

3

Lab

Färg

Jörgen Rydenius, 1997

Chapter 1.

PREPARATIONS

To fully understand this computer exercise, some work has to be done prior to the actual occasion. Note that this might take a few hours.

The first thing to do is to read the theory described in chapter 2, “Basic Colour Science” thoroughly. Without knowledge of that theory the computer exercises will be useless! Then you should be able to solve the preparation exercises below. Your solutions will ease your work on the computer exercise.

The Matlab environment will be utilised, so if you have not used it for a while it could be a good idea to rehearse the basic functionalities of that program.

1.1 TRISTIMULUS CALCULATION

- Find a way to calculate the *XYZ* tristimulus values in Matlab. That is: find out how to multiply two equal sized spectrum vectors component wise, and how to summarise the product without using *for* or *while* statements.

1.2 RGB TO CMYK CONVERSION

- Find the direct mathematical conversions between *RGB* and *CMYK* on page 6 (and the inverse transformation), without using the intermediate *CMY* colour space. Try to simplify the equations algebraically.
- Write the Matlab functions *rgb2cmk* and *cmk2rgb*. The first shall have three input argument and four output argument. The second shall have four input arguments and three output arguments. All arguments can be matrices, but it can be assumed that all input matrices have the same size. Try to vectorise your code, by using matrix manipulations instead of iterative

loops with *for* or *while* statements. It is possible to completely avoid loops in this code.

Help is available on-line at:

<http://www.mathworks.com/support/tech-notes/1100/1109.html>

1.3 CIELAB TO XYZ CONVERSION

- Figure out how to algebraically invert equations 2.18 to 2.21. That is: how to convert from CIELAB to *XYZ* tristimulus values.

1.4 WHITE POINT

- How does the white point move in the *xy* chromaticity diagram depending on the temperature of the standard illuminant? Towards which hue does it move for higher or lower temperatures? Use the following values, and plot them in the diagram on page 18:

<i>Illuminant</i>	X_n	Y_n	Z_n
D50	96.42	100.00	82.49
D55	95.68	100.00	92.14
D65	95.04	100.00	108.89
D75	94.96	100.00	122.61

1.5 HALFTONES

- Calculate the 16 fractional areas when printing with 30% cyan, 40% magenta, 50% yellow, and 10% black, by using Demichel’s equations. Write the results in column one in the table on page 15.

Chapter 2.

BASIC COLOUR SCIENCE

This is a brief introduction to basic colour science. The emphasis is on mathematical models and different colour spaces. The text was written as a course material for a course in graphical image technology. Most of the illustrations are taken from “Inverse Halftoning of Scanned Colour Images” (see references).

2.1 WHAT IS COLOUR?

The human eye is able to detect light—i.e., electromagnetic radiation—with wavelengths in the interval between 380–780nm. The radiant flux of the observed light at each wavelength is expressed by a *Spectral Power Distribution* (SPD), such as in figure 2.1. We see colour by means of light sensitive cells called cones on the eye’s retina. There are three types of cones, sensitive to wavelengths approximately corresponding to red, green, and blue light¹. The response signals from the cones upon stimuli are processed by the brain, which associates the signal to a visual colour sensation.

Sir Isaac Newton said: “Indeed rays, properly expressed, are not coloured.” He was right—SPDs exist in the physical world, but colour only exists in the eye and the brain.

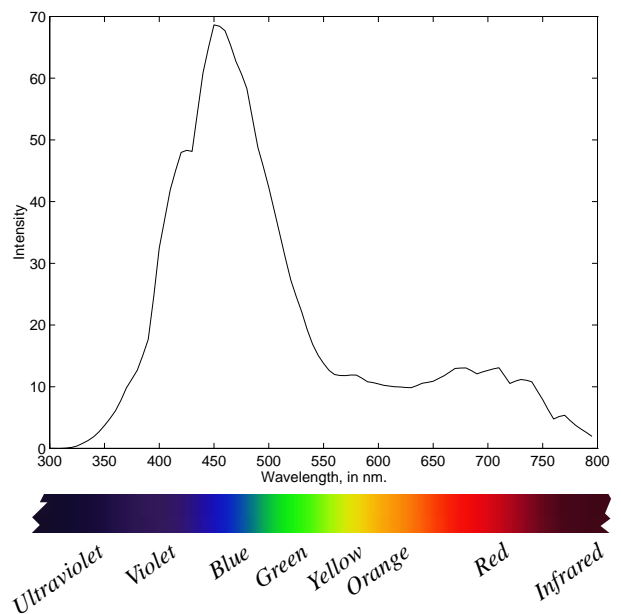


Figure 2.1 An example of a spectral power distribution. This distribution will be perceived by the human observer as a bluish colour.

2.2 THE HUMAN EYE

The human eye has three different colour receptors called cones. They are referred to as *L*, *M*, and *S* cones, since they are sensitive to long, medium, and short wavelengths in the visible spectrum, respectively. In addition to the cones there are also receptors called rods. They are of one type only, are more sensitive to light, and are used for night vision. Most of the cones are concentrated to the central part of the retina, called *the fovea*, while the rods are frequent outside the fovea (see figure 2.2). This means that in the dark, the human eye has the least sight in the central visual field. This becomes very obvious when looking at distant stars in

1. Actually, according to Bourgin (see references), recent studies show that the human eye has more than three cone types. However, these additional cone types do not have any large impact on the basics of colour vision and are not considered here.

the night sky. When looking a little bit away from the star it is visible, but when looking directly upon it, it disappears.

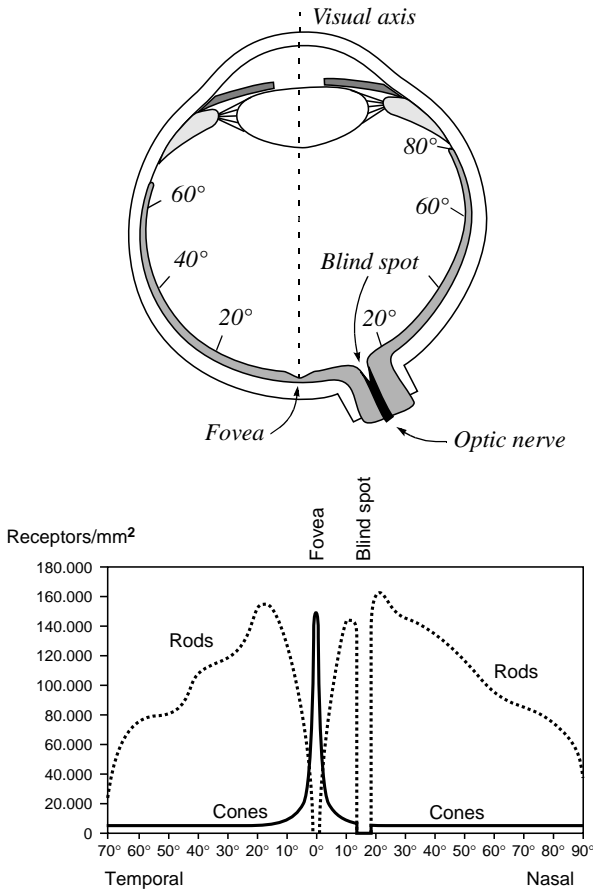


Figure 2.2 Rod and cone density, as a function of retinal location. The cones are mostly concentrated to the fovea, while the rods are frequent outside the fovea.

It is rather hard to determine the spectral sensitivity for each type of cone. One possible approximation is to measure how much energy the L , M , and S cones absorb, as a function of the light's wavelength. This is shown in figure 2.3.

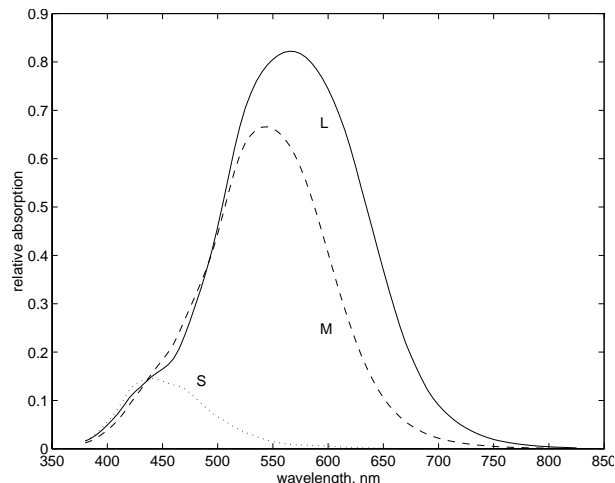


Figure 2.3 Energy absorption spectra for L , M , and S cones.

The actual detection of light in the eye, both for cones and rods, is an electro-chemical process. Photons from the incoming light interact with pigments in the cones and rods, changing their chemical characteristics. The cones are photon detectors, rather than energy detectors. The functions above are thus only a rough approximation of the actual sensitivity functions. In practice, power distributions are used rather than photon distributions, but the error this introduces is compensated for. Two different incoming photon distributions may create the same chemical reaction in the cones, and are thus perceived as the same colour. Because of this, the SPD is not an intuitive measure for humans perception of colour.

Sometimes the L , M , and S cones are referred to as *red*, *green*, and *blue* cones, respectively, named after the dominant colour of the wavelengths they are sensitive to.

Together with the information from the rod cells, the information from the cones is encoded and sent to higher brain centres. The encoding, known as opponent process theory, consists of three opponent channels:

- Redness – greenness.
- Blueness – yellowness.
- Blackness – whiteness.

2.3 MEASURING COLOURS

According to the previous discussion, stating that the impression of colour is related to how the human eye works, it is natural to use the eye's three sensitivity functions as a mathematical foundation. Furthermore, light with different spectral distributions yielding the same colour impression, should be measured as the same colour.

Denote the incoming light's spectral photon distribution $E(\lambda)$, and the sensitivity functions of the cones as $L(\lambda)$, $M(\lambda)$, and $S(\lambda)$. This gives the following total stimulations of each cone type:

$$L_{tot} = \int_{\lambda} E(\lambda)L(\lambda)d\lambda \quad [\text{EQ 2.1}]$$

$$M_{tot} = \int_{\lambda} E(\lambda)M(\lambda)d\lambda \quad [\text{EQ 2.2}]$$

$$S_{tot} = \int_{\lambda} E(\lambda)S(\lambda)d\lambda \quad [\text{EQ 2.3}]$$

The values received by calculating such integrals over the incoming light and sensitivity functions are referred to as *tristimulus values*. $E(\lambda)$ can originate from a light source, or it can be reflected light from some object. The latter case can be described in the following way: denote the photon distribution from the light source illu-

minating the object as $I(\lambda)$, and the objects influence on the incoming light, the reflectance function, as $R(\lambda)$. Then make the following substitution in the equations above:

$$E(\lambda) = R(\lambda) \cdot I(\lambda) \quad \text{[EQ 2.4]}$$

The implications of equation 2.4 are that two objects may have the same colour in one illumination and then look very different from each other in another, and that a single object is perceived as having different colours when viewed under different illuminations. These effects are called *metamerism*. Metamerism is a large problem when two colours are supposed to match each other. It is more or less unavoidable, unless the two objects have exactly the same reflectance functions.

The equations 2.1 to 2.3 would be a useful way of expressing colours, if only the sensitivity functions of the cone types were known exactly. Since they are not, a different approach has to be taken. In 1931, CIE² proposed that the sensitivity functions for the L , M , and S cones should be replaced by three other well defined sensitivity functions, called $r(\lambda)$, $g(\lambda)$, and $b(\lambda)$. These were found by an experiment, described in figure 2.4. The idea is to let a test person create the same colour sensation as a monochromatic variable reference light, by mixing three monochromatic sources of fixed wavelengths. The test is repeated with different reference wavelengths (but with the same intensity), and different test persons. The average results (“the standard observer”) are the CIE colour matching functions (see figure 2.5).

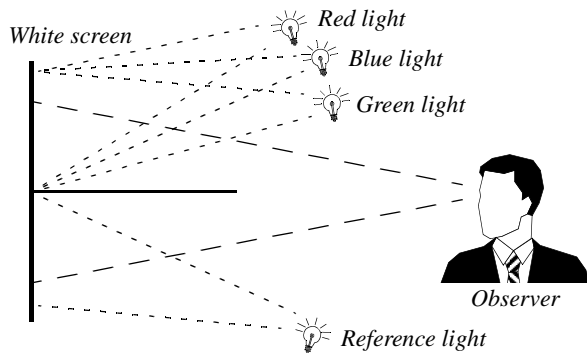


Figure 2.4 Simplified illustration of the experimental setup, when determining the colour matching functions.

The wavelengths of the red, green, and blue lights used in the experiment above are very well defined: red is 700nm, green 546.1 nm, and blue is 435.8nm. The green and blue lights were chosen to coincide with lines in the discharge spectrum of mercury, making calibration easy. The red light was chosen in a region where the eye is less sensitive to variations in hue.

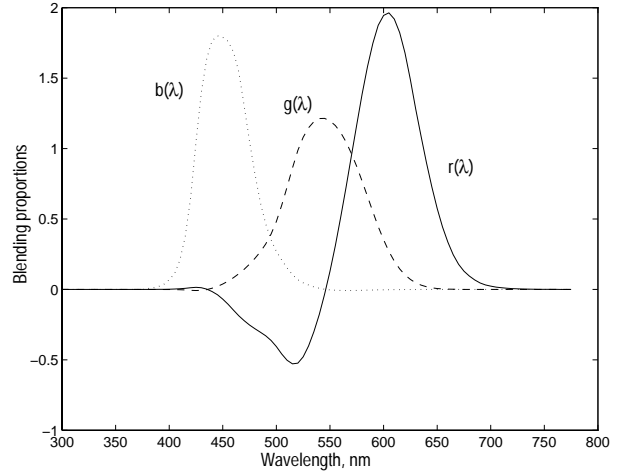


Figure 2.5 The rgb colour matching functions.

As seen in figure 2.5, not all of the reference wavelength colours were possible to create with the basis colours. Then one or more of the lights were moved over to the reference side, until it was possible to balance the two sides. This is seen in the graph as negative values. Of course, there is no such thing as negative light. It is only a result from the fact that the chosen basis colours can not reproduce all colours. The set of colours that can be reproduced is referred to as *the colour gamut*.

To avoid negative values in the colour matching functions, a linear transformation was concocted:

$$\begin{bmatrix} \bar{x}(\lambda) \\ \bar{y}(\lambda) \\ \bar{z}(\lambda) \end{bmatrix} = \begin{bmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,81240 & 0,01063 \\ 0 & 0,01 & 0,99 \end{bmatrix} \begin{bmatrix} r(\lambda) \\ g(\lambda) \\ b(\lambda) \end{bmatrix} \quad \text{[EQ 2.5]}$$

These transformed functions can be seen in figure 2.6. The transformation matrix may seem strange—why more decimals in the middle row? The answer is that the $\bar{y}(\lambda)$ was desired to match the eye’s spectral luminous efficiency curve for high light level (photopic) vision. It simply took five decimals to achieve a good match.

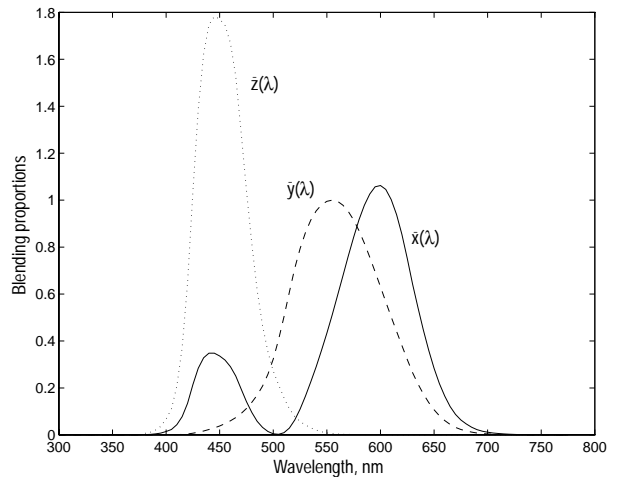


Figure 2.6 The xyz colour matching functions.

2. CIE is short for *Commission Internationale de l’Éclairage*.

From the colour matching functions defined in equation 2.5, tristimulus values can be calculated. They are normalized for the current illumination, so that a completely white surface (reflectance function equal to one, for all wavelengths) always will give $Y = 100$:

$$X = k \int_{\lambda} R(\lambda) I(\lambda) x(\lambda) d\lambda \quad [\text{Eq 2.6}]$$

$$Y = k \int_{\lambda} R(\lambda) I(\lambda) y(\lambda) d\lambda \quad [\text{Eq 2.7}]$$

$$Z = k \int_{\lambda} R(\lambda) I(\lambda) z(\lambda) d\lambda \quad [\text{Eq 2.8}]$$

$$k = \frac{100}{\int_{\lambda} I(\lambda) y(\lambda) d\lambda} \quad [\text{Eq 2.9}]$$

where, as before, $I(\lambda)$ is the photon distribution from the light source illuminating the object, and $R(\lambda)$ is the objects influence on the incoming light, the reflectance function. SPDs are often only measured in sample intervals of 5nm, so the integrals above will in practice be replaced by sums.

The illumination is often assumed to be one out of several standard illuminations, defined by CIE. Commonly either D50, or D65 is used. They have approximately the radiation characteristics of black bodies at the temperature (in Kelvin) given by the number after D multiplied by 100. Each illumination set has its own *white point*, for example:

- $X_n = 96,42$, $Y_n = 100$, $Z_n = 82,49$ for D50.
- $X_n = 95,04$, $Y_n = 100$, $Z_n = 108,89$ for D65.

The white point is calculated by assigning $R(\lambda) \equiv 1$ in the equations above. Note that the colours of all objects in the particular illumination are in the intervals $0 \leq X \leq X_n$, $0 \leq Y \leq Y_n$, and $0 \leq Z \leq Z_n$. They can not be negative since the colour matching functions are positive, and they can not be higher than the white point, since no object can reflect incoming light better than an object with reflectance function equal to one, for all wavelengths³.

2.4 COLOUR SPACES

From the XYZ tristimulus values defined above, several different colour spaces can be derived, each suitable for different applications. We are going to look a little closer on the following spaces:

- *RGB, CMY, and CMYK*—simple device dependent colour spaces for reproduction on computer monitors, or on paper.
- *Chromaticity spaces, such as CIE xyY*—colour spaces based on the chromaticity characteristics. What this means will be explained later.
- *CIE 1976 ($L^*a^*b^*$ or $L^*u^*v^*$)*—device independent colour spaces created for good perceptual uniformity. That is, colours at equal distance in any direction and at any location in the coordinate system, will be perceived as equally different by the human eye.
- *HSV, HSL, and related colour spaces*—colour spaces based on *RGB*, designed to be intuitive for humans to use.

RGB, CMY, and CMYK

In this section all variables are assumed to be in the interval $[0, 1]$.

The most popular colour spaces are probably *RGB* and *CMYK*. The first is an acronym for Red-Green-Blue and is used on computer monitors, since those utilise these three colours as primary colours. The second is an acronym for Cyan-Magenta-Yellow-blacK and is used for paper printing, since these are the ink colours utilised in the colour printing process. The link between them is the *CMY* colour space, Cyan-Magenta-Yellow. They are all device dependent spaces—the actual colour is depending on the device's characteristics. One proposed conversion between these formats is shown below⁴. However, there are several suggested conversions available. When converting between *CMY* and *CMYK*, an index showing the number of colour components in the space is present to distinguish the different components. Why the fourth (black) colour is used will be explained later.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 - R \\ 1 - G \\ 1 - B \end{bmatrix}; \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 - C \\ 1 - M \\ 1 - Y \end{bmatrix} \quad [\text{Eq 2.10}]$$

$$\begin{bmatrix} K \\ C_4 \\ M_4 \\ Y_4 \end{bmatrix} = \begin{bmatrix} \min(C_3, M_3, Y_3) \\ \frac{C_3 - K}{1 - K} \\ \frac{M_3 - K}{1 - K} \\ \frac{Y_3 - K}{1 - K} \end{bmatrix} \quad [\text{Eq 2.11}]$$

3. This is not the whole truth. Fluorescence could theoretically make the values larger, but that is not considered here.

4. PostScript defines the conversion to *CMYK* slightly different, but the equations above produce better result. Adobe Photoshop, however, uses the equations above.

$$\begin{bmatrix} C_3 \\ M_3 \\ Y_3 \end{bmatrix} = \begin{bmatrix} C_4 \cdot (1 - K) + K \\ M_4 \cdot (1 - K) + K \\ Y_4 \cdot (1 - K) + K \end{bmatrix} \quad \text{[Eq 2.12]}$$

The question that now remains is: how do we convert from XYZ tristimulus values to any of the three colour spaces above? This is where the device dependency becomes obvious. The transformation from XYZ to RGB must depend on the device's characteristics, and primarily its white point. These characteristics must be measured, or obtained from the manufacturer of the device. Computer monitors often have a white point close to that of D65 and a suitable transformation matrix is proposed to be:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 41,2453 & 35,7580 & 18,0423 \\ 21,2671 & 71,5160 & 7,2169 \\ 1,9334 & 11,9193 & 95,0227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{[Eq 2.13]}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = A^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{[Eq 2.14]}$$

where A is the matrix in equation 2.13. The connection to D65's white point is that the row sums in A are the white point coordinates for X , Y , and Z , respectively, for this particular illumination.

Tightly connected with the RGB colour space is the subject of *gamma correction*. The intensity of light generated by a physical device is usually not a linear function of the applied signal. A conventional cathode ray tube has a power-law response to voltage. The numerical value of the exponent in the response function is known as *gamma*. This non-linearity has to be compensated for to achieve a correct intensity reproduction on the computer monitor. The gamma correction is often applied as:

$$x_{new} = x_{old}^{1/\gamma} \quad \text{[Eq 2.15]}$$

The values of γ are normally in the interval from 1.0 to 3.0, which effectively lightens up colours in the shadows and the midtones.

Chromaticity

Sometimes the relative difference between the X , Y , and Z values are of more interest than the actual values. Then normalized versions, called chromaticity values, can be calculated⁵:

5. These x , y , and z are not the same thing as the colour matching functions which also were denoted with lower-case letters.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{X + Y + Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{[Eq 2.16]}$$

Since $x + y + z = 1$, one of the chromaticity values can be calculated from the other two, and can thus be ignored. Usually z is ignored. The chromaticity values gives a possibility to plot colours in a two-dimensional subspace, with x and y as coordinates. By filling out with the Y component from the tristimulus values, a complete colour representation is achieved, called CIExyY:

$$\begin{bmatrix} x \\ y \\ Y \end{bmatrix} = \begin{bmatrix} \frac{X}{X + Y + Z} \\ \frac{Y}{X + Y + Z} \\ Y \end{bmatrix}; \quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{Y}{y} \begin{bmatrix} x \\ y \\ 1 - x - y \end{bmatrix} \quad \text{[Eq 2.17]}$$

A plot of all monochromatic colours (pure colours of only one wavelength) in a chromaticity diagram is often referred to as *the spectral locus*. The spectral locus in the xy plane has the form of a horse shoe (see figure 2.7). Chromaticity diagrams are very common in colour science literature, and are useful as long as one remembers that the colours exist in a three-dimensional space, and not two-dimensional as the chromaticity diagrams may imply.

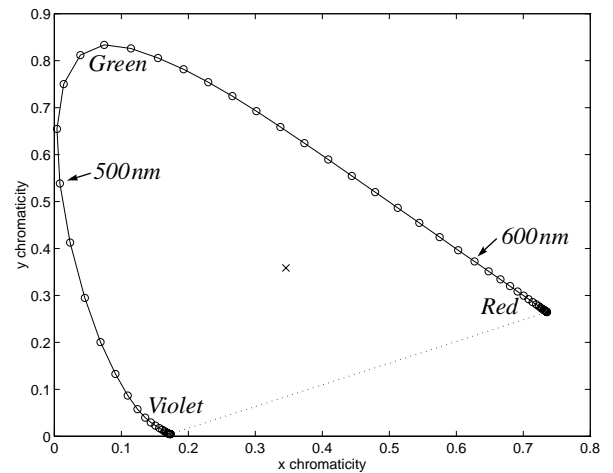


Figure 2.7 The spectral locus in a chromaticity diagram. The purple boundary is dotted. The white point of D50 is marked with an "x". The distance between the circles is 5nm.

A chromaticity diagram can be used to visualise the fact that not all colours can be reproduced on a computer monitor (see figure 2.8). It is surprising to see how small the colour gamut of the monitor is, compared to the spectral locus. The monitor is especially bad on saturated green colours.

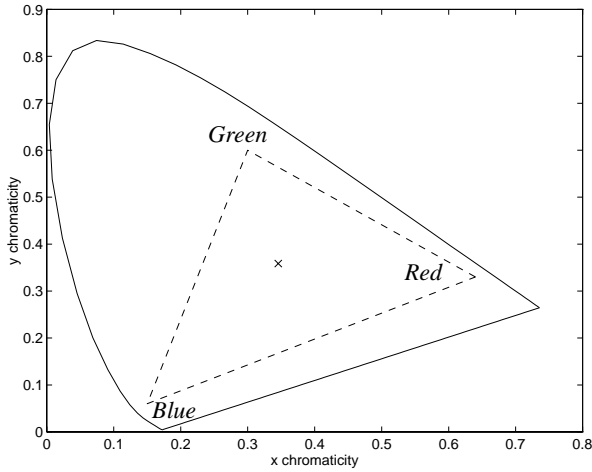


Figure 2.8 The colour gamut of a typical computer monitor. The corners of the triangle are the chromaticity coordinates for red, green, and blue phosphor.

CIELAB and CIELUV

The CIE 1976 ($L^* a^* b^*$) colour space, usually written CIELAB, is derived from XYZ coordinates with aim on perceptual uniformity. Originally it was developed to give the textile industry an accurate way to describe colours. Now it serves as one of the most well known device independent colour spaces, for all kinds of applications.

The transformation between XYZ values and CIELAB values is defined by:

$$L^* = \begin{cases} 116 \cdot \left(\frac{Y}{Y_n}\right)^{1/3} - 16, & \left(\frac{Y}{Y_n}\right) > 0,008856 \\ 903,3 \cdot \left(\frac{Y}{Y_n}\right), & \left(\frac{Y}{Y_n}\right) \leq 0,008856 \end{cases} \quad [\text{EQ 2.18}]$$

$$a^* = 500 \cdot \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad [\text{EQ 2.19}]$$

$$b^* = 200 \cdot \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \quad [\text{EQ 2.20}]$$

$$f(x) = \begin{cases} x^{1/3}, & x > 0,008856 \\ 7,787x + \frac{16}{116}, & x \leq 0,008856 \end{cases} \quad [\text{EQ 2.21}]$$

The constants X_n , Y_n , and Z_n are the XYZ values for the chosen reference white point. When working with colour monitors good choices could be something close to D65's.

How shall the coordinates in CIELAB be interpreted? Figure figure 2.9 gives a hint. L^* is the lightness, while a^* is (approximately) corresponding to the greenness/redness and b^* is (also approximately) the yellowness/blueness.

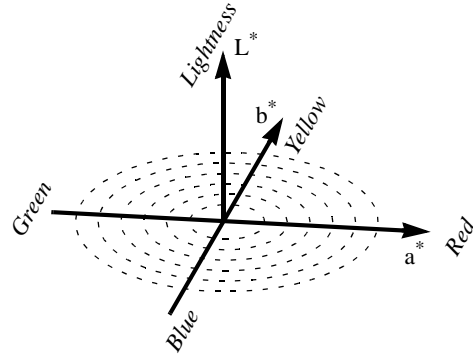


Figure 2.9 The CIELAB colour space.

In addition to $L^* a^* b^*$ coordinates there is also another device independent colour space called $L^* u^* v^*$ coordinates, defined by slightly modified equations (L^* is the same, however). CIELAB works best in subtractive colour applications such as paper printing, while CIELUV works better for additive applications such as computer monitors and TVs. Additive and subtractive colour mixing will be discussed in section 2.5.

Hue and Saturation based spaces

The colour spaces described earlier, although practical to use in technical applications, are not very similar to the human brain's way of classifying colours. Rather, terms like hue (H), saturation (S), lightness (L), brightness (B), intensity (I), or value (V), becomes interesting for these purposes. These characteristics could be grouped three by three to form "humanistic colour spaces". Usual combinations are *HSB*, *HSL*, *HSI*, and *HSV*. The definitions of the important properties are as follows:

- **Hue**—the attribute of a visual sensation whether an area appears to be similar to one of the perceived colours red, yellow, green, and blue, or a combination of two of them. The hue is roughly the colour of the dominant wavelength in the SPD.
- **Saturation**—the colourfulness of an area, judged in proportion to its brightness. The more an SPD is concentrated to one wavelength, the more saturation will be associated with the colour. A colour can be desaturated by adding light that contains power at all wavelengths.
- **Lightness**—the non-linear perceptual response to brightness or luminance (the subjective and objective measure of the same thing), defined as L^* in the previous section.
- **Brightness, intensity, or value**—alternative technical definitions of lightness.

The exact mathematical conversions from *RGB* to (for example) *HSV* are not very interesting, since the hue and saturation based colour spaces very rarely are used in any practical or technical applications.

2.5 NEUGEBAUER'S EQUATIONS

When colour tones are to be reproduced, usually a set of basis colours are used. For example, the picture on a TV screen is formed by dots of the basis colours *red*, *green*, and *blue*. When all the three colours are present, white colour is perceived, and when none of them is present, black is perceived. This is called *additive* colour mixing and is illustrated in figure 2.10. The spectral distributions of the lights are added:

$$I_{tot}(\lambda) = I_1(\lambda) + I_2(\lambda) \quad \text{[EQ 2.22]}$$

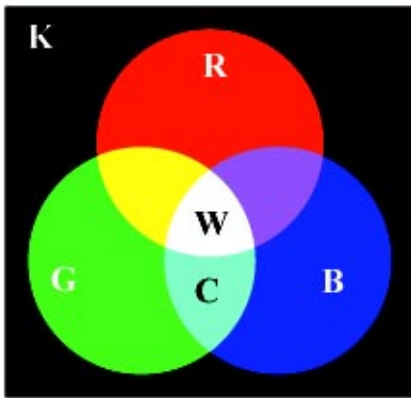


Figure 2.10 Additive colour mixing.

Unlike the TV, a piece of paper does not add energy to the illuminating light. The printed inks can be seen as filters which absorb some of the incoming light before it is reflected (see equation 2.23). The more colours printed on the same spot, the more filters will subtract energy at certain wavelengths from the illumination. This is called *subtractive* colour mixing and is illustrated in figure 2.11, although the term *multiplicative* colour mixing would be more mathematically correct:

$$I_{tot}(\lambda) = R_1(\lambda) \cdot R_2(\lambda) \cdot I(\lambda) \quad \text{[EQ 2.23]}$$

where R_1 and R_2 are the reflectance functions of the two colours. In subtractive colour applications, the basis colours *cyan*, *magenta*, and *yellow*, are usually used. When all the three colours are present, most of the light is absorbed and black is perceived.

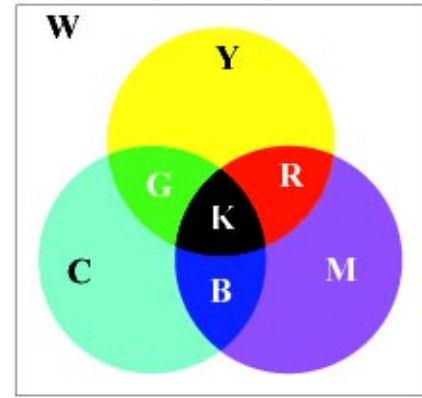


Figure 2.11 Subtractive colour mixing.

When printing colour images, four colours are usually used. The three basis colours above (*CMY*), and pure *black* ink (*K*). The reason for using black, despite the fact that it can be produced by the other three colours, is practical. There are three main reasons:

- It is cheaper to print large black areas with black ink than with cyan, magenta, and yellow ink. Also, one layer of ink dries faster than three, and smears less in the printing process.
- There is a large risk that small black letters etc. will be a bit blurred if printed in three colours, due to misregistration between the printing plates corresponding to the printing colours.
- Introducing the fourth colour, the shadowed areas of images can be printed with more colour depth, and a richer blackness. The colour gamut is expanded.

Neugebauer's equations deal with an alternative additive colour mixing, which arises when several small different coloured areas are averaged together by the human eye. The simplest form of the equations are:

$$\begin{bmatrix} X_{tot} \\ Y_{tot} \\ Z_{tot} \end{bmatrix} = \sum_i a_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad \text{[EQ 2.24]}$$

where a_i is the fractional area covered with the colour (X_i, Y_i, Z_i), and thus:

$$\sum_i a_i = 1 \quad \text{[EQ 2.25]}$$

When printing with four inks as described above it is possible to create 16 different colours (of which 9 are black), so the sum in equation 2.24 will have 16 terms. Neugebauer's equations may also be used in other applications where there are many small different coloured areas next to each other.

2.6 APPLICATIONS

One interesting application of the colour science is colour halftoning. That is, how to approximate a continuous tone image with one or several binary images (one for each ink colour).

When two different coloured inks are printed on top of each other, there is a typical subtractive (multiplicative) colour mixing. Let $R_p(\lambda)$ be the paper’s reflectance function. Let $C(\lambda)$, $M(\lambda)$, $Y(\lambda)$, and $K(\lambda)$ be the absorption filters of the four inks, and let $I(\lambda)$ be the illuminant of the total area. Then the blue colour (cyan and magenta on top of each other) will have the spectrum:

$$I_{blue}(\lambda) = R_p(\lambda)C(\lambda)M(\lambda)I(\lambda) \quad \text{[EQ 2.26]}$$

This should be interpreted as that both the paper and the cyan and magenta inks remove energy of certain wavelengths from the incoming light. By using this in the tristimulus equations, X, Y, and Z values can be calculated for the blue colour. The same can be done for all the 16 colours that are possible when printing with four inks. They may then be used in the Neugebauer equations, defined in the previous section.

We may know the fractional coverage for each of the printing colours separately, but how do we know for example the fractional area covered with both cyan and magenta inks? Won’t it differ a lot if one of the printing colours is translated a bit (misregistration). Yes, it would if all printing colours had the same screen angles. Then only a small translation of one of the printing plates would yield a large colour shift. This is one of the main reasons why the printing industry does not print all inks with the same screen angle (see figure 2.12). If different angles are used, a resistance from colour shifts due to misregistration is achieved—approximately the same fractional coverages remain despite translation.

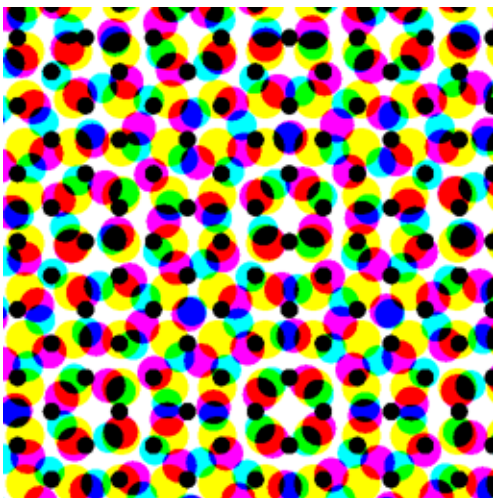


Figure 2.12 A colour halftone with 30% cyan, 40% magenta, 50% yellow, and 10% black.

The approximation of the fractional coverage is calculated in the following way, assuming a semi-stochastic overlap behaviour, and known as the Demichel equations:

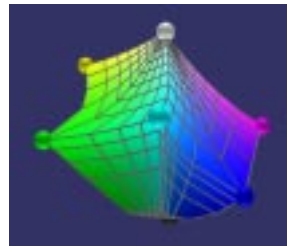
Let c , m , y , and k be the fractional areas covered with cyan, magenta, yellow and black ink, respectively. The fractional area not covered with each particular ink is $1 - x$, where x represents one of the four letters above. Then the fractional area covered with cyan and magenta, but not yellow or black (yielding blue) is:

$$a_{cm} = cm(1 - y)(1 - k) \quad \text{[EQ 2.27]}$$

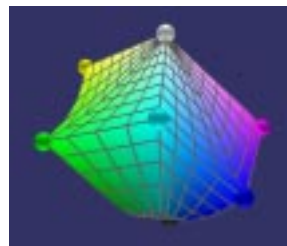
In the same manner, for example the area solely covered by yellow is:

$$a_y = (1 - c)(1 - m)y(1 - k) \quad \text{[EQ 2.28]}$$

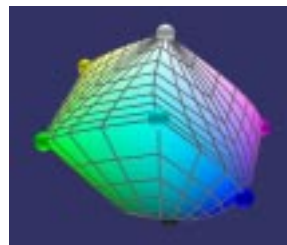
In this way 16 different fractional areas can be calculated for the current CMYK colour. These values can then be used in the Neugebauer equations, to calculate the colour perceived by the human eye. The accuracy of the values calculated above depends on halftone characteristics, such as dot shape and screen angles.



No dot gain



Extreme optical dot gain



Extreme physical dot gain

Figure 2.13 Colour gamut expansion due to dot gain. Plots made by Stefan Gustavson.

If the printing process was ideal, the model described above would be sufficient. However, in real printing the shapes of the halftone dots are distorted and the concept of dot gain is introduced, yielding a darker image. The

compensation for the dot gain has always been a large problem for the graphical industry. But the dot gain has recently proved to have another surprising aspect. It enlarges the colour gamut of reproducible colours. This is illustrated in figure 2.13. If the dot gain characteristics are known or can be predicted, dot gain can be used to enhance the reproduction of colour images. Research efforts are currently being made in this area.

2.7 FINAL WORDS

Colour science is quite tough to master. There are lots of new terms to learn, and the number of available colour spaces is huge. Hopefully this text has produced some understanding for the reader, making it possible to perform the exercises in the next chapter.

The interested (or confused...) reader should refer to the literature listed under references. Especially the books by Field and Hunt are recommended.

GLOSSARY

- *Colour gamut.* Färgomfång.
- *Cones.* Tappar (på näthinnan).
- *Continuous tone image.* Halvtonsbild. Bild med kontinuerligt varierande (så när som på fin kvantisering) färg eller gråskala.
- *Distribution.* Fördelning.
- *Fovea.* Gula fläcken.
- *Halftone image.* Rasterbild.
- *Halftoning.* Rastrering.
- *Hue.* Färgnyans eller kulör.
- *Misregistration.* De olika tryckfärgernas tryckplåtar råkar hamna snett i förhållande till varandra under tryckprocessen.
- *Retina.* Näthinnan.
- *Rods.* Stavar (på näthinnan).

- *Saturation.* Färgmättnad.
- *Screen.* Här: rastermönster.
- *Screen angle.* Rastervinkel. Normalt definierad i intervallet [0,90[.

REFERENCES

“*Color And Its Reproduction*”

by **Gary G. Field**

Graphic Arts Technical Foundation, 1988.

“*Measuring Colour (second edition)*”

by **Robert W.G. Hunt**

Ellis Horwood, 1991. ISBN 0-13-567686-X.

“*Color Spaces FAQ*”

by **David Bourgin**

Report from Internet, found at

<http://www.dcs.ed.ac.uk/%7Emxr/gfx/faqs/colourspace.faq>

“*Frequently Asked Questions about Colour*”

by **Charles A. Poynton**

Report from Internet, found at

<ftp://ftp.inforamp.net/pub/users/poynton/doc/colour/>

“*Color Calibration of Digital Devices*”

by **Anders Gustafsson**

Master's thesis, Department of Electrical Engineering, Image Processing Group, Linköping University and Institute of Technology. LiTH-ISY-EX-1682. 20/9/96.

“*The Colour Gamut of Halftone Reproduction*”

by **Stefan Gustavson**

Proceedings of the Fourth IS&T/SID Color Imaging Conference, November 19–22, 1996, pp. 80–85. ISBN 0-89208-196-1.

“*Inverse Halftoning of Scanned Colour Images*”

by **Jörgen Rydenius**

Master's thesis, Department of Electrical Engineering, Image Processing Laboratory, Linköping University and Institute of Technology. LiTH-ISY-EX-1713. 31/1/97.

<http://www.isy.liu.se/~jorry/xjobb.html>

Chapter 3.

EXERCISES

The first thing to do is to make sure that the right settings have been made in course tool. If they are correct, the Matlab environment, with proper settings, should be possible to start from the Application Manager, which is available on the CDE front panel. Do so!

3.1 COMPARING INTUITIVENESS

The first exercise is to compare the intuitiveness of two different colour spaces, in this case *RGB* and *HSV*. Start the application by executing the function *sliders* in Matlab:

```
> sliders
```

The window shown in figure 3.1 will be visible.



Figure 3.1 The “sliders” window.

When the button called *New* is pressed, a new randomly picked reference colour is shown in the centre. Then you shall try to create the reference colour in two different ways by using the sliders. When you are satisfied with your colour mixing, you can display the correct answer by clicking *Show RGB* or *Show HSV*. Try with a couple of different reference colours and notice the time it takes to come close to the reference colour, and the time it takes to make a perfect match, for both of the two colour spaces.

Which of the two colour spaces is the most intuitive to use? Which is faster to work in?

Why do you think that is the case?

3.2 WORKING WITH SPECTRAL POWER DISTRIBUTIONS

Next we shall examine how to calculate tristimulus values from a spectral power distribution. Also metamerism is going to be visualised.

Load some necessary variables available in the file *spectra.mat* and show their variable names:

```
> load spectra  
> who
```

Plot the *xyz* colour matching functions with Matlab command:

```
> plot(xyz(:,1),xyz(:,2:4))
```

Check that their appearance on the screen is similar to figure 2.6. Use the standard illuminant D65 and prepare for tristimulus calculation, by calculating the factor k according to equation 2.9. The $\bar{y}(\lambda)$ function is the third column of *xyz*.

Calculate the white point of D65, by using equations 2.6 to 2.9. What values do you receive, and do they agree with the values in section 1.4?

If they do not match in all decimals try to explain why they differ. A hint is to study the values of D65 and xyz in both ends of the spectrum.

Now plot the two reflectance functions $R1$ and $R2$ on the screen. As you can see they differ quite a lot. Compare with the spectrum in figure 2.1 and try to predict the colours of the two objects. Calculate the XYZ tristimulus values of the two objects, both illuminated with D65.

R1:

R2:

Now we are going to change the illumination to a fluorescent lamp, representative of *the three-band type*. The emission is mainly concentrated to three narrow bands in the spectrum, designed to occur around wavelengths of approximately 435, 545, and 610nm. This kind of lamp tends to increase the saturation of most colours, making it attractive for some purposes, such as lighting goods in stores. However, the appearance of some colours can be somewhat distorted, so it is less suitable for critical evaluation of colours in general.

Change the illumination to F11 (the CIE code for the lamp described above) and once again calculate the tristimulus values.

R1:

R2:

What has happened, and what is the scientific term for the phenomenon?

Convert the four XYZ values to RGB values of the screen by using the transformation described in equations 2.13 and 2.14. RGB values shall be in the range of zero to one. What has occurred if some of the transformed values are outside this range? Refer to figure 2.8.

Create a colour map of the four calculated values, project the values into the monitor's colour gamut, perform gamma correction (here set to 1.4), and create an image which displays the four colours:

```
> map=[r1 g1 b1;r2 g2 b2 ...];
> map=min(1,max(0,map));
> map=map.^(1/1.4);
> image([1 2;3 4])
> colormap(map)
```

Move the mouse pointer to the figure window to ensure that the colours are shown right. Explain in what way the colours changed when switching to F11. What has happened to the white point in the xy chromaticity diagram when changing the light source? Towards which hue has it moved? Does that show in the colours?

3.3 ON-DOT AND OFF-DOT HALFTONING

To know the exact colour of an area of a halftone image, it is important to know if the halftone dots of the different inks are mainly on top of each other, or next to each other. There is a large difference between the two extreme cases—maximum overlap or minimum overlap. Let us now consider two printing colours: cyan and magenta. We wish to create a 50% blue. As blue is constructed by both cyan and magenta, it is natural to use 50% cyan coverage and 50% magenta coverage. Then the two extreme cases are (see also figure 3.2):

- The two inks do not overlap at all, leaving no unprinted area. This is referred to as off-dot printing.
- The two inks are completely on top of each other, leaving 50% of the paper unprinted. This is referred to as on-dot printing.

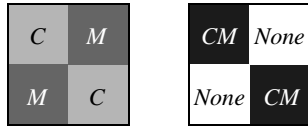


Figure 3.2 The two extreme cases when creating blue by using 50% cyan and 50% magenta. Left: off-dot, and right: on-dot.

The following values have been measured, using the white point corresponding to D65:

Ink	L^*	a^*	b^*
None	93.45	1.15	-3.40
Cyan	71.23	-52.10	-30.73
Magenta	50.02	81.70	-12.62
Cyan & Mag.	28.74	20.63	-66.53

Convert these values to XYZ tristimulus values by inverting equations 2.18 to 2.21. This can be done by using the function `lab2xyz` (see on-line help for parameter information).

Ink	X	Y	Z
None			
Cyan			
Magenta			
Cyan & Mag.			

Now use these values in Neugebauer's equations and calculate the perceived colours in the two extreme cases described earlier:

Ink	X	Y	Z
On-dot			
Off-dot			

Convert the six XYZ values to RGB , create a gamma corrected colour map, and show an image with these six colours, in the same manner as described earlier. Is there a noticeable difference between on-dot and off-dot? Is it very obvious?

To avoid this sensitivity to misregistration, different screen angles are used for each of the four printing colours. This way, the global colour shift due to misregistration is decreased. The effectiveness of this will be examined in the next section.

3.4 COLOUR HALFTONING ACCORDING TO DEMICHEL

Load the *CMYK*-image called *halftone.tif*¹:

```
> [C,M,Y,K]=tiffread('halftone');
```

Use your pre-written function to convert $c, m, y,$ and k to an *RGB* image, and display it:

```
> [R,G,B]=cmyk2rgb(C,M,Y,K);
> imshow(R,G,B)
```

Each of the four colour channels can be displayed by for example:

```
> shgr(C)
```

The image is a halftoned version of the colour constructed by: 30% cyan, 40% magenta, 50% yellow, and 10% black. The four binary matrices have the size 512x512 pixels. Because they are binary, logical operations may be performed on them. This fact can be used when calculating the fractional area covered with certain inks. For example, to calculate the fractional area covered with cyan and magenta, but not yellow or black is:

```
> sum(sum(C & M & ~Y & ~K))/512^2
```

We are not going to use the whole image. Rather, we are going to choose a centred partition of size 256x256, for example:

```
> C1=C(129:384,129:384);
```

Then calculate the fractional coverages as described above (do not forget to divide by 256 to the power of two instead of 512), for the sixteen possible ink combinations in four-colour printing. Write the results in the table below in the column marked *test 1*. Try also to simulate misregistration by translating the choice of partition by ten to twenty positions in any direction, for one or several of the four matrices. Once more calculate the fractional coverages with the new matrices and write in the column marked *test 2*. *It could be useful to write a function that takes four binary matrices as input, and automatically calculate the sixteen searched values.*

Ink	Demichel	Test 1	Test 2
None			
Only C			

1. The standard version of *tiffread* does not allow 32 bit *CMYK* format. However, the file available here is modified.

<i>Ink</i>	<i>Demichel</i>	<i>Test 1</i>	<i>Test 2</i>
Only M			
Only Y			
Only K			
Only C & M			
Only C & Y			
Only C & K			
Only M & Y			
Only M & K			
Only Y & K			
C, M & Y			
C, M & K			
C, Y & K			
M, Y & K			
C, M, Y & K			

Compare the three columns. Are Demichel's equations a good model of the reality? Does it work reasonably well even when misregistration occurs?

What would have happened in case of misregistration if all the four printing colours had had the same screen angle?

3.5 COLOUR ADJUSTMENT IN CIELAB

As a last exercise we are going to study how to adjust some colour attributes of a CIELAB image, such as brightness, contrast, hue, and saturation. This kind of colour manipulation is commonly performed in image processing software like Photoshop. Here we are going to perform these manipulations mathematically in Matlab, using the CIELAB colour space. Load the image called *butterfly.tif*:

```
> [R,G,B]=tiffread('butterfly');
> imshow(R,G,B)
```

Matlab's colour reproduction of full colour images on the 256 colour screens available for this exercise is not very reliable, but it is enough for our purposes here. Always remember to move the mouse pointer to the figure window to ensure that the right colour palette is used. Better colour reproduction can sometimes be achieved simply by saving the manipulated image from Matlab using the *tiffwrite* function, and then view the image with an external program such as *xv*.

When manipulations are made in *cielab*, one has to remember that the colour gamut of the monitor limits what is possible to display. As an example: if colours are made extremely saturated, then most of the colours will have to be projected back into the monitor's colour gamut to be displayed, yielding a distortion.

Begin by converting the *RGB* image to *CMYK*, by using your pre-written *rgb2cmyk* function. Display them simultaneously. This is the kind of monochromic images that will be individually halftoned with different screen angels.

```
> [C,M,Y,K]=rgb2cmyk(R,G,B);
> figure(1)
> shgr(C)
> figure(2)
> shgr(M)
> figure(3)
> shgr(Y)
> figure(4)
> shgr(K)
```

The images seem inverted. Why is that the case? Refer to how the images will be used in the printing and halftoning processes.

Convert the image from *RGB* to CIELAB.

```
> [L,a,b]=rgb2lab(R,G,B);
```

Test what happens if the values of *L* are translated up or down.

```
> [R2,G2,B2]=lab2rgb(L+20,a,b);
> imshow(R2,G2,B2)
> [R2,G2,B2]=lab2rgb(L-20,a,b);
> imshow(R2,G2,B2)
```

What happens? Compare to the original image.

Scale the lightness with a fixed point around 60 (which is close to the mean lightness value of this image). What happens? What is the technical term for this?

```
> [R2,G2,B2]=lab2rgb((L-60)*2+60,a,b);
> [R2,G2,B2]=lab2rgb((L-60)*0.8+60,a,b);
```

Change sign for all a^* -values. Compare to figure 2.9. Then set all a^* -values to zero and predict what will happen. Were you right? What attribute of the colour do we change when switching sign of a^* or b^* ?

```
> [R2,G2,B2]=lab2rgb(L,-a,b);
> [R2,G2,B2]=lab2rgb(L,0,b);
```

Now, scale the a^* - and b^* -values. What colour attribute is now affected?

```
> [R2,G2,B2]=lab2rgb(L,0.5*a,0.5*b);
```

```
> [R2,G2,B2]=lab2rgb(L,3*a,3*b);
```

Finally, we are going to convert our colour image to a gray scale image. This is done by simply using the Y tristimulus value for all pixels in the image. It can be derived from the RGB values by using the same matrix as we have done before. Now we use the function *rgb2xyz* to do the work instead:

```
> [X,Y,Z]=rgb2xyz(R,G,B);
> grayim=Y/100;
> shgr(im,map)
```

Try to enhance the lightness and contrast levels of the gray scale image. Which operations produce good results?

If there is time left: feel free to play around with different image manipulations, such as scaling and translation, in different colour spaces. Try to predict the result before displaying the image!

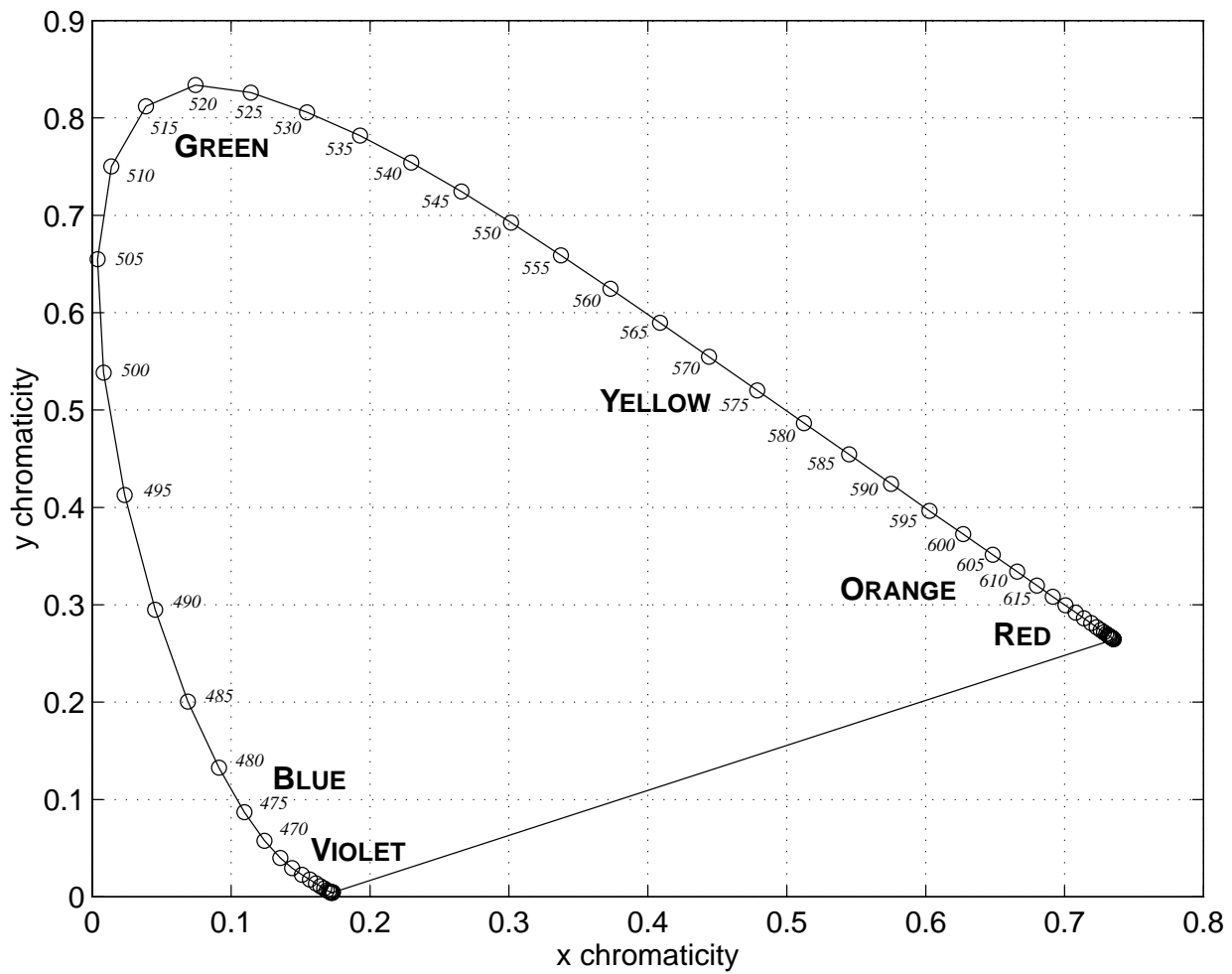


Figure 3.3 Plot your calculated white points in this chromaticity diagram.