alter software's handling of gamma on the Mac, including replacing the system's 'scrn' and 'gama' resources to change QuickDraw's default behavior. These methods are rarely used in practice, and the description given here applies providing your application does not invoke one of these techniques and your system 'scrn' and 'gama' resources have not been changed. If the default handling were to be changed, it would have the effect of changing the colors of pixmaps – such as icons and PICT files – whose RGB values were selected under the assumption that the display will impose a $^{1}/_{1.45}$-power function. Users would find it unsatisfactory to have the carefully-selected icon colors change, of course, and that is a major reason that display card manufacturers stick so firmly to Apple's gamma convention established by the TFB card.

### What's the gamma of a Macintosh?

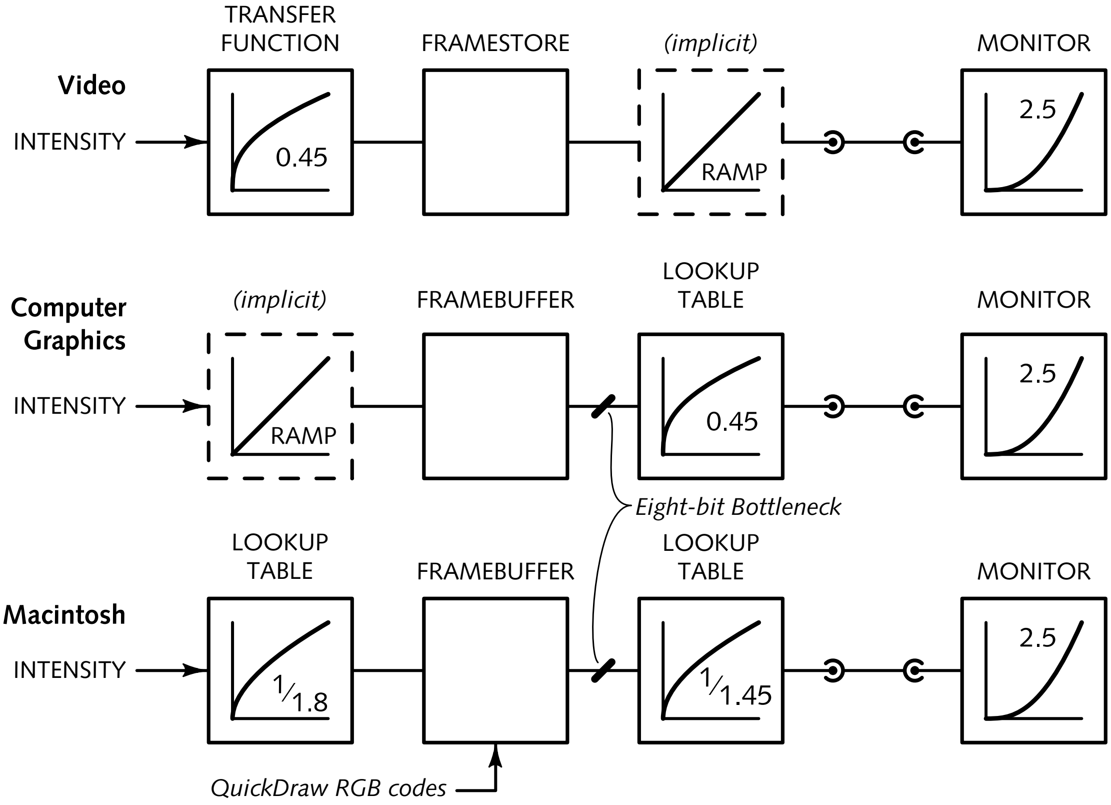
It is meaningful to talk about the gamma value of a monitor. It is also meaningful to talk about gamma-correction of a video signal for a particular value of gamma. It is not really meaningful to ask "What's the gamma value of a Macintosh?" Nonetheless, the question is frequently heard, often accompanied by one of the following three answers.

If you are a computer graphics expert, you assume that linear-light RGB intensity values are passed to the graphics library and reside in the framebuffer. No discussion of nonlinearity and no information contrary to this assumption can be found in Apple's documentation. You know that a Mac has a lookup table with a power function that maps code to voltage, in order to compensate the CRT's nonlinearity. You examine the lookup table and find that it contains a power function with an exponent $^{1}/1.45$. You conclude – incorrectly – that the Macintosh video monitor itself must map voltage to intensity with a gamma value of 1.45 instead of the more conventional value of 2.5. You declare the gamma of a Mac to be 1.45.

If you are a monitor expert, you know that all CRT monitors – including monitors from Apple – have fundamentally the same nonlinearity. The coding in the framebuffer and the values loaded into the lookup table are of no concern to you. You declare gamma of the Mac to be 2.5.

If you are a video expert, you can determine the function that maps QuickDraw RGB codes into intensity by loading image data codes into the framebuffer – say a ramp from 0 to 255 – and measuring the intensity values that result at the face of the screen. You discover that codes are mapped into intensity values by a power function with an exponent (gamma) of $^{1}/_{1.8}$. You are surprised that this is different from the $^{1}/_{2.5}$ relationship between signal codes and intensity that you are familiar with from video, but you conclude that the gamma of a Macintosh is 1.8.

The last of these three interpretations is the one used by Adobe's Macintosh applications. In the absence of any clear description from Apple on the way things work, Adobe's interpretation has achieved *de facto* status. The illustration below shows these three interpretations.



## Further reading

The volumes of *Inside Macintosh* detail the software interface to QuickDraw. However, they are not explicit about either numerical gamma values or the intensity interpretation applied to RGB values that are presented to QuickDraw. Chapter 9 of *Designing Cards and Drivers for the Macintosh Family, Second Edition*, has a hardware-oriented outline of the gamma correction process on the Mac, however it does not reveal any numerical value of gamma. Both of these documents describe the plumbing but make no reference to what substance is going through the pipes. I wrote this note because I found both of these sources to be unsatisfactory, and because I found different experts giving different answers to the same question.