Colors and Quantization

JPEG images, as well as some PICT images and quite a few QuickTime images, contain a lot of color information — 16.7 million colors' worth, to be exact.   This is just fine if you have 24-bit display hardware, capable of displaying in Apple's "Millions" mode, but unfortunately we aren't all so lucky.   The fact is that the single most popular color setup on the Mac is the standard 8-bit, or 256 color display.   It may not be immediately obvious that you can, in fact, view 24-bit color images with rather high quality on a 256 color display.   After all, how can you possibly go from 16.7 million colors down to 256 colors without losing something significant?

Video modes on the Macintosh

To understand how colors and quantization work, it is necessary to understand something about the way the Mac handles its monitors.   The current version of 32-bit Color QuickDraw, which is responsible for all drawing on your Macintosh, supports six different screen modes, depending on how much memory your Mac has available for the screen.   These modes are identified by the number of colors that each pixel, or dot, on your screen can represent. There are the direct video modes (Millions and Thousands in the Monitors control panel), the CLUT (or indirect) video modes (256 and 16 colors), and the monochrome modes (4-color and Black & White).

The direct video modes are so named because you can directly control the exact color of each pixel on the screen independently.   When you look at a pixel on your color screen, what you are actually looking at are three dots of light superimposed on top of one other: one red, one blue, and one green.   By varying the brightness of each of these three components, you can "create" virtually any color imaginable.   Thus, you can say that red, green, and blue are the three "primary" colors of your monitor, in much the same way that red, yellow, and blue are the three primary colors of your watercolor set; if you combine them in just the right proportions, you can get any other color you want.

In direct video modes, you specify for each pixel exactly how much red, blue, and green it should display on your screen.   This gives you complete control over the final result.   What then becomes important is how finely you can adjust the brightness levels of each of these components.   In Apple's Thousands mode, you can specify one of 32 possible brightness levels for each component, red, green, and blue, giving you 32 x 32 x 32 = 32,768 colors total. In Millions mode, you get to choose one of 256 possible brightness levels for each component, or 256 x 256 x 256 = 16,777,216 colors.

When you scale back to the 256-color and 16-color video modes, this direct control over the color components isn't nearly as effective, since you would only be allowed at most 6 or 7 possible brightness levels in each component. The Mac's solution to this problem lies in its use of color lookup tables (CLUTs) and palettes.   A CLUT is a small table of colors (with 256 or 16 entries, in this case) chosen from any of the 16.7 million colors available in the Millions mode.   To display one of these colors, the Macintosh gives the display hardware a color index, say, from 0 to 255, which is then used to pick a corresponding color from the CLUT to display on your screen.   A palette is essentially just another way of representing a CLUT.

This way of doing things gives you tremedous flexibility in choosing your colors.   For example, on a 256-color system, you can design a palette with 256 levels of greyscales for your display just as easily as you can design a palette with 256 colors chosen at random from the 16.7 million available.   Of course, what you would obviously prefer is the ability to choose the optimal set of 256 colors for displaying a given image, and this is what JPEGView's two-pass color quantization feature does for you.

Finally, I should admit that the 4-color mode is also technically a CLUT mode, even though I called it a monochrome mode before.   The reason I grouped it in with the black & white mode is that your Macintosh requires every display to be capable of showing black and white, which would leave only 2 colors that are truly free to be chosen.   Since you can't even get the essential red, green, and blue with those 2 colors, it is best simply to choose two intermediate greyscales to give you the highest-quality image, leaving you with a monochrome display.

JPEGView and color palettes

If at least one monitor in your system setup is operating in one of the CLUT modes — either with 16 colors or with 256 colors — then you can use JPEGView's color menu to select a new palette for the current image.   There are currently four different palettes available: the System Palette (âŒ˜Y), the Greyscale Palette (âŒ˜G), the Image Palette (âŒ˜J), and the palette produced with Two-Pass Quantization (âŒ˜T).

The first two options — the System Palette and the Greyscale Palette — are palettes that you're probably familiar with already, though you may not have known it. The former is simply the set of colors that you're used to seeing when you're in the Finder, while the latter is what you see when you switch your monitor into Greys mode in the Monitors control panel. To see what sort of color selection is provided by these other any other JPEGView palettes, you can open the Monitors control panel and look at the colors shown at the bottom. As you switch palettes within JPEGView, these colors will change to reflect the new set of colors.

Some images that you load with JPEGView, in particular GIF images, contain a palette internally, defining which set of colors should be used to display the image. If JPEGView finds such a palette within an image, it will allow you to choose that palette via the Image Palette item in the Colors menu. If no palette information is provided with the image, this menu item will not be available.

Finally, for images that contain Thousands and Millions of colors, JPEGView provides the ability to generate a custom palette for the image, produced by examining the distribution of colors within the image. This is known as two-pass color quantization, for the simple reason that JPEGView must first scan through the entire image once to determine the proper color palette, and then the image must be scanned a second time during the actual display process. For the other palettes available, the display process only requires one pass through the image.

Two-pass quantization will very nearly always give you far better results than any of the other palette options. The main downside to this feature is that it simply requires more time to display an image. JPEGView leaves the default choice up to you; see the Preferences Settings section for more details on how you can control this behavior. In addition, once you have performed two-pass quantization, JPEGView allows you to save the calculated palette with the image, so that you don't need to perform the calculation again. The section on Saving Images explains how you can do this.

Dithering — how and why it works

No matter which palette is selected, you also have control over whether or not to dither the resulting image via the Dithering item in the Colors menu (âŒ˜B). Dithering is a technique that "simulates" colors that aren't in the current palette by modifying the colors of the surrounding pixels. As a simple example of how dithering accomplishes this feat, consider a random pixel in an image, whose color is supposed to be purple. If the palette contains a color that exactly matches that purple, that particular color is chosen, and there are no problems; dithering has no net effect.

But imagine a palette had no purples at all, but did have shades of red and blue. Without dithering, that purple pixel would appear as a red pixel, since red is the closest match we can get to purple with the current (red and blue) palette; unfortunately, this won't look very purple in the end. Now turn dithering on. Dithering finds the closest match — red — just as before, but then it looks at the difference between this closest match and the original color we wanted. In this case, dithering would notice that there should have been more blue in this pixel, and, to compensate for this loss, it would add a little bit of blue to the neighboring pixels.

Now imagine that one of these neighboring pixels, whose blue component has just been increased, was originally the same purple as the original pixel. With the adjustment, it is now a rather bluish-purple color. So, when it comes time to find the closest match to this new color, we will choose blue rather than red, because blue is closer to bluish-purple than red is. Again, we calculate the difference between the chosen blue color and the original color we wanted (bluish-purple), and see that some red was missing. Dithering will thus add a little bit of red to pixels surrounding this one, and proceed on to the next pixel, continuing this for all the pixels in the image.

In the end, an area of solid purple — which cannot be displayed as such with our example blue and red palette — will appear as a pattern of alternating blue and red dots, which will actually give the appearance of purple if you don't look to closely. In this way, dithering approximates colors by finding what was missing in the current pixel and distributing that difference to the surrounding pixel. The advantage of dithering is that it effectively simulates more colors than your monitor can actually produce. The disadvantage is that the image comes out a little less distinct, and can sometimes suffer from dithering artifacts. See the Hints and Tips section for more information on the latter problem.