Eerste Industriestraat 19-21
1401VL Bussum Netherlands
tel. +31 35 693 9261
info@compuphase.com

CompuPhase®

IBAN: NL79 ABNA 0477 3565 08
BIC / SWIFT: ABNANL2A
EU-VAT: NL170160656B01
Chamber of Commerce: 32041148

Colour metric

A personal note

Colour perception is a difficult and little understood problem, which seems to defy even the most ingenious mathematical expressions. When researching the implementation of colour quantization algorithms, I stumbled more than once on theoretical discussions that were difficult to follow, and sometimes nearly impossible to verify (or to reject, for that matter). I am an engineer, not a scientist, and the adage for engineers is "through measurement to knowledge".

The conclusions drawn in this paper are based on simple tests with real people. Somewhere in the paper I state that evaluating colour differences is subjective: when asked to pick the "closest" match for a specific colour from a small palette, the selections by the test persons differ. On the one hand, you may claim that the test setup must have been flawed. A good test should rule out subjectivity. On the other hand, you should also recognize that about one out of eight men is more or less colour-blind. As it turned out, individuals with a (slight) green-red deficiency were amongst my test persons, and their results count as much as everyone else's.

Once, I asked on UseNet about the "equivalent gray level" of a colour. The replies were "30% red + 59% green + 11% blue", without exception, and without hesitation or a critical note. I started a simple paint program and drew two rectangles: a blue one with RGB values (0, 0, 255) and a green one with RGB values (0, 48, 0). According to the formula, that everyone that was kind enough to respond to me had given, these rectangles should have the same brightness, but they clearly had not. Foley [et al.] implies (p. 589) that the equation applies to linear RGB, which is contradicted in Poynton's "colorspace-faq". After gamma correction (the green cube gets RGB (0, 131, 0), blue cube stays at RGB (0, 0, 255), for a gamma of 2.5), the result was still debatable. I wondered whether anyone had done the same, simple, 10-second test before asserting the 30%-59%-11% rule. For me, this rule is how NTSC encodes luma/chroma channels, not how the human eye perceives brightness. I have difficulty to believe that the weighting was correct for the phosphors that were used at the time that NTSC was carved in stone, because the phosphors used in contemporary computer monitors have not changed that much from those used in the early days.

I concluded a paper on the Microsoft Windows Palette Manager by insisting that you should experiment and verify your assumptions. I would like to conclude this note in the same spirit: do not take my word for granted, nor that of anyone else. Experiment! Compare!

Through measurement to knowledge
(H. Kamerlingh Onnes, 1882)

Motivation

- When you map a true colour (photographic) image to a reduced palette (say, 256 colours), every true-colour pixel must be mapped to the palette entry that comes closest to original colour.
- Vector Quantization is a lossy compression technique that, when applied to images, quantizes multiple pixels at a time. It is thereby an extension of the straight colour quantization (or palette mapping) described in the previous point, and it has the same quality criterions.
- Other lossy image compression techniques use a "quality criterion" —the difference between the original colour and the quantized colour should remain below some threshold. This requires that you can determine the difference between pictures, starting with the difference between pixels.

Thus the questions "what is the closest colour?" and "how does one measure the distance between colours?" become relevant.

This paper evaluates several common metrics of colour distance and presents new formula that is both simple and produces good results.

Overview

If we can map a colour in an (abstract) orthogonal three-dimensional space, the distance between two colours is $\|C_1 - C_2\|$, where $\|\ldots\|$ denotes the Euclidean distance. For a three-dimensional space (with dimensions *R*, *G* and *B*) the Euclidean distance between two points is calculated as follows:

$$\|C_1 - C_2\| = \sqrt{(C_{1,R} - C_{2,R})^2 + (C_{1,G} - C_{2,G})^2 + (C_{1,B} - C_{2,B})^2}$$

Graphic applications for the PC usually employ the *Red-Green-Blue* (RGB) colour space. This model maps well to the way the common *Cathode Ray Tube* (CRT) display works. These displays have three kinds of phosphors that emit red, green or blue light when they are struck by an electron beam. Another advantage of the RGB model is that it is a three-dimensional orthogonal space, precisely what we need for the Euclidean distance function.

In this paper, I will abbreviate the term $C_{1,R} - C_{2,R}$ to ΔR, and similarly for the green and blue components.

The problem with RGB is that it is not always the easiest model to use (that is why printers usually use the CMYK model), and more fundamentally, it does not model the way in which we (humans) perceive colour. Specifically, colour perception is *non-linear* and not exactly orthogonal.

The standard colour model is CIE XYZ colour space. All other models can be interpreted as different mappings or subsets of the CIE XYZ colour space. CIE is a complex model and, most of all, although it defines the space of all colours that we can distinguish, it does *not* provide a *perceptually uniform* colour space. Thus, the distance between two points in the CIE XYZ space has no relation to the relative closeness of these colours. After ten years of debate, the CIE could not reach agreement on a definition of a perceptual uniform colour space, and therefore applied its stamp of approval on *two* competing perceptual uniform colour models: CIE L*a*b* and CIE L*u*v* —but they, too, are regarded today as inadequate models for the perception of colour [Granger, 1994]. The CIE has more or less acknowledged this itself by taking a proposed modification by D. Alman into recommendation [Alman, 1993]. A specific flaw that several studies have identified is that CIE L*a*b* progressively overemphasizes colour differences as the chroma of the colours increases.

Both [Foley, 1990] and [Poynton, 1999] give formulae to convert from RGB to CIE XYZ and from there to CIE L*u*v*. Alman's modification is a bit harder to come by, unless you can find the magazine. However, there is an easy improvement of CIE L*u*v*, and one that has been suggested by several independent sources ([Granger, 1994], [Nemcsics, 1993]) is that the brightness L* is proportional to $\sqrt{Y}$ rather than to $\sqrt[3]{Y}$ (square root instead of cube root). According to reports, Nemcsics came to this conclusion after experiments with 2500 observers in a "complex environment" —an environment where the eyes of the observers are not fully adapted to the luminance level.

The video and television industry researched the issue of perceptually uniform colour models, to achieve high quality compression. Video channels have limited bandwidth; compactly coding the luminance and chrominance information is a requirement. But when you decide to throw some of the data away, you will wish to do so while retaining the best visual quality. Two well-known models that the video industry developed are "YIQ" and "YUV". YUV is used by the Betamax standard and PAL and SECAM (European television), as well as by a few computer graphic formats. NTSC (American television) uses YIQ; basically, YIQ is YUV with scaling factors optimized for a reduced bandwidth. For completeness, below are the matrices for transforming from gamma corrected R'G'B' to YUV and YIQ:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.463 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

R'G'B' to YUV

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

R'G'B' to YIQ

By the way, the "Y" of YUV and YIQ is the gamma corrected "brightness" component of the colour; the "Y" of the CIE XYZ model is linear (= *uncorrected*) brightness. Both are related, but they are not the same.

Other colour models have been developed with the goals of being easy to compute and making better use of the features and limitations of the human visual system. The document by E.M. Granger mentions the Guth ADT colour space, and Charles A. Poynton wrote in his "colorspace-faq": "Although it was not specifically optimized for this purpose, the non-linear R'G'B' coding [...] is quite perceptually uniform".

To recapitulate, and to set out a direction towards a solution:

- What we need is a formula that gives a "distance" between two colours. This distance will only be used in comparisons, to verify whether one colour, *A*, is closer to colour *B* or to colour *C*.
- In a perceptually uniform colour space, the Euclidean distance function gives this distance. This is the most straightforward (and obvious) solution, but not the *only* solution.
- There are three well-known colour models (CIE L*u*v*, YUV and R'G'B') that all score "quite well" in perceptual uniformity (or so they say...).
- The proper test for these and other formulae is to compare their choice of "the closest colour" to the colour that a person would pick.

With the last point in mind, I wrote a small program that creates a palette with colours from an RGB cube that is cut into 64 small cubes (two bits for each red, green and blue component, giving $2^2 \times 2^2 \times 2^2 = 64$ colours). The program displays one colour from the palette, e.g. colour 24, and the tester chooses the closest colour from the remaining 63 palette entries.

The next step is to let the program automatically choose the closest colour using any of the aforementioned colour spaces.

Note that the goal of my program is *not* to find the smallest noticeable difference in tint from one colour pad to the next; the colours that one of my "observers" may use to match to another colour are obviously different. My intent is to let them select the closest entry in order to gain insight in how people evaluate colour "closeness".

The results

Due to the small size of the test group that I used in this experiment, the results below should perhaps be considered anecdotal.

- Not surprisingly, the choice of the "correct" colour is subjective. One person may find the proper brightness more important than the proper hue, others feel that the replacement colour should approximate the hue and saturation as best as possible. Therefore, you will not find a single formula that suits everyone.
- **Non-linear R'G'B'** is only fair. In many cases, it selects colours that are too dark or too blue.
- **YUV** is always better then non-linear R'G'B', but it is far from perfect.
- **CIE $L^*u^*v^*$** makes many excellent choices, but in a few situations, it makes unacceptable errors. The modified CIE $L^*u^*v^*$ from [Granger] performs much better (the document by E.M. Granger suggested to use a *square* root for the L^*, instead of the standardized *cube* root, [Nemcsics, 1993] comes to the same conclusion). But even with the modified lightness curve, $L^*u^*v^*$ does not perform well in the range of pink colours (skin colours of the Caucasian race).
- Several individuals suggested a **weighted Euclidean distance in R'G'B'**, according to the formula:

  $$||\Delta C|| = \sqrt{3 \times \Delta R'^2 + 4 \times \Delta G'^2 + 2 \times \Delta B'^2}$$

  This function has practically the same result as YUV. Its simplicity and speed of calculation make it a better choice than YUV.

- As explained in the section "gamma correction" below, the perception of brightness by the human eye is non-linear. From the experiments it appears that the curve for this non-linearity is not the same for each colour. The weighted Euclidean distance presented works quite well for the subset of colours where the "red" signal is 128 or more (on a scale of 0-255). For the other half of the full R'G'B' cube, this different weighting produced better results:

  $$||\Delta C|| = \sqrt{2 \times \Delta R'^2 + 4 \times \Delta G'^2 + 3 \times \Delta B'^2}$$

  Although blue has a small contribution (about 10%) to the sensation of brightness, human vision has an extraordinarily good colour discrimination capability in blue colours [Poynton, 1999]. This might explain why colours with a large "blue" contribution need a different weighting than colours with little blue.

A low-cost approximation

The proposed algorithm (used by our products EGI, AniSprite and PaletteMaker) is a combination both weighted Euclidean distance functions, where the weight factors depend on how big the "red" component of the colour is. First one calculates the mean level of "red" and then weights the $\Delta R'$ and $\Delta B'$ signals as a function of the mean red level. The distance between colours C_1 and C_2 (where each of the red, green and blue channels has a range of 0-255) is:

$$\bar{r} = \frac{C_{1,R} + C_{2,R}}{2}$$

$$\Delta R = C_{1,R} - C_{2,R}$$

$$\Delta G = C_{1,G} - C_{2,G}$$

$$\Delta B = C_{1,B} - C_{2,B}$$

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) \times \Delta R^2 + 4 \times \Delta G^2 + \left(2 + \frac{255 - \bar{r}}{256}\right) \times \Delta B^2}$$

This formula has results that are very close to $L^*u^*v^*$ (with the modified lightness curve) and, more importantly, it is a more *stable* algorithm: it does not have a range of colours where it suddenly gives far from optimal results. The weights of the formula could be optimized further, but again, the selection of the closest colour is subjective. My goal was to find a reasonable compromise.

A C code snippet for the colour distance function

```
typedef struct {
   unsigned char r, g, b;
} RGB;

double ColourDistance(RGB e1, RGB e2)
{
  long rmean = ( (long)e1.r + (long)e2.r ) / 2;
  long r = (long)e1.r - (long)e2.r;
  long g = (long)e1.g - (long)e2.g;
  long b = (long)e1.b - (long)e2.b;
  return sqrt((((512+rmean)*r*r)>>8) + 4*g*g + (((767-rmean)*b*b)>>8));
}
```

Gamma correction

You may have noticed that the above paragraphs mention both RGB and *non-linear* R'G'B' colour spaces. The linear and non-linear RGB spaces are related through gamma-correction. The area of gamma correction is confusing, because the same term is used to describe several entirely different (but related) phenomena:

- Within a certain range, the non-linearity of the human eye matches a power function. The $1/\gamma$ fraction represents this power function, and its value is somewhere between 1/2 and 1/3 (the value depends, amongst others, on the viewing conditions — surround light, for example). As a compromise, much literature nowadays assumes a value between 1/2.2 and 1/2.5.
- The non-linearity between the voltage applied to a grid of a CRT and the resulting lightness of the phosphor is a power function whose value is close to 2.5. Incidentally, the non-linearity of CRT displays is a function of the electrostatics of the cathode and the grid of an electronic gun. The phosphors themselves are quite linear, at least until an intensity of about 75% where saturation starts to set in. The recurring theme in talks and newsgroups that computer displays vary widely in gamma is almost always due to bad adjustment of the monitor (the "black-level" error) and the display's reflection of ambient light.
- Video (television) and computer imaging copied the non-linearity, and the gamma symbol, from photography. The camera film is not linear, but it can be approximated (within a certain range) with a power function.

Common computer displays have a non-linear relation between the light intensity (*I*) output by a phosphor and the voltage (*V*) that is applied to the grid of the CRT. The non-linearity of a CRT is usually defined as:

$$I = k \times V^{\gamma}$$

for constants *k* and γ. The value of γ is approximately 2.5 for all CRT displays; *k* may be assumed 1.0 for purpose of illustration. Note that this formula does not take the black-level error into account. When black becomes dark gray (the essence of the black-level error), the perceived error in gamma changes dramatically. Since computer monitors are often badly adjusted, a better approximation of the gamma correction formula would be to fix the gamma at 2.5 and to make the black-level error explicit:

$$I = k \times (V + \varepsilon)^{2.5}$$

To put these formulae into a more "visual" perspective: when a typical computer monitor has "contrast" and "brightness" knobs, the "contrast" knob sets the *k* variable and the "brightness" setting adjusts the ε variable.

The transformation of the linear RGB colour space to the non-linear R'G'B' space is known as gamma correction. The red, green and blue channels of a colour each go through the formula:

$$S' = S^{1/\gamma}$$

where *S* is the source signal (colour component) and *S'* is the corrected source; both *S* and *S'* are in the range 0.0 to 1.0.

In practice, the palettes of 256-colour images that are to be displayed on a computer monitor have already been "corrected" for this non-linearity so that the image shows the correct colours without further processing.

See Charles Poynton's FAQ and article for more fascinating reading on these topics, as well as the "proper" formula (standardized for video and HDTV) of gamma correction. When your pictures or screen lay-outs must look *right* on a typical computer display in a typical office however, you need to "tweak" and tune the brightness levels, rather than rigidly applying formulae: the black-level error and the reflection of ambient light make the issue of gamma correction on computer displays quite senseless.

References

Alman, D.H.; "Industrial color difference evaluation"; Color Research and Application; No. 18 (1993); pp. 137-139

Foley, J.D., A. van Dam, S.K. Feiner, J.F. Hughes; "Computer Graphics, Principles and Practice"; second edition; Addison-Wesley; 1990; pp. 574-600.
An overview of many colour models, with the focus on how they relate to each other and to the CIE XYZ model. On page 589, the book says: "The Y component of YIQ [...] is defined to be the same as the CIE Y primary". As the CIE Y (luminance) is linear, this implies that the Y channel of YIQ (video luma) is linear as well. This is explicitly contradicted by Poynton's colorspace-faq, item 10.

Granger, E.M.; "Is CIE L*a*b* Good Enough for Desktop Publishing?"; technical report, Light Source Inc.; 1994.
Granger claims that CIE L*a*b* has flaws. The Guth ADT colour space is proposed as an alternative.

Nemcsics, A.; "Color Dynamics"; Publisher Akadmiai Kiad, Budapest; 1993.

Poynton, Charles A.; "Gamma and its Disguises"; Journal of the Society of Motion Picture and Television Engineers; Vol. 102, No. 12 (December 1993); pp. 1099-1108.

Poynton, Charles A.; "Frequently Asked Questions About Colour" ("colorspace-faq"); 1999.
Maintained and available on the Internet in text (ASCII), Postscript and Adobe Acrobat formats.

Updated: 2019-05-23