The Language of Colour Management

This document contains a brief outline of a colour management system and definitions of the more common terms associated with it.

Those looking for more authoritative documentation can find it at the International Color Consortium's website, currently at http://www.color.org.

Why Colour Manage?

If accurate colour processing is to be done there must be some method to match the way different devices see colours. A scanner may see red better than blue so an image coming from this scanner would appear abnormally red on a monitor which wasn't able to compensate. If the printer's green ink was a brighter than its blue then this would contribute to more colour inaccuracies. Whilst it is quite possible to balance these factors out and make good prints on a single system it is a difficult calibration process to carry out and one which needs constant supervision. *Colour Management* provides a method by which this calibration process can be largely ignored by the user whilst also providing the opportunity to move images between different computer systems with a reasonable expectation that the image can be rendered the same on each.

The conversions outlined are carried out by a *Colour Management System* (CMS). This is a set of routines made available to an image processing application.

Examples of CMSs are the Windows ICM, Adobe's ACE and LittleCMS, an open source CMS.

The CMS maintains an image within the *Profile Connection Space* (PCS). This is an internal device-independent space which can represent all colours. The CMS also handles the movement of an image into the PCS by using *input profiles* to convert an image's colours to the chosen colour space on introduction, and *output profiles* to convert the image from the chosen *colour space* into the gamut of the output device, normally screen or printer.

A *device input profile* is not absolutely necessary to produce a satisfying image. However, it is possible for an input device to be able to reproduce colours which fall outside the boundaries of the common *colour spaces*. This is especially true of top-end digital cameras. A *device input profile* will control how these colours are treated.

Many input devices do not have *device input profiles*. As it is necessary for an image to be assigned a specific *colour profile*, if an image does not have one, one should be allocated prior to any image manipulation. Many digital cameras will assign a *colour space* name, normally sRGB, to an image in the EXIF data. This is not a profile and there is no mapping of colours but the information allows an image manipulation application to assign the correct profile on loading.

A *device output profile* is absolutely necessary for each output device to allow the correct rendering of an image in a colour managed environment. Without them an image can not be viewed correctly on-screen nor can it be correctly printed.

Colour Models.

Colours are often described in a relatively abstract mathematical way . Normally this involves three (in the case of RGB, for instance) or four (in the case of CMYK) values or colour components.

In the RGB *Colour Model* one can describe a 3-D shape with (say) Red on the x-axis, Green on the y-axis and Blue on the z-axis. The 3-D shape so defined contains all theoretical colours.

Examples of *colour models* are RGB, CMYK and LAB.

Colour Spaces.

Although all the colours in such a 3-D shape are theoretically possible, in practice this is not so. Hence it is necessary to apply a mapping function to the colour model to define a subset of colours which are reproducible. This subset of colours is called a *colour space* and it occupies a volume, and is completely contained, within our original 3-D shape.

Examples of *colour space*s within the RGB *colour model* are sRGB, AdobeRGB and ProPhotoRGB.

In the CMYK *colour model*, a *colour space* may refer to any of a large number of printer's ink sets, examples are Euroscale, US Sheetfed and Japan Coloured.

Color space is thus a term for a certain combination of a *color model* plus a mapping function. The term *color space* is often incorrectly used to also identify *color models* since identifying a *color space* automatically identifies the associated *color model*. Thus although several *color spaces* are based on the RGB *colour model*, it would be wrong to refer to the RGB *color space*.

Gamut

The range of colours occupied by a *colour space* within a *colour model* is called the space's *Gamut*. Any colour which dos not fall within this volume is referred to as *Out-of-Gamut*.

Colour Profiles

A *Colour Profile* is a piece of computer code which contains a mapping function. It can stand alone or be contained within an image file. There are a number of different types of *profile* but the two most common are:

Document Profiles

This is often a small generic piece of code which defines the *colour space* in which the colours in an image reside. Several such *profiles* are normally available on a computer which is capable of colour management. When there is need to embed a profile into an image and no specific *Input Device Profile* is available then a document profile should be embedded instead.

Device Profiles

A *device profile* provides the data required to map the range of colours the device is capable of rendering, to a *colour space*. An input device, such as a scanner, 'sees' colours quite differently to a digital camera so its *Input Device Profile* must take that into account. An output device, such as a monitor or printer, must have an *output device profile* to allow the mapping of the image colours to the *gamut* of the device. Such *profiles* are normally provided by the device manufacturer.

Rendering Intent

When a colour managing image processing application sends an image to the monitor it does so using the monitor *output profile* which relates the colours in the *colour spac*e assigned to the image to the capabilities of the monitor. Normally this works well because the monitor's *gamut* is relatively wide. When the same image is sent to a printer this is not often the case. The inks used in the printer work in a way which reduces the saturation available as the image darkens and a printer has problems rendering a true black. In other words its *gamut* is narrow.

In consequence the printer's *device profile* must convert any of the image's colours which the printer cannot render into colours which it can. This conversion can be handled in a number of ways according to the *Rendering Intent*.

Relative Colorimetric

One such method might be to simply clip all the *out-of-gamut* colours to the nearest available colour while leaving all the *in-gamut* colours alone. This is the *Relative Colorimetric rendering intent*. The image's white point is also remapped to coincide with the white point in the new *gamut*.

Perceptual

Conversely the *Perceptual rendering intent* converts an image by maintaining the relationship between all the colours in an image whilst pulling the *out-of-gamut* colours into *gamut*. Thus all the colours in the image are altered but the relationship between them is preserved.

Absolute Colorimetric

The *Absolute Colorimetric rendering intent* is very much like *relative colorimetric* except for the handling of the white point. The *relative* method remaps the white point from its position in the the original *gamut* to the white point of the new *gamut* whereas the *absolute* method does not. The *absolute* method can thus introduce colour shifts.

Saturation

The *Saturation rendering intent* aims to maintain saturation during conversions. It is more useful when moving from a more restricted *gamut* to a wider one. Accurate colour relationships are not maintained during conversion.

The latter two intents are of little interest to image manipulators. Which of the former intents is chosen is largely down to the individual but many would recommend the use of *relative colorimetric* first.

Choosing a colour space

The format of the image may determine what colour space is appropriate. If an image is for publication it is likely to be CMYK. Thus the colour space and the profile will be dictated by the requirements of the printing method, notably the inkset used.

When using RGB images the choice of *colour space* is largely personal as the differences between many RGB *colour spaces* is not great. A *colour space* encompassing the colours present in the *gamuts* of the devices used would be perfect but is often not possible so it is necessary to choose that which gives the nearest fit.

This is difficult to test in practice and many photographers have concluded, through experience, that AdobeRGB is a better choice than sRGB as the loss of saturation is more than compensated for by the increased range of colours available.

sRGB

The most common space is sRGB which was primarily designed to show well on a CRT monitor. It is the default space for viewing images on the web and, if that is the intended target for an image, then the choice of *colour space* should be sRGB.

AdobeRGB

The *gamut* of the sRGB space is quite limited and many devices can capture or display a wider range of colours than sRGB allows. AdobeRGB was defined to give a wider *gamut* but does suffer a drop in saturation level. It is more suited to colour printing and is currently the most popular choice for many photographers.

ProPhotoRGB

A newer *colour space*, called ProPhotoRGB, is gaining in popularity in some circles. It has a much wider *gamut* than even AdobeRGB but suffers few of the saturation problems normally associated with *wide gamut* spaces. ProPhotoRGB encompasses better the wide range of colours detectable by modern high-end digital cameras but no output device is currently able to match this. Development in this area is rapid and it is likely that this will change but it may be a space to avoid until this occurs.