Chapter 4

COLOUR

The experience of colour is caused by the vision system responding differently to different wavelengths of light. As a result, it is important to deal with radiance and with reflection at a variety of wavelengths. These quantities are not colour, however, and much of this chapter will cover different ways of describing colours.

We will first extend our radiometric vocabulary to describe energy arriving in different quantities at different wavelengths; we then study human colour perception, which will yield methods for describing colours;

4.1 Spectral Quantities

All of the physical units we have described can be extended with the phrase "per unit wavelength" to yield **spectral units**. These allow us to describe differences in energy, in BRDF or in albedo with wavelength. We will ignore interactions where energy changes wavelength; thus, the definitions of Chapter 2 can be extended by adding the phrase "per unit wavelength," to obtain what are known as **spectral quantities**.

Spectral radiance is usually written as $L^{\lambda}(\boldsymbol{x}, \theta, \phi)$, and the radiance emitted in the range of wavelengths $[\lambda, \lambda + d\lambda]$ is $L^{\lambda}(\boldsymbol{x}, \theta, \phi)d\lambda$. Spectral radiance has units Watts per cubic meter per steradian ($Wm^{-3}sr^{-1}$ — cubic meters because of the additional factor of the wavelength). For problems where the angular distribution of the source is unimportant, **spectral exitance** is the appropriate property; spectral exitance has units Wm^{-3}.

Similarly, the **spectral BRDF** is obtained by considering the ratio of the spectral radiance in the outgoing direction to the **spectral irradiance** in the incident direction. Because the BRDF is defined by a ratio, the spectral BRDF will again have units sr^{-1}.

4.1.1 The Colour of Surfaces

The colour of coloured surfaces is a result of a large variety of mechanisms, including differential absorbtion at different wavelengths, refraction, diffraction and bulk scattering (for more details, see, for example []). Usually these effects are bundled into a macroscopic BRDF model, which is typically a Lambertian plus specular approximation; the terms are now **spectral reflectance** (sometimes abbreviated to **reflectance**) or (less commonly) **spectral albedo**. Figure 4.1 shows examples of spectral reflectances for a number of different coloured pigments.

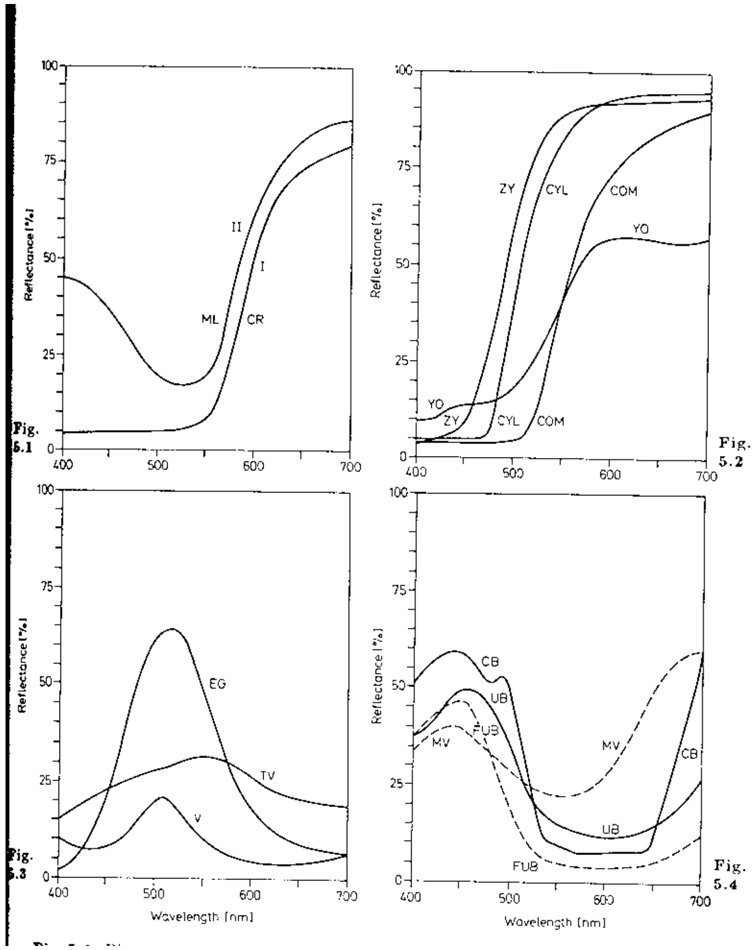

Figure 4.1. Spectral albedos from a variety of pigments. **Key:** CR — cadmium red; ML — madder lake; COM — cadmium orange medium; CYL — cadmium yellow light; ZY — zinc yellow; YO — yellow ochre; EG — emerald green; TV— terre verte; V — viridian; CB — cobalt blue; UB — ultramarine blue; FUB — French ultramarine blue. (Taken in the fervent hope of receiving permission from Agoston, Color Theory and its application in art and design, p. 31)

The colour of the light returned to the eye is affected both by the spectral radiance (colour!) of the illuminant and by the spectral reflectance (colour!) of the surface. If we use the Lambertian plus specular model, we have:

$$E(\lambda) = \rho_{dh}(\lambda)S(\lambda) \times \text{geometric terms} + \text{specular terms}$$

where $E(\lambda)$ is the spectral radiosity of the surface, $\rho_{dh}(\lambda)$ is the spectral reflectance and $S(\lambda)$ is the spectral irradiance. The specular terms have different colours depending on the surface — i.e. we now need a **spectral specular albedo**.

Generally, metal surfaces have a specular component that is wavelength dependent — a shiny copper penny has a yellowish glint. Surfaces that do not conduct — **dielectric surfaces** — have a specular component that is independent of wavelength — for example, the specularities on a shiny plastic object are the colour of the light. Section **??** describes how these properties can be used to find specularities, and to find image regions corresponding to metal or plastic objects.

4.1.2 The Colour of Natural Sources

The most important natural light source is the sun. The sun is usually modelled as a distant, bright point. The colour of sunlight varies with time of day and time of year. These effects have been widely studied, and figure 4.2 gives some models of the sun's spectral radiance for different latitudes and times of year.

The sky is another important natural light source. A crude geometrical model is as a hemisphere with constant exitance. The assumption that exitance is constant is poor, however, because the sky is substantially brighter at the horizon than at the zenith. The sky is bright because light from the sun is scattered by the air. The natural model is to consider air as emitting a constant amount of light per unit volume; this means that the sky is brighter on the horizon than at the zenith, because a viewing ray along the horizon passes through more sky.

For clear air, the average distance that light travels before being scattered depends on the fourth power of wavelength (this is known as Rayleigh scattering). This means that, when the sun is high in the sky, blue light is scattered out of the ray from the sun to the earth — meaning that the sun looks yellow — and can scatter from the sky into the eye — meaning that the sky looks blue. There are standard models of the spectral radiance of the sky at different times of day and latitude, too. Surprising effects occur when there are fine particles of dust in the sky (the larger particles cause very much more complex scattering effects, usually modelled rather roughly by the Mie scattering model []) — one author remembers vivid sunsets in Johannesburg caused by dust in the air from mine dumps, and there are records of blue and even green moons caused by volcanic dust in the air.

4.1.3 Artificial Sources

Typical artificial light sources are of a small number of types.

- **Incandescent lights** work by heating metal elements to a high temperature. The spectrum roughly follows the black-body law, meaning that incandescent

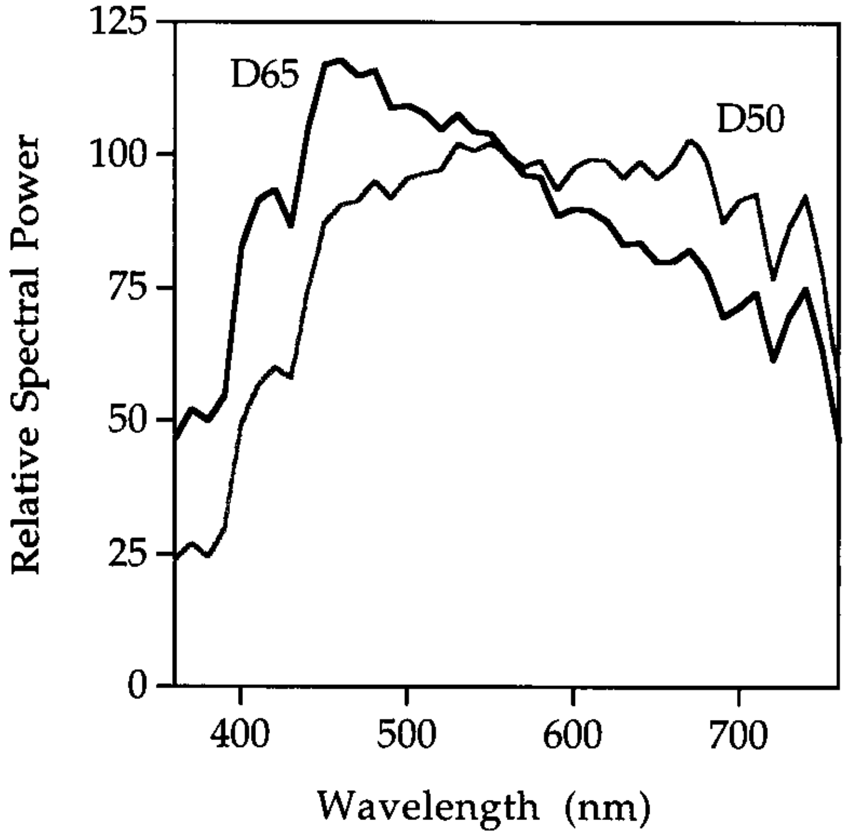

Figure 4.2. There is a variety of different models of daylight; the graph shows the relative spectral power distribution of two standard CIE models of daylight, D-50 and D-65. These represent "industry standard" daylights under different circumstances; `which for when` *(Taken in the fervent hope of receiving permission from Fairchild, Color Appearance Models, p. 69)*

lights in most practical cases have a reddish tinge (Figure 4.8 shows the locus of colours available from the black-body law at different temperatures).

- **Fluorescent lights** work by generating high speed electrons that strike phosphors on the inside of the bulb. These phosphors fluoresce. Typically the coating consists of three or four phosphors, which fluoresce in quite narrow ranges of wavelengths. Most fluorescent bulbs generate light with a bluish tinge, but bulbs that mimic natural daylight are increasingly available.

- **Metal halide lights** work by `how?`

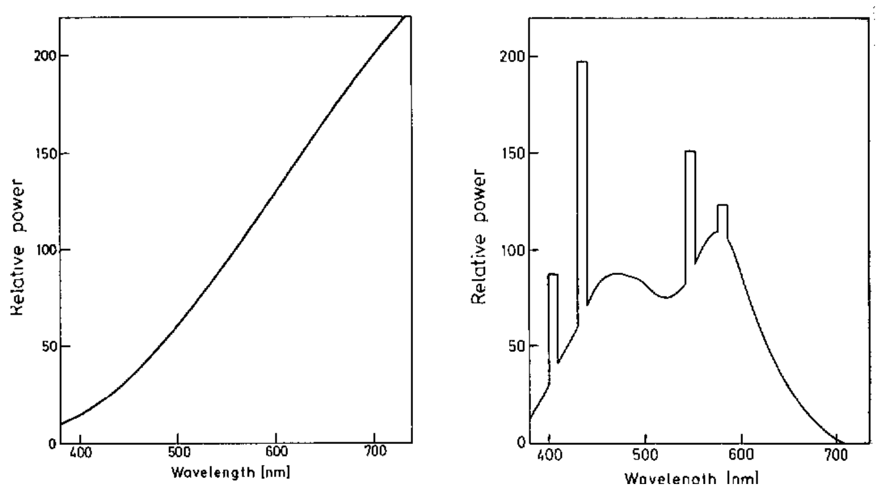

Figure 4.3. On the left, the relative spectral power distribution of light from a tungsten filament lamp with a colour temperature of 2856K. On the right, the relative spectral power distribution of light from a daylight flourescent lamp — note the bright, narrow bands that come from the flourescing phosphors. On the right, we see the relative spectral power distribution of several different standard CIE models of flourescent lamps. (Taken in the fervent hope of receiving permission from Agoston, Color Theory and its application in art and design, p. 22)

- **Mercury vapour lights** work by `how?`
- **Halogen lights** work by `how?`

Figure 4.3 shows a sample of spectra from different light bulbs.

4.2 Human Colour Perception

To be able to describe colours, we need to know how people respond to them. Human perception of colour is a complex function of context; illumination, memory, object identity and emotion can all play a part. The simplest question is to understand which spectral radiances produce the same response from people under simple viewing conditions. This yields a simple, linear theory of colour matching which is accurate and extremely useful for describing colours (section 4.2.1). We sketch the mechanisms underlying the transduction of colour in section 4.2.2.

4.2.1 Colour Matching

In a typical experiment a subject sees a coloured light — the **test light** — in one half of a split field. The subject can then adjust a mixture of lights in the other half

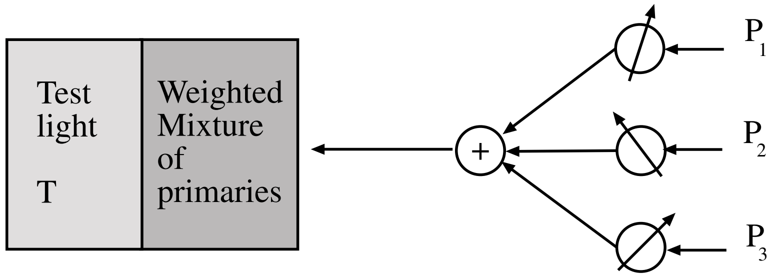

Figure 4.4. Human perception of colour can be studied by asking observers to mix coloured lights to match a test light, shown in a split field. The drawing shows the outline of such an experiment. The observer sees a test light T, and can adjust the amount of each of three primaries in a mixture that is displayed next to the test light. The observer is asked to adjust the amounts so that the mixture looks the same as the test light. The mixture of primaries can be written as $w_1P_1+w_2P_2+w_3P_3$; if the mixture matches the test light, then we write $T = w_1P_1+w_2P_2+w_3P_3$. It is a remarkable fact that for most people three primaries are sufficient to achieve a match for many colours, and for all colours if we allow subtractive matching (i.e. some amount of some of the primaries is mixed with the test light to achieve a match). Some people will require fewer primaries. Furthermore, most people will choose the same mixture weights to match a given test light.

to get it to match. The adjustments involve changing the intensity of some fixed number of **primaries** in the mixture. In this form, a large number of lights may be required to obtain a match, but many different adjustments may yield a match.

We can obtain an algebraic expression, where T denotes the test light, an equals sign denotes a match, and the weights w_i are the intensities of the primaries P_i (and so are non-negative). A match is written in an algebraic form as:

$$T = w_1P_1 + w_2P_2 + \ldots$$

The situation is simplified if **subtractive matching** is allowed: in subtractive matching, the viewer can add some amount of some primaries to the test light instead of to the match. This can be written in algebraic form by allowing the weights in the expression above to be negative.

Trichromacy

It is a matter of experimental fact that for most observers only three primaries are required to match a test light. There are some caveats. Firstly, subtractive matching must be allowed, and secondly, the primaries must be independent — meaning that no mixture of two of the primaries may match a third. This phenomenon is known as the the principle of **trichromacy**. It is often explained by assuming that there are three distinct types of colour transducer in the eye; recently, evidence has emerged from genetic studies to support this view [**?**].

It is a remarkable fact that, given the same primaries and the same test light, most observers will select the same mixture of primaries to match that test light. This phenomenon is usually explained by assuming that the three distinct types of colour transducer are common to most people. Again, there is now some direct evidence from genetic studies to support this view [**?**].

Grassman's Laws

It is a matter of experimental fact that matching is (to a very accurate approximation) linear. This yields **Grassman's laws**.

Firstly, if we mix two test lights, then mixing the matches will match the result, that is, if

$$T_a = w_{a1}P_1 + w_{a2}P_2 + w_{a3}P_3$$

and

$$T_b = w_{b1}P_1 + w_{b2}P_2 + w_{b3}P_3$$

then

$$T_a + T_b = (w_{a1} + w_{b1})P_1 + (w_{a2} + w_{b2})P_2 + (w_{a3} + w_{b3})P_3$$

Secondly, if two test lights can be matched with the same set of weights, then they will match each other, that is, if

$$T_a = w_1P_1 + w_2P_2 + w_3P_3$$

and

$$T_b = w_1P_1 + w_2P_2 + w_3P_3$$

then

$$T_a = T_b$$

Exceptions

Given the same test light and the same set of primaries, most people will use the same set of weights to match the test light. This, trichromacy and Grassman's laws are about as true as any law covering biological systems can be. The exceptions include:

- people with aberrant colour systems as a result of genetic ill-fortune (who may be able to match everything with fewer primaries);
- people with aberrant colour systems as a result of neural ill-fortune (who may display all sorts of effects, including a complete absence of the sensation of colour);
- some elderly people (whose choice of weights will differ from the norm, because of the development of macular pigment in the eye);
- very bright lights (whose hue and saturation look different from less bright versions of the same light);
- and very dark conditions (where the mechanism of colour transduction is somewhat different than in brighter conditions).

4.2.2 Colour Receptors

Trichromacy suggests that there are profound constraints on the way colour is transduced in the eye. One hypothesis that satisfactorily explains this phenomenon is to assume that there are three distinct types of receptor in the eye that mediate colour perception. Each of these receptors turns incident light into neural signals. It is possible to reason about the sensitivity of these receptors from colour matching experiments. If two test lights that have different spectra look the same, then they must have the same effect on these receptors.

The Principle of Univariance

The **principle of univariance** states that the activity of these receptors is of one kind — i.e. they respond strongly or weakly, but do not, for example, signal the wavelength of the light falling on them. Experimental evidence can be obtained by carefully dissecting light sensitive cells and measuring their responses to light at different wavelengths. Univariance is a powerful idea, because it gives us a good and simple model of human reaction to coloured light: two lights will match if they produce the same receptor responses, *whatever their spectral radiances.*

Because the system of matching is linear, the receptors must be linear. Let us write p_k for the response of the k'th receptor, $\sigma_k(\lambda)$ for its sensitivity, $E(\lambda)$ for the light arriving at the receptor and Λ for the range of visible wavelengths. We can obtain the overall response of a receptor by adding up the response to each separate wavelength in the incoming spectrum so that

$$p_k = \int_{\Lambda} \sigma_k(\lambda) E(\lambda) d\lambda$$

Rods and Cones

Anatomical investigation of the retina shows two types of cell that are sensitive to light, differentiated by their shape. The light sensitive region of a **cone** has a roughly conical shape, whereas that in a **rod** is roughly cylindrical. Cones largely dominate colour vision and completely dominate the fovea. Cones are somewhat less sensitive to light than rods are, meaning that in low light, colour vision is poor and it is impossible to read (one doesn't have sufficient spatial precision, because the fovea isn't working).

Studies of the genetics of colour vision support the idea that there are three types of cone, differentiated by their sensitivity The sensitivities of the three different kinds of receptor to different wavelengths can be obtained by comparing colour matching data for normal observers with colour matching data for observers lacking one type of cone. Sensitivities obtained in this fashion are shown in Figure 4.5. The three types of cone are properly called **S cones**, **M cones** and **L cones** (for their peak sensitivity being to short, medium and long wavelength light respectively). They are occasionally called blue, green and red cones; this is bad practice, because the sensation of red is definitely not caused by the stimulation of red cones, etc.

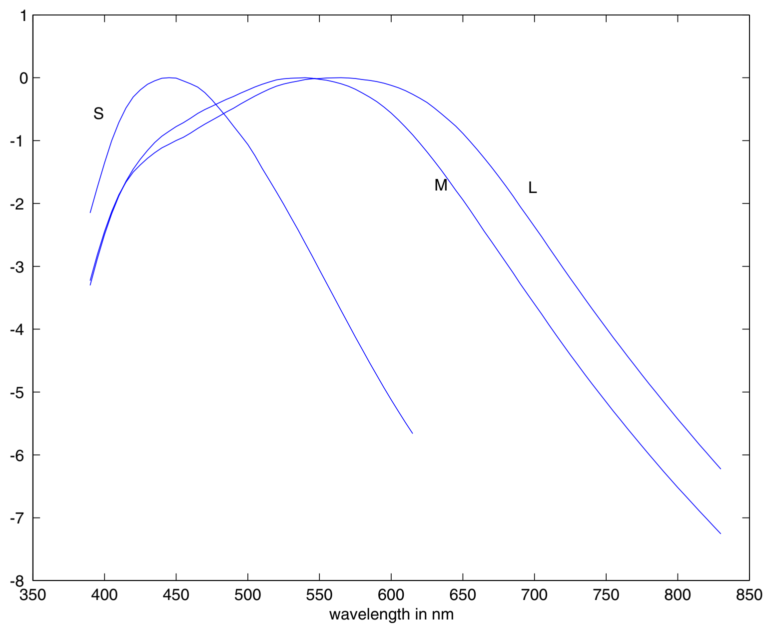

Figure 4.5. There are three types of colour receptor in the human eye, usually called cones. These receptors respond to all photons in the same way, but in different amounts. The figure shows the log of the relative spectral sensitivities of the three kinds of colour receptor in the human eye. The first two receptors —sometimes called the red and green cones respectively, but more properly named the long and medium wavelength receptors — have peak sensitivities at quite similar wavelengths. The third receptor has a very different peak sensitivity. The response of a receptor to incoming light can be obtained by summing the product of the sensitivity and the spectral radiance of the light, over all wavelengths.

4.3 Representing Colour

Describing colours accurately is a matter of great commercial importance. Many products are closely associated with very specific colours — for example, the golden arches or Kodak yellow — and manufacturers are willing to go to a great deal of trouble to ensure that different batches have the same colour. This requires a standard system for talking about colour. Simple names are insufficient, because relatively few people know many colour names, and most people are willing to associate a large variety of colours with a given name.

Colour matching data yields simple and and highly effective linear colour spaces (section 4.3.1). Specific applications may require colour spaces that emphasize particular properties (section 4.3.2) or uniform colour spaces, which capture the significance of colour differences (section 4.3.3).

4.3.1 Linear Colour Spaces

There is a natural mechanism for representing colour: first, agree on a standard set of primaries, and then describe any coloured light by the three values of the weights that people would use to match the light using those primaries. In principle, this is easy to use — to describe a colour, we set up and perform the matching experiment and transmit the match weights. Of course, this approach extends to give a representation for surface colours as well if we use a standard light for illuminating the surface (and if the surfaces are equally clean, etc.).

Colour Matching Functions

Performing a matching experiment each time we wish to describe a colour can be practical. For example, this is the technique used by paint stores; you take in a flake of paint, and they'll mix paint, adjusting the mixture until a match is obtained. Paint stores do this because pigments don't usually mix linearly. If a linear colour model applies, doing a large number of experimental matches is unnecessary (as well as extremely tedious).

Instead, we need an algorithm to determine which weights would be used to match this source of some known spectral radiance. The spectral radiance of the source can be thought of as a weighted sum of single wavelength sources. Because colour matching is linear, the combination of primaries that matches a weighted sum of single wavelength sources is obtained by matching the primaries to each of the single wavelength sources, and then adding up these match weights.

If we have a record of the weight of each primary required to match a single-wavelength source — a set of **colour matching functions** — we can obtain the weights used to match an arbitrary spectral radiance. The colour matching functions — which we shall write as $f_1(\lambda)$, $f_2(\lambda)$ and $f_3(\lambda)$ — can be obtained from a set of primaries P_1, P_2 and $P_3$by experiment. Essentially, we tune the weight of each primary to match a unit radiance source at every wavelength. We then obtain a set of weights, one for each wavelength, with the property

$$U(\lambda) = f_1(\lambda)P_1 + f_2(\lambda)P_2 + f_3(\lambda)P_3$$

where $U(\lambda)$ is a source of unit radiance at the single wavelength λ and we are again interpreting the equals sign to mean "matches".

The source — which we shall write $S(\lambda)$ — is a sum of a vast number of single wavelength sources, each with a different intensity. We now match the primaries to each of the single wavelength sources, and then add up these match weights, obtaining

$$S(\lambda) = \left\{ \int_\Lambda f_1(\lambda)S(\lambda)d\lambda \right\} P_1 + \left\{ \int_\Lambda f_2(\lambda)S(\lambda)d\lambda \right\} P_2 + \left\{ \int_\Lambda f_3(\lambda)S(\lambda)d\lambda \right\} P_3$$

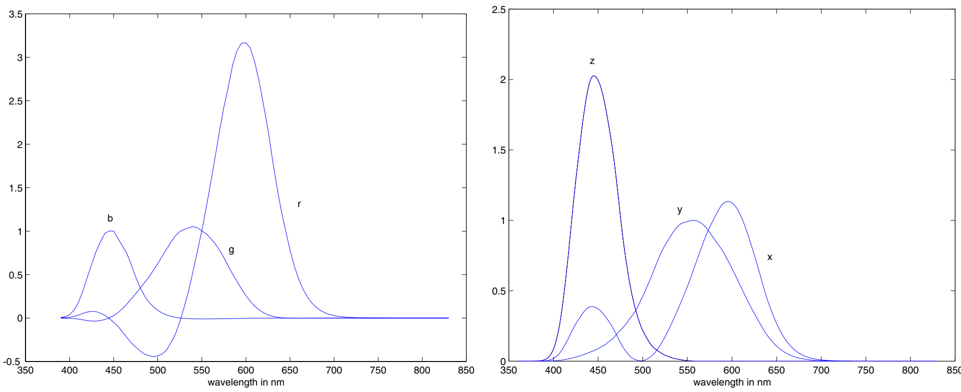

Figure 4.6. On the left, colour matching functions for the primaries for the RGB system. The negative values mean that subtractive matching is required to match lights at that wavelength with the RGB primaries. On the right, colour matching functions for the CIE X, Y and Z primaries; the colourmatching functions are everywhere positive, but the primaries are not real. Figures plotted from data available at `http://www-cvrl.ucsd.edu/index.htm`.

General Issues for Linear Colour Spaces

Linear colour naming systems can be obtained by specifying primaries — which imply colour matching functions – or by specifying colour matching functions — which imply primaries. It is an inconvenient fact of life that, if the primaries are real lights, at least one of the colour matching functions will be negative for some wavelengths. This is not a violation of natural law — it just implies that subtractive matching is required to match some lights, whatever set of primaries is used. It is a nuisance though.

One way to avoid this problem is to specify colour matching functions that are everywhere positive (which guarantees that the primaries are imaginary, because for some wavelengths their spectral radiance will be negative).

Although this looks like a problem — how would one create a real colour with imaginary primaries? — it isn't, because colour naming systems are hardly ever used that way. Usually, we would simply compare weights to tell whether colours are similar or not, and for that purpose it is enough to know the colour matching functions. A variety of different systems have been standardised by the CIE (the *commission international d'éclairage*, which exists to make standards on such things).

The CIE XYZ Colour Space

The **CIE XYZ colour space** is one quite popular standard. The colour matching functions were chosen to be everywhere positive, so that the coordinates of any real light are always positive. It is not possible to obtain CIE X, Y, or Z primaries because for some wavelengths the value of their spectral radiance is negative. However, given colour matching functions alone, one can specify the XYZ coordinates of a colour and hence describe it.

Linear colour spaces allow a number of useful graphical constructions which are more difficult to draw in three-dimensions than in two, so it is common to intersect the XYZ space with the plane $X + Y + Z = 1$ (as shown in Figure 4.7) and draw the resulting figure, using coordinates

$$(x, y) = \left(\frac{X}{X+Y+Z}, \frac{Y}{X+Y+Z}\right)$$

This space is shown in Figure 4.8. CIE xy is widely used in vision and graphics textbooks and in some applications, but is usually regarded by professional colorimetrists as out of date.

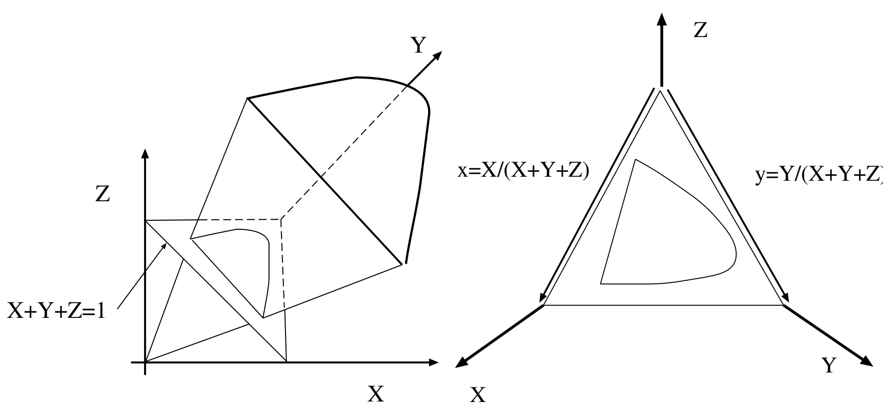

Figure 4.7. The volume of all visible colours in CIE XYZ coordinate space is a cone whose vertex is at the origin. Usually, it is easier to suppress the brightness of a colour — which we can do because to a good approximation perception of colour is linear — and we do this by intersecting the cone with the plane $X + Y + Z = 1$ to get the CIE xy space shown in figure 4.8

The RGB Colour Spaces

Colour spaces are normally invented for practical reasons, and so a wide variety exist. The **RGB colour space** is a linear colour space that formally uses single wavelength primaries (645.16 nm for R, 526.32nm for G and 444.44nm for B — see Figure 4.6). Informally, RGB uses whatever phosphors a monitor has as primaries (in Section 4.6, we explore some of the difficulties that result). Available colours are usually represented as a unit cube — usually called the **RGB cube** — whose edges represent the R, G, and B weights.

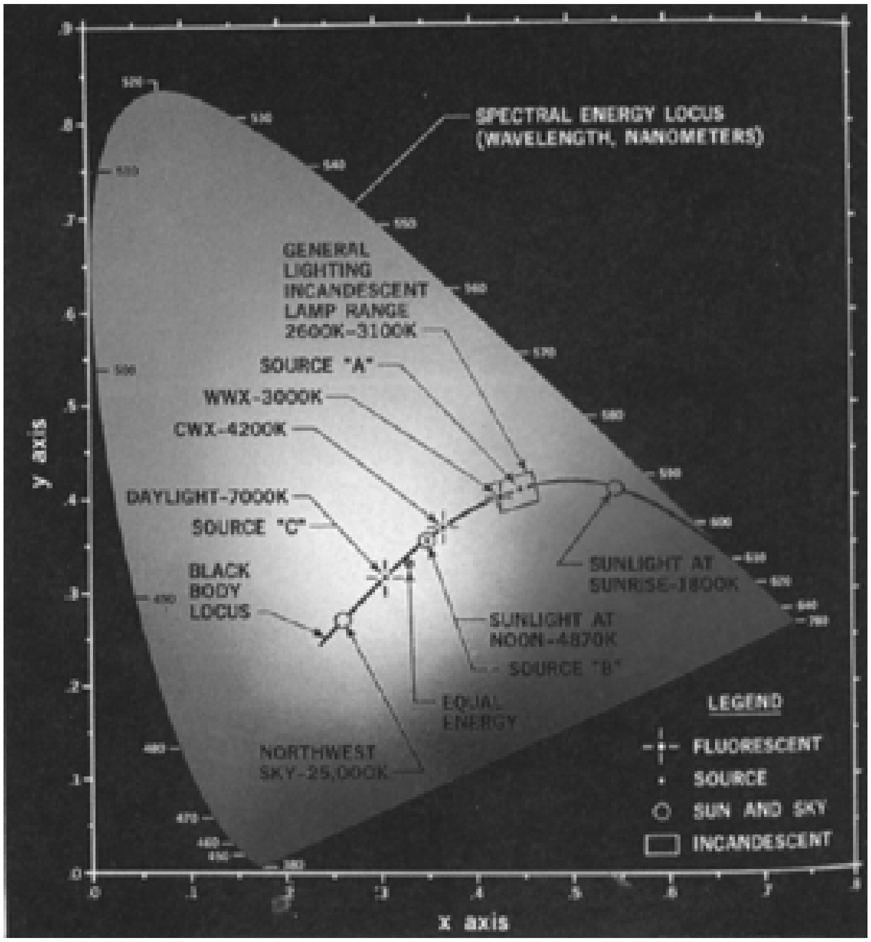

Figure 4.8. The figure shows a constant brightness section of the standard 19** standard CIE xy colour space. This space has two coordinate axes. The curved boundary of the figure is often known as the spectral locus — it represents the colours experienced when lights of a single wavelength are viewed. The colours shown represent only approximately the actual colours, because it is hard to reproduce the exact colours with printers ink. The figure shows a locus of colours due to black-body radiators at different temperatures, and a locus of different sky colours. Near the center of the diagram is the neutral point, the colour whose weights are equal for all three primaries. CIE selected the primaries so that this light appears achromatic. Generally, colours that lie further away from the neutral point are more saturated — the difference between deep red and pale pink — and hue — the difference between green and red — as one moves around the neutral point. (Taken in the fervent hope of receiving permission from Lamb and Bourriau, Colour Art and Science, p. 88)

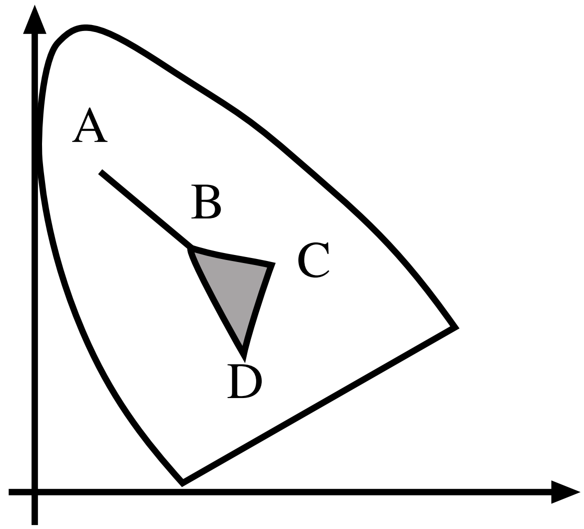

Figure 4.9. The linear model of the colour system allows a variety of useful constructions. If we have two lights whose CIE coordinates are $\boldsymbol{A}$ and $\boldsymbol{B}$ all the colours that can be obtained from non-negative mixtures of these lights are represented by the line segment joining $\boldsymbol{A}$ and $\boldsymbol{B}$. In turn, given $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$, the colours that can by obtained by mixing them lie in the triangle formed by the three points. This is important in the design of monitors — each monitor has only three phosphors, and the more saturated the colour of each phosphor the bigger the set of colours that can be displayed. This also explains why the same colours can look quite different on different monitors. The curvature of the spectral locus gives the reason that no set of three real primaries can display all colours without subtractive matching.

4.3.2 Hue, Saturation and Value

The coordinates of a colour in a linear space may not necessarily encode properties that are common in language or are important in applications. Useful colour terms include: **hue** — the property of a colour that varies in passing from red to green; **saturation** — the property of a colour that varies in passing from red to pink; and **brightness** (sometimes called **lightness** or **value**) — the property that varies in passing from black to white. For example, if we are interested in checking whether a colour lies in a particular range of reds, we might wish to encode the hue of the colour directly.

Another difficulty with linear colour spaces is that the individual coordinates do not capture human intuitions about the topology of colours; it is a common intuition that hues form a circle, in the sense that hue changes from red, through orange to yellow and then green and from there to cyan, blue, purple and then red again. Another way to think of this is to think of local hue relations: red is next to purple and orange; orange is next to red and yellow; yellow is next to orange and

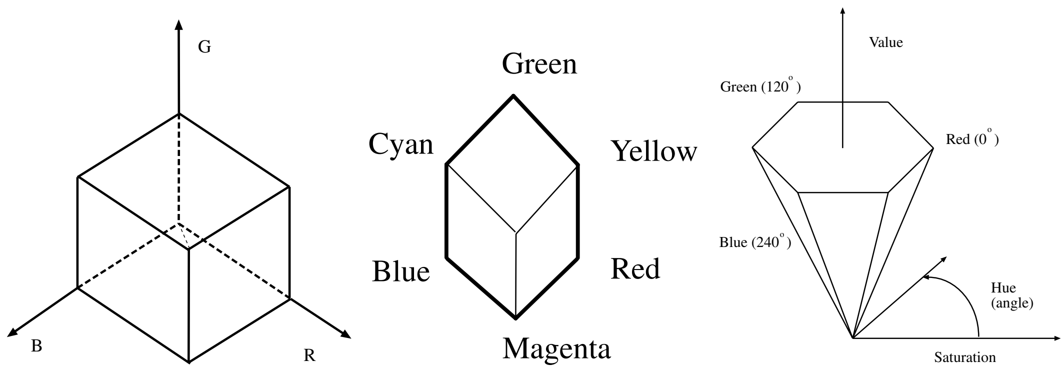

Figure 4.10. On the left, we see the RGB cube; this is the space of all colours that can be obtained by combining three primaries (R, G, and B — usually defined by the colour response of a monitor) with weights between zero and one. It is common to view this cube along its neutral axis — the axis from the origin to the point (1, 1, 1) — to see a hexagon, shown in the middle. This hexagon codes hue (the property that changes as a colour is changed from green to red) as an angle, which is intuitively satisfying. On the right, we see a cone obtained from this cross-section, where the distance along a generator of the cone gives the value (or brightness) of the colour, angle around the cone gives the hue and distance out gives the saturation of the colour.

green; green is next to yellow and cyan; cyan is next to green and blue; blue is next to cyan and purple; and purple is next to blue and red. Each of these local relations works, and globally they can be modelled by laying hues out in a circle. This means that no individual coordinate of a linear colour space can model hue, because that coordinate has a maximum value which is far away from the minimum value.

A standard method for dealing with this problem is to construct a colour space that reflects these relations by applying a non-linear transformation to the RGB space. There are many such spaces. One, called **HSV space** (for hue, saturation and value) is obtained by looking down the center axis of the RGB cube. Because RGB is a linear space, brightness — called value in HSV — varies with scale out from the origin, and we can "flatten" the RGB cube to get a 2D space of constant value, and for neatness deform it to be a hexagon. This gets the hexagon shown in Figure 4.10, where hue is given by an angle that changes as one goes round the neutral point and saturation changes as one moves away from the neutral point.

There are a variety of other possible changes of coordinate from between linear colour spaces, or from linear to non-linear colour spaces (Fairchild's book [] is a good reference). There is no obvious advantage to using one set of coordinates over another (particularly if the difference between coordinate systems is just a one-one transformation) unless one is concerned with coding and bit-rates, etc. or with perceptual uniformity.

4.3.3 Uniform Colour Spaces

Usually, one cannot reproduce colours exactly. This means it is important to know whether a colour difference would be noticeable to a human viewer; it is generally useful to be able to compare the significance of small colour differences[1].

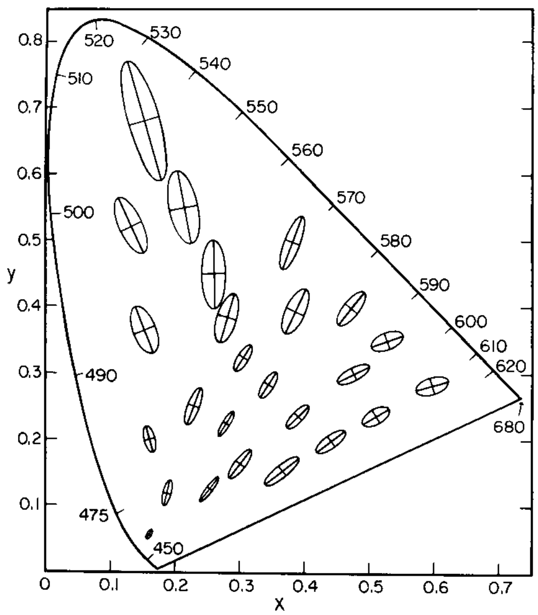

Figure 4.11. This figure shows variations in colour matches on a CIE x, y space. At the center of the ellipse is the colour of a test light; the size of the ellipse represents the scatter of lights that the human observers tested would match to the test colour; the boundary shows where the just noticeable difference is. The ellipses have been magnified 10x for clarity. The ellipses are known as MacAdam ellipses, after their inventor. Notice that the ellipses at the top are larger than those at the bottom of the figure, and that they rotate as they move up. This means that the magnitude of the difference in x, y coordinates is a poor guide to the difference in colour. (Taken in the fervent hope of receiving permission from Macadam, Color Measurement, theme and variations, p. 131)

Just noticeable differences can be obtained by modifying a colour shown to an observer until they can only just tell it has changed in a comparison with the original colour. When these differences are plotted on a colour space, they form the boundary of a region of colours that are indistinguishable from the original colours. Usually, ellipses are fitted to the just noticeable differences. It turns out that in CIE

[1] It is usually dangerous to try and compare large colour differences; consider trying to answer the question "is the blue patch more different from the yellow patch than the red patch is from the green patch?"

xy space these ellipses depend quite strongly on where in the space the difference occurs, as the Macadam ellipses in Figure 4.11 illustrate.

This means that the size of a difference in (x, y) coordinates, given by $(\Delta x)^2 + (\Delta y)^2)$, is a poor indicator of the significance of a difference in colour (if it was a good indicator, the ellipses representing indistinguishable colours would be circles). A **uniform colour space** is one in which the distance in coordinate space is a fair guide to the significance of the difference between two colours — in such a space, if the distance in coordinate space was below some threshold, then a human observer would not be able to tell the colours apart.

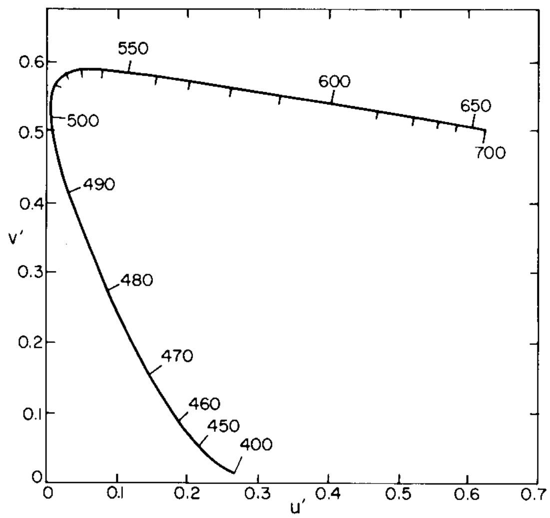

Figure 4.12. This figure shows the CIE 1976 u', v' space, which is obtained by a projective transformation of CIE x, y space. The intention is to make the MacAdam ellipses uniformly circles — this would yield a uniform colour space. A variety of non-linear transforms can be used to make the space more uniform (see [**?**] for details) (Taken in the fervent hope of receiving permission from Macadam, Color Measurement, theme and variations, p. 150)

A more uniform space can be obtained from CIE XYZ by using a projective transformation to skew the ellipses; this yields the **CIE u'v' space**, illustrated in Figure 4.12. The coordinates are:

$$(u', v') = (\frac{4X}{X + 15Y + 3Z}, \frac{9Y}{X + 15Y + 3Z})$$

Generally, the distance between coordinates in u', v' space is a fair indicator of the significance of the difference between two colours. Of course, this omits differences in brightness. **CIE LAB** is now almost universally the most popular uniform colour space. Coordinates of a colour in LAB are obtained as a non-linear mapping of the XYZ coordinates:

$$
\begin{aligned}
L^* &= 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16\\
a^* &= 500\left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right]\\
b^* &= 200\left[\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}}\right]
\end{aligned}
$$

(here X_n, Y_n and Z_n are the X, Y, and Z coordinates of a reference white patch). The reason to care about the LAB space is that it is substantially uniform. In some problems, it is important to understand how different two colours will look *to a human observer*, and differences in LAB coordinates give a good guide.

4.3.4 Chromatic Adaptation

Predicting the appearance of complex displays of colour — i.e. a stimulus that is more interesting than a pair of lights — can be quite difficult, because coloured patches in the view affect the appearance of other colour patches. The process is known as **chromatic adaptation** (illustrated in Figure 4.13). The effects of chromatic adaptation can be quite substantial.

4.4 Application: Finding Skin Using Image Colour

It is useful to find human skin; for example, gesture-based user interfaces usually need to know where the face and hands of the current user are. Similarly, if we were searching for pictures of people, a natural thing to do is to look for faces. Finally, there is a growing interest in telling whether images depict people not wearing clothing (to allow the proprietors of search engines to avoid offending sensibilities) — one way to do this is to look for human skin.

4.4.1 The Appearance of Skin

The colour of skin largely results from the effects of blood and melanin (which contribute respectively red and yellow/brown hues). Skin cannot be modelled well with a BRDF, because its appearance is affected by scattering from quite deep below the surface and by oil and sweat films on the surface. These effects make skin extremely hard to render convincingly; [Hanrahan *et al.*, 1991] has the best account.

For skin detection, we can safely ignore these difficulties. A simple model of skin reflection has a surprisingly limited range of hues and has colours that are not deeply saturated. The model must also allow bright areas due to specular reflection

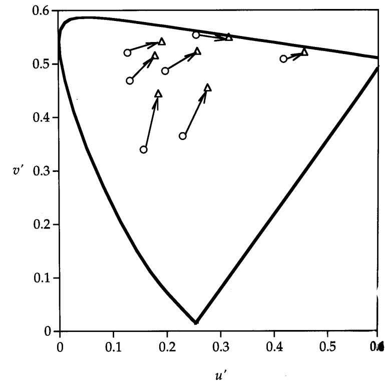

Figure 4.13. The figure shows the effects of chromatic adaptation, given on a CIE u', v' diagram. An observer adapted to illuminant D-65 will regard the chromaticity represented by a circle as the same colour as an observer adapted to illuminant A will regard the chromaticity represented by the triangle. One can think of chromatic adaptation as skewing the colour diagram; if we name colours by their appearance when we are adapted to D-65, then when we are adapted to A we disregard a significant yellow component. *(Taken in the fervent hope of receiving permission from Fairchild, Color Appearance Models, p. 192)*

on oil and sweat films. These specularities take the illumination color. Skin is typically imaged under illuminants which are close to white — otherwise the people look strange — and so the specularities have a colour that varies slightly from image to image, so that the specular skin regions appear as blueish or greenish off-white. Finally, skin has very little texture unless viewed close up or at a high resolution; we shall assume that the images of interest have a fairly coarse resolution.

4.4.2 Simple Skin Finding

In section **??**, we show an advanced skin finder which uses a learned model of skin appearance. A less sophisticated approach works acceptably (though nowhere near as well). In this approach, we look for pixels with colours in a given range and where there is little coarse-scale texture. The method sketched here is due to Forsyth and Fleck [Forsyth and Fleck, 1999].

Computing Hue and Saturation

We start by subtracting the dark response of the camera system. This response causes all colours to desaturate, and can be approximated using the smallest value in any of the three color planes omitting points closer than 10 pixels to the image boundary.

The camera R, G, and B values are then transformed into log-opponent values

Missing Figure

Figure 4.14. Marking skin pixels is useful for, for example, finding faces or hands. The figures on the top show a series of images of faces; below, we show the output of the skin marking process described in the text. While colour and texture are not an exact test for skin, much of the background is masked and faces generally appear. Notice that this approach works reasonably well for a range of skin colours.

I, R_g, and B_y (cf. e.g. [Gershon *et al.*, 1986]) as follows:

$$\begin{aligned} L(x) &= 105 \log_{10}(x + 1 + n) \\ I &= L(G) \\ R_g &= L(R) - L(G) \\ B_y &= L(B) - \frac{L(G) + L(R)}{2} \end{aligned}$$

We use the green channel to represent intensity because the red and blue channels from some cameras have poor spatial resolution. In this transformation, 105 is a convenient scaling constant and n is a random noise value, generated from a distribution uniform over the range $[0, 1)$. The random noise is added to ensure that the dark areas of the image do not look banded. This transformation makes the R_g and B_y values independent of intensity.

The colour information is smoothed by computing new R_g and B_y images. In these images, each pixel takes the median value of the pixels in a neighbourhood of the corresponding location in the old image. This means that individual, highly coloured pixels are suppressed in the smoothed images. We construct a hue image and a saturation image by forming the direction (for hue) and magnitude (for saturation) of the vector (R_g, B_y).

Computing a Texture Mask

The amount of texture surrounding a pixel can be estimated by computing a smoothed version of the intensity image, subtracting it from the original image, and

```
Compute the log-difference images described in the text
to get Rg, By, and Int

create two new images, hue and sat, the same size as the
log-difference images
for each pixel in these images
  hue(pixel) is the direction of (Rg(pixel), By(pixel))
  sat(pixel) is the magnitude of (Rg(pixel), By(pixel))
end
create two new images, smhue and smsat, the same size
as the log-difference images
for each pixel in hue, sat
  compute the median of all the pixel values in
  a neighbourhood

  insert this median into smhue, smsat at that location
end

smhue and smsat are smoothed hue and saturation images
```

Algorithm 4.1: *Computing Smoothed Hue, Saturation and Intensity Images*

then looking at the absolute values of the differences. Each pixel in the smoothed intensity image takes the median value of the pixels in a neighbourhood of the corresponding location in the original intensity image. We then form the absolute values of the difference between the smoothed image and the original image, and use these absolute values to construct a texture mask. Each pixel in the texture mask takes the median value of the pixels in a neighbourhood of the corresponding location in the absolute value image. A pixel will have a large value in the texture mask if, in a neighbourhood of the corresponding location in the intensity image, there are many quite different values.

Marking Skin

We can now mark those pixels which are probably skin, by comparing the saturation values with a range of values, the hue values with an upper and lower threshold (which vary as a function of saturation), and the texture mask with an upper threshold. Any pixel that passes all three tests is marked as a skin pixel.

Finally, we apply a trick that makes it possible to mark many of the specular components on skin. Typically, a specularity on skin is surrounded by skin, but may not pass the test we described. A natural trick is to mark all pixels as skin which pass a more relaxed version of the tests *and* which have many neighbours which are skin.

```
Create a new image, medval, the same size as the intensity image

for each pixel location in the intensity image
  compute the median of all pixel values in
  a neighbourhood

  insert this median into medval at that location
end

subtract the intensity image and medval

take the absolute value of the differences
to get absdiff

create a new image, texmask, the same size as the intensity image
for each pixel location in absdiff
  compute the median of all pixel values in
  a neighbourhood

  insert this median into texmask at that location
end

texmask is the texture mask
```

Algorithm 4.2: *Computing a Texture Mask*

This method is *ad hoc*, but surprisingly successful. The success of these methods seems to be due to cultural practices. Skin changes colour as it is viewed under different coloured lights, just like any other material, but people seem to choose lighting and to adjust pictures to reduce these effects, probably because pictures of skin where the hue is too green or too blue are very disturbing (subjects tend to look sick or dead, respectively).

4.5 Surface Colour from Image Colour

The colour of light arriving at a camera is determined by two factors: firstly, the spectral reflectance of the surface that the light is leaving, and secondly, the spectral radiance of the light falling on that surface. The colour of the light falling on surfaces can vary very widely — from blue fluorescent light indoors, to warm orange tungsten lights, to orange or even red light at sunset — so that the colour of the light arriving

```
compute smhue, smsat and texmask, as above

construct a boolean image, tempskinmask

for each pixel location
  if smsat(pixel) lies in an appropriate range
    and smhue(pixel) lies in an appropriate range
      (which depends on smsat(pixel))
    and texmask(pixel) is small

    tempskinmask(pixel) is yes
  else
    tempskinmask(pixel) is no
  end
end

construct a boolean image, skinmask

for each pixel location
  if tempskinmask(pixel) is no
    and there are many neighbours for which
      tempskinmask(pixel) is yes
    and smsat(pixel) lies in an appropriate range, broader than above
    and smhue(pixel) lies in an appropriate range, broader than above
      (which depends on smsat(pixel))
    and texmask(pixel) is small
    or tempskinmask(pixel) is yes

    skinmask(pixel) is yes
  else
    skinmask(pixel) is no
  end
end

skinmask is yes for skin pixels, otherwise no
```

Algorithm 4.3: *Simple skin finding*

at the camera can be quite a poor representation of the colour of the surfaces being viewed.

Missing
Figure

Figure 4.15. Illustrating the skin finding process.

It would be attractive to have a **colour constancy** algorithm that could take an image, discount the effect of the light, and report the actual colour of the surfaces being viewed. Colour constancy is an interesting subproblem that has the flavour of a quite general vision problem: we are determining some parameters of the world from ambiguous image measurements; we need to use a model to disentangle these measurements; and we should like to be able to report more than one solution.

4.5.1 Surface Colour Perception in People

There is some form of colour constancy algorithm in the human vision system. People are often unaware of this, and inexperienced photographers are sometimes surprised that a scene photographed indoors under fluorescent lights has a blue cast, while the same scene photographed outdoors may have a warm orange cast.

It is common to distinguish between colour constancy — which is usually thought of in terms of intensity independent descriptions of colour like hue and saturation — and **lightness constancy**, the skill that allows humans to report whether a surface is white, grey or black (the **lightness** of the surface) despite changes in the intensity of illumination (the **brightness**). Colour constancy is neither perfectly accurate [], nor unavoidable. Humans can report:

- the colour a surface would have in white light (often called **surface colour**);
- colour of the light arriving at the eye, a skill that allows artists to paint surfaces illuminated by coloured lighting [];
- and sometimes, the colour of the light falling on the surface [].

All of these reports could be by-products of a colour constancy process.

The colorimetric theories of Section 4.3 can predict the colour an observer will perceive when shown an isolated spot of light of a given power spectral distribution.

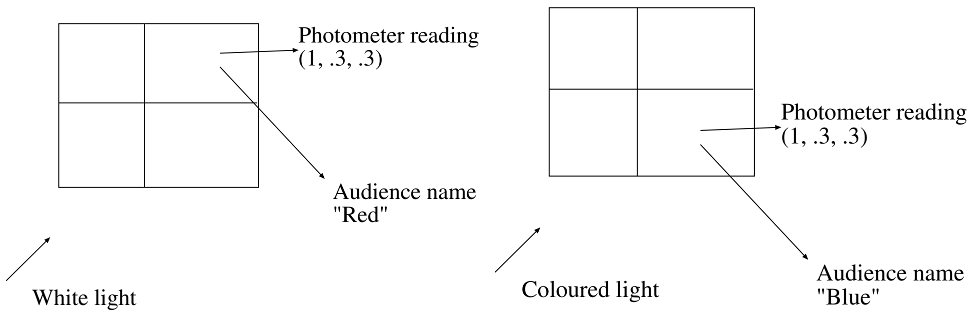

Figure 4.16. Land showed an audience a quilt of rectangles of flat coloured papers - since known as a Mondrian, for a purported resemblance to the work of that artist - illuminated using three slide projectors, casting red, green and blue light respectively. He used a photometer to measure the energy leaving a particular spot in three different channels, corresponding to the three classes of receptor in the eye. He recorded the measurement, and asked the audience to name the patch - say the answer was "red". Land then adjusted the slide projectors so that some other patch reflected light that gave the same photometer measurements, and asked the audience to name that patch. The reply would describe the patch's colour in white light - if the patch looked blue in white light, the answer would be "blue". In later versions of this demonstration, Land put wedge-shaped neutral density filters into the slide-projectors, so that the colour of the light illuminating the quilt of papers would vary slowly across the quilt. Again, although the photometer readings vary significantly from one end of a patch to another, the audience sees the patch as having a constant colour.

The human colour constancy algorithm appears to obtain cues from the structure of complex scenes, meaning that predictions from colorimetric theories can be wildly inaccurate if the spot of light is part of a larger, complex scene. Edwin Land's demonstrations [Land and McCann, 1971] (which are illustrated in Figure 4.16) give convincing examples of this effect. It is surprisingly difficult to predict what colours a human will see in a complex scene [**?**; **?**]; this is one of the many difficulties that make it hard to produce really good colour reproduction systems (section 4.6).

Human competence at colour constancy is surprisingly poorly understood. The main experiments on humans [McCann *et al.*, 1976; Arend and Reeves, 1986; **?**] do not explore all circumstances and it is not known, for example, how robust colour constancy is or the extent to which high-level cues contribute to colour judgements. Little is known about colour constancy in other animals — except that goldfish have it [Ingle, 1985]. Colour constancy clearly fails — otherwise there would be no film industry — but the circumstances under which it fails are not well understood. There is a large body of data on surface lightness perception for achromatic stimuli. Since the brightness of a surface varies with its orientation as well as with the intensity of the illuminant, one would expect that human lightness constancy would be poor: it is in fact extremely good over a wide range of illuminant variation [Jacobsen and Gilchrist, 1988].

4.5.2 Inferring Lightness

There is a lot of evidence that human lightness constancy involves two processes: one compares the brightness of various image patches, and uses this comparison to determine which patches are lighter and which darker; the second establishes some form of absolute standard to which these comparisons can be referred (e.g. [**?**]).

A Simple Model of Image Brightness

The radiance arriving at a pixel depends on the illumination of the surface being viewed, its BRDF, its configuration with respect to the source and the camera responses. The situation is considerably simplified by assuming that the scene is plane and frontal; that surfaces are Lambertian; and that the camera responds linearly to radiance.

This yields a model of the camera response C at a point $\boldsymbol{X}$ as the product of an illumination term, an albedo term and a constant that comes from the camera gain

$$C(\boldsymbol{x}) = k_c I(\boldsymbol{x}) \rho(\boldsymbol{x})$$

If we take logarithms, we get

$$\log C(\boldsymbol{x}) = \log k_c + \log I(\boldsymbol{x}) + \log \rho(\boldsymbol{x})$$

A second set of assumptions comes into play here.

- Firstly, we assume that albedoes change only quickly over space — this means that a typical set of albedoes will look like a collage of papers of different greys. This assumption is quite easily justified: firstly, there are relatively few continuous changes of albedo in the world (the best example occurs in ripening fruit); and secondly, changes of albedo often occur when one object occludes another (so we would expect the change to be fast). This means that spatial derivatives of the term $\log \rho(\boldsymbol{x})$ are either zero (where the albedo is constant) or large (at a change of albedo).
- Secondly, illumination changes only slowly over space. This assumption is somewhat realistic: for example, the illumination due to a point source will change relatively slowly unless the source is very close — so the sun is a source that is particularly good for this example; as another example, illumination inside rooms tends to change very slowly, because the white walls of the room act as area sources. This assumption fails dramatically at shadow boundaries however; we will have to see these as a special case, and assume that either there are no shadow boundaries, or that we know where they are.

Recovering Lightness from the Model

It is relatively easy to build algorithms that use our model. The earliest algorithm, Land's Retinex algorithm [**?**], has fallen into disuse. A natural approach is to differentiate the log transform, throw away small gradients, and then "integrate" the results [Horn, 1974]. There is a constant of integration missing, so lightness ratios are available, but absolute lightness measurements are not. Figure 4.17 illustrates the process for a one-dimensional example, where differentiation and integration are easy.

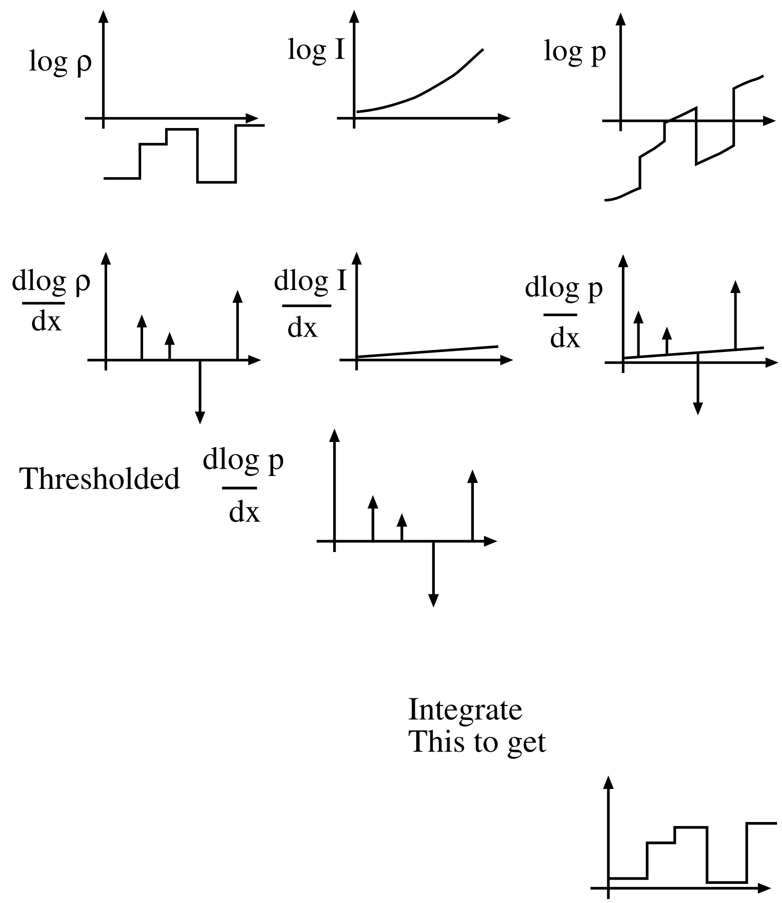

Figure 4.17. The lightness algorithm is easiest to illustrate for a 1D image. In the top row, the graph on the left shows $\log \rho(x)$; that on the center $\log I(x)$ and that on the right their sum which is $\log p$. The log of image intensity has large derivatives at changes in surface reflectance and small derivatives when the only change is due to illumination gradients. Lightness is recovered by differentiating the log intensity, thresholding to dispose of small derivatives, and then integrating, at the cost of a missing constant of integration.

This approach can be extended to two dimensions as well. Differentiating and thresholding is easy: at each point, we estimate the magnitude of the gradient, and if the magnitude is less than some threshold, we set the gradient vector to zero, else we leave it alone. The difficulty is in integrating these gradients to get the log albedo map. The thresholded gradients may not be the gradients of an image, because the mixed second partials may not be equal (integrability again; compare with section 3.5.2).

The problem can be rephrased as a minimization problem: choose the log albedo map whose gradient is most like the thresholded gradient. This is a relatively simple problem, because computing the gradient of an image is a linear operation. The x-component of the thresholded gradient is scanned into a vector $\boldsymbol{p}$ and the y-component is scanned into a vector $\boldsymbol{q}$. We write the vector representing log-albedo as $\boldsymbol{l}$. Now the process of forming the x derivative is linear, and so there is some matrix $\mathcal{M}_x$ such that $\mathcal{M}_x\boldsymbol{l}$ is the x derivative; for the y derivative, we write the corresponding matrix $\mathcal{M}_y$.

```
Form the gradient of the log of the image

At each pixel, if the gradient magnitude is below
  a threshold, replace that gradient with zero

Reconstruct the log-albedo by solving the minimization
problem described in the text

Obtain a constant of integration

Add the constant to the log-albedo, and exponentiate
```

Algorithm

4.4: *Determining the Lightness of Image Patches*

The problem becomes to find the vector $\boldsymbol{l}$ that minimizes

$$| \mathcal{M}_x\boldsymbol{l} - \boldsymbol{p} |^2 + | \mathcal{M}_y\boldsymbol{l} - \boldsymbol{q} |^2$$

This is a quadratic minimisation problem, and the answer can be found by a linear process. Some special tricks are required, because adding a constant vector to $\boldsymbol{l}$ cannot change the derivatives, so the problem does not have a unique solution. We explore the minimisation problem in the exercises.

The constant of integration needs to be obtained from some other assumption. There are two obvious possibilities:

- we can assume that the *brightest patch is white*;
- we can assume that the *average lightness is constant*.

We explore the consequences of these models in the exercises.

4.5.3 A Model for Image Colour

To build a colour constancy algorithm, we need a model to interpret the colour of pixels. By suppressing details in the physical models of Chapters **??** and above, we can model the value at a pixel as:

$$\boldsymbol{p}(\boldsymbol{x}) = g_d(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x}) + g_s(\boldsymbol{x})\boldsymbol{s}(\boldsymbol{x}) + \boldsymbol{i}(\boldsymbol{x})$$

In this model

- $\boldsymbol{d}(\boldsymbol{x})$ is the *image* colour of an equivalent *flat* frontal surface viewed under the same light;
- $g_d(\boldsymbol{x})$ is a term that varies over space and accounts for the change in brightness due to the orientation of the surface;
- $\boldsymbol{s}(\boldsymbol{x})$ is the *image* colour of the specular reflection from an equivalent *flat* frontal surface;
- $g_s(\boldsymbol{x})$ is a term that varies over space and accounts for the change in the amount of energy specularly reflected;
- and $\boldsymbol{i}(\boldsymbol{x})$ is a term that accounts for coloured interreflections, spatial changes in illumination, and the like.

We are primarily interested in information that can be extracted from colour at a local level, and so we are ignoring the detailed structure of the terms $g_d(\boldsymbol{x})$ and $\boldsymbol{i}(\boldsymbol{x})$. All evidence suggests that it is extremely hard to extract information from $\boldsymbol{i}(\boldsymbol{x})$; it can be quite small with respect to other terms and usually changes quite slowly over space. We shall not attempt to benefit from this term, and assume that it is small (or that its presence does not disrupt our algorithms too severely).

Specularities are small and bright, and can be found using these properties. In principle, we could use the methods of that section to generate new images without specularities. This brings us to the term $g_d(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x})$ in the model above. For the moment, assume that $g_d(\boldsymbol{x})$ is a constant, so we are viewing a flat, frontal surface.

The resulting term, $\boldsymbol{d}(\boldsymbol{x})$, models the world as a collage of flat, frontal diffuse coloured surfaces. We shall assume that there is a single illuminant that has a constant colour over the whole image. This term is a conglomeration of illuminant, receptor and reflectance information. It is impossible to disentangle completely in a realistic world. However, current algorithms can make quite usable estimates of surface colour from image colours, given a well populated world of coloured surfaces and a reasonable illuminant.

Finite-Dimensional Linear Models

The term $\boldsymbol{d}(\boldsymbol{x})$ is results from interactions between the spectral irradiance of the source, the spectral albedo of the surfaces, and the camera sensitivity. If a patch of perfectly diffuse surface with diffuse spectral reflectance $\rho(\lambda)$ is illuminated by a light whose spectrum is $E(\lambda)$, the spectrum of the reflected light will be $\rho(\lambda)E(\lambda)$ (multiplied by some constant to do with surface orientation, which we have already decided to ignore).

Thus, if a photoreceptor of the k'th type sees this surface patch, its response will be:

$$p_k = \int_\Lambda \sigma_k(\lambda)\rho(\lambda)E(\lambda)d\lambda$$

where Λ is the range of all relevant wavelengths and $\sigma_k(\lambda)$ is the sensitivity of the k'th photoreceptor. This response is linear in the surface reflectance and linear in the illumination, which suggests using linear models for the families of possible surface reflectances and illuminants. A **finite-dimensional linear model** models surface spectral albedoes and illuminant spectral irradiance as a weighted sum of a finite number of basis functions. We need not use the same bases for reflectances and for illuminants.

If a finite-dimensional linear model of surface reflectance is a reasonable description of the world, any surface reflectance can be written as

$$\rho(\lambda) = \sum_{j=1}^{n} r_j \phi_j(\lambda)$$

where the $\phi_j(\lambda)$ are the basis functions for the model of reflectance, and the r_j vary from surface to surface.

Similarly, if a finite-dimensional linear model of the illuminant is a reasonable model, any illuminant can be written as

$$E(\lambda) = \sum_{i=1}^{m} e_i \psi_i(\lambda)$$

where the $\psi_i(\lambda)$ are the basis functions for the model of illumination.

When both models apply, the response of a receptor of the k'th type is:

$$\begin{aligned} p_k &= \int \sigma_k(\lambda) \left(\sum_{j=1}^{n} r_j \phi_j(\lambda)\right) \left(\sum_{i=1}^{m} e_i \psi_i(\lambda)\right) d\lambda \\ &= \sum_{i=1,j=1}^{m,n} e_i r_j \left(\int \sigma_k(\lambda)\phi_j(\lambda)\psi_i(\lambda)\right) d\lambda \\ &= \sum_{i=1,j=1}^{m,n} e_i r_j g_{ijk} \end{aligned}$$

where we expect that the $g_{ijk} = \int \sigma_k(\lambda)\phi_j(\lambda)\psi_i(\lambda)d\lambda$ are known, as they are components of the world model (they can be learned from observations; see the exercises).

4.5.4 Surface Colour from Finite Dimensional Linear Models

Each of the indexed terms can be interpreted as components of a vector, and we shall use the notation $\boldsymbol{p}$ for the vector with k'th component p_k, etc. We could represent surface colour either directly by the vector of coefficients $\boldsymbol{r}$, or more indirectly by computing $\boldsymbol{r}$ and then determining what the surfaces would look like under white light. The latter representation is more useful in practice; among other things, the results are easy to interpret.

Normalizing Average Reflectance

Assume that the average reflectance in all scenes is constant and is known. In the finite-dimensional basis for reflectance we can write this average as

$$\sum_{j=1}^{n} \overline{r_j}\phi_j(\lambda)$$

Now if the average reflectance is constant, the average of the receptor responses must be constant too (the imaging process is linear), and the average of the response of the k'th receptor can be written as:

$$\overline{p_k} = \sum_{i=1,j=1}^{m,n} e_i g_{ijk} \overline{r_j}$$

If $\overline{p}$ is the vector with k'th component $\overline{p}_k$ (using the notation above) and $\mathcal{A}$ is the matrix with k, i'th component

$$\sum_{j=1}^{n} \overline{r}_j g_{ijk}$$

then we can write the above expression as:

$$\overline{\boldsymbol{p}} = \mathcal{A}\boldsymbol{e}$$

For reasonable choices of receptors, the matrix $\mathcal{A}$ will have full rank, meaning that we can determine $\boldsymbol{e}$, which gives the illumination, *if* the finite dimensional linear model for illumination has the same dimension as the number of receptors. Of course, once the illumination is known, we can report the surface reflectance at each pixel, or correct the image to look as though it were taken under white light.

The underlying assumption that average reflectance is a known constant is dangerous, however, because it is usually not even close to being true. For example, if we assume that the average reflectance is a medium gray (a popular choice - see, for example, [Buchsbaum, 1980; **?**]), an image of a leafy forest glade will be reported as a collection of objects of various grays illuminated by green light (figure 4.18 illustrates this effect). One way to try and avoid this problem is to change the

```
Compute the average colour p̄ for the image

Compute e from p̄ = Ae

To obtain a version of the image under white light, e^w:
  Now for each pixel, compute r from p_k = Σ_{i=1,j=1}^{m,n} e_i g_ijk r_j

  Replace the pixel value with p_k^w = Σ_{i=1,j=1}^{m,n} e_i^w g_ijk r_j
```

Algorithm 4.5: *Colour Constancy from Known Average Reflectance*

average for different kinds of scenes [?] - but how do we decide what average to use? Another approach is to compute an average that is not a pure spatial average; one might, for example, average the colours that were represented by ten or more pixels, but without weighting them by the number of pixels present. It is hard to say in practice how well this approach could work; there is no experimental data in the literature.

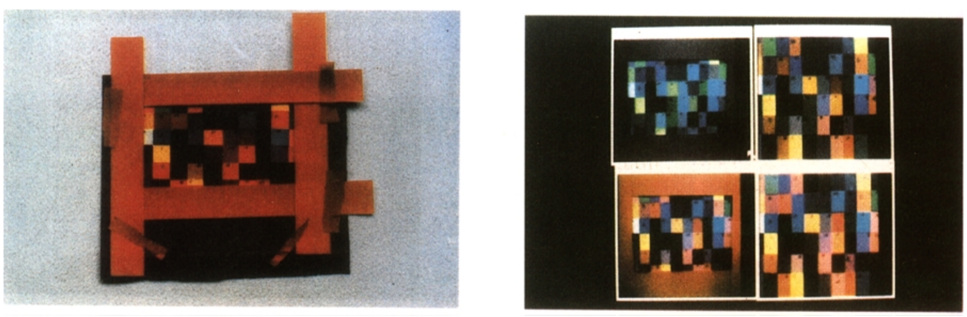

Figure 4.18. Assuming that the average reflectance is constant leads to characteristic bad behaviour. The column of figures on the right show images of a collage of coloured papers, viewed frontally under different light sources; on the left, we show estimates of the appearance under white light obtained by assuming that the average reflectance is constant. Notice that by attaching a red boundary to the collage, we have been able to fool the algorithm into assuming that the collage has a green hue.

Normalizing the Gamut

Not every possible pixel value can be obtained by taking images of real surfaces under white light. It is usually impossible to obtain values where one channel responds strongly and others do not - for example, 255 in the red channel and 0 in the green and blue channels. This means that the gamut of an image - the collection of all pixel values - contains information about the light source. For example, if one observes a pixel that has value $(255, 0, 0)$, then the light source is likely to be red in colour.

If an image gamut contains two pixel values, say $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, then it must be possible to take an image *under the same illuminant* that contains the value $t\boldsymbol{p}_1 + (1-t)\boldsymbol{p}_2$ for $0 \leq t \leq 1$ (because we could mix the colorants on the surfaces). This means that the convex hull of the image gamut contains the illuminant information. These constraints can be exploited to constrain the colour of the illuminant.

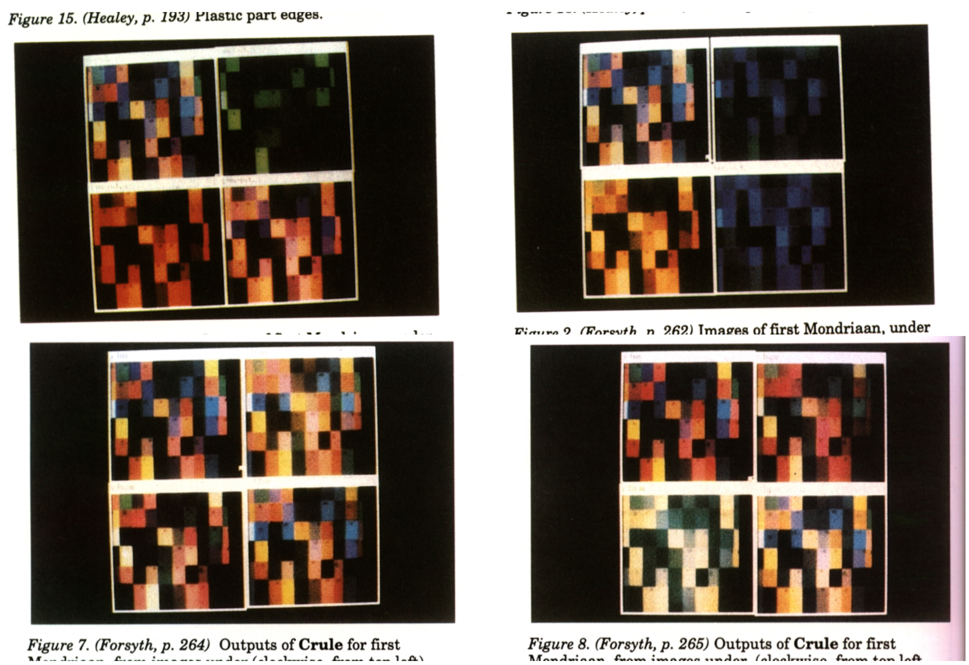

Figure 4.19. The two figures in the top row show various views of a collage of flat, frontally viewed coloured papers under different coloured light sources. The appearance of this collage changes dramatically with the change in source. On the bottom, we see predictions of the appearance of this collage under white light, using the gamut mapping algorithm described in Section 5 with illuminant chosen to get the largest possible gamut. The algorithm is quite successful, because these predictions are similar; however, deeply coloured illuminants (like the red one marked on the left) often lead to poor performance, because some colour channels make unreliable, near-zero measurements.

Write G for the convex hull of the gamut of the given image, W for the convex hull of the gamut of an image of many different surfaces under white light, and $\mathcal{M}_{\boldsymbol{e}}$ for the map that takes an image seen under illuminant $\boldsymbol{e}$ to an image seen under white light. Then the only illuminants we need to consider are those such that

$\mathcal{M}_{\boldsymbol{e}}(G) \in W$. This is most helpful if the family $\mathcal{M}_{\boldsymbol{e}}$ has a reasonable structure; in the case of finite dimensional linear models, $\mathcal{M}_{\boldsymbol{e}}$ depends linearly on $\boldsymbol{e}$, so that the family of illuminants that satisfy the constraint is also convex. This family can be constructed by intersecting a set of convex hulls, each corresponding to the family of maps that takes a hull vertex of G to some point inside W (or we could write a long series of linear constraints on $\boldsymbol{e}$).

```
Obtain the gamut W of many images of
  many different coloured surfaces under white
  light (this is the convex
  hull of all image pixel values)

Obtain the gamut G of the image (this is the convex
  hull of all image pixel values)

Obtain every element of the family of illuminant maps Me
such that MeG ∈ W
this represents all possible illuminants

Choose some element of this family, and apply
it to every pixel in the image
```

Algorithm 4.6: *Colour Constancy by Gamut Mapping*

Once we have formed this family, it remains to find an appropriate illuminant. There are a variety of possible strategies: if something is known about the likelihood of encountering particular illuminants, then one might choose the most likely; assuming that most pictures contain many different coloured surfaces leads to the choice of illuminant that makes the restored gamut the largest (which is the approach that generated the results of figure 4.19); or one might use other constraints on illuminants - for example, all the illuminants must have non-negative energy at all wavelengths - to constrain the set even further [**?**].

4.6 Digression: Device-Independent Colour Imaging

Problems of colorimetry tend to look like insignificant detail bashing until one looks closely. One such problem is growing around us with the proliferation of digital libraries. Typically, universities or museums would like to provide access to their artwork and their documents. One way to do this is via a web library with a search interface. Of course, finding a picture presents interesting problems. However, assume that an image has been found for two experts to discuss on the phone.

They will be lucky if they agree on the colours.

This is because the images are displayed on different monitors, of different ages, at different temperatures, and in different rooms. The brand, age and temperature of the monitor affects the colours that it can display.

It would be attractive for a manufacturer to be able to say that on whatever device an observer displayed an image — a monitor, an ink-jet printer, a thermal-wax printer, a dye sublimation printer or a slide imager — the image would evoke substantially the same internal experience, factored by the different resolutions of the devices. Anyone who has worked with printers knows that a colour image looks different — often very substantially so — when printed on different printers.

The technology that is required to build this ideal is known as **device independent colour imaging**. The difficulty is that typical devices can reproduce quite limited ranges of colours: only very good monitors can display anything like a 1000-1 range of brightnesses; deeply saturated pigments are hard to find; and paints and pigments come in a surprisingly limited range of albedos. A 75-1 range from dead black to bright white is a good printer; among other things, the maximum brightness of a printed page depends on the whiteness of the paper, and papers with high albedoes are difficult to produce and relatively expensive, as a visit to an office supply store will confirm.

Device independent colour imaging systems are built around two significant components:

- **Colour appearance models** predict the appearance of coloured patches to observers in different states of adaptation. A typical colour appearance model takes X,Y,Z coordinates of a coloured patch and of an adapting field — which models, for example, the room in which the viewer sees the monitor — and constructs a prediction of the viewer's internal experience.

- **Gamut mapping algorithms** which adjust the collection of colours required to produce an image to line up with the collection of colours that can actually be produced. Gamut mapping is difficult, because good solutions depend quite critically on the semantics of the image being viewed. For example, a good gamut mapping solution for images adjusts all colours reasonably evenly whereas gamut mapping business graphics usually requires that highly saturated colours remain highly saturated, but does not require accurate hue representations. Furthermore, human observers profoundly object to poor reproduction of skin colour.

With these components we could build a device independent colour imaging system for the museum problem. The museum, when it is digitizing its pictures, records an estimate of the state of adaptation of an observer in the gallery (usually to medium sunlight). It then records the X,Y,Z coordinates for each pixel in its picture.

When I decide to display this picture on my monitor, the display system must know the state of adaptation I expect in my office and the calibration of my monitor (i.e. what are the XYZ coordinates of the colour obtained by stimulating the red

Figure 4.20. Device independent colour imaging is a very old problem. Jan Vermeer was a painter particularly competent at rendering the effects of light and shade on complex surfaces. This picture, "Lady writing a letter with her maid," is a beautiful rendering of sunlight in a room. Look at the writer's left arm, which is shadowed by her body, and is dark as a result — why is the wall next to that arm bright? In fact, sunlight travelled in straight lines in 17th century Holland; the problem is that Vermeer needs to make the writers arm look dark, and the paint has a fairly low range of albedoes. As a result, the best way to make it look dark is to place it next to something bright. People are generally poor at spotting errors in the rendering of illumination, so Vermeer has modified the distribution of illumination to make her arm look dark.

phosphor on the monitor at full blast?). It must apply a colour appearance model to the museum's data, to predict what I would see in the museum. It must then use a gamut mapping algorithm to massage the result into the gamut into the range that can be displayed, and invert the colour appearance model and the monitor calibration to decide at what intensities to stimulate the phosphors.

All of these steps are uncertain; there is no universally accepted colour appearance model; there is no universally accepted gamut mapping algorithm; and monitor and printer calibrations tend to be poor (for example, most monitors have knobs on the front that the viewer can fiddle with; this makes the viewer happy, but the monitor hard to calibrate). The topic's importance is growing with the growth of digital imaging systems. A good introduction appears in Fairchild's book [?].

4.7 Notes

The use of colour in computer vision is surprisingly primitive. One difficulty is some legitimate uncertainty about what it is good for. John Mollon's remark that the primate colour system could be seen as an innovation of some kinds of fruiting tree [] is one explanation, but it is not much help.

4.7.1 Trichromacy and Colour Spaces

Up until quite recently, there was no conclusive explanation of why trichromacy applied, although it was generally believed to be due to the presence of three different types of colour receptor in the eye. Work on the genetics of photoreceptors by Nathans *et al.* can be interpreted as confirming this hunch (see []), though a full explanation is still far from clear because this work can also be interpreted as suggesting many individuals have more than three types of photoreceptor [].

There is an astonishing number of colour spaces and colour appearance models available. We discourage the practice of publishing papers that compare colour *spaces* for, say, segmentation, because the spaces are within one-one transformations of one another. The important issue is not in what coordinate system one measures colour, but how one counts the difference — so colour metrics may still bear some thought.

Colour metrics are an old topic; usually, one fits a metric tensor to MacAdam ellipses. The difficulty with this approach is that a metric tensor carries the strong implication that you can measure differences over large ranges by integration, whereas it is very hard to see large range colour comparisons as meaningful. Another concern is that the weight observers place on a difference in a Maxwellian view and the semantic significance of a difference in image colours are two very different things.

4.7.2 Lightness and Colour Constancy

There has not been much recent study of lightness constancy algorithms. The basic idea is due to Land []; his work was formalised for the computer vision community by Horn []; and a variation on Horn's algorithm was constructed by Blake [Blake, 1985]. The techniques are not as widely used as they should be, particularly given that there is some evidence they produce useful information on real images [Blake and Brelstaff, 1987]. Classifying illumination vs albedo simply by looking at the magnitude of the gradient is crude, and ignores at least one important cue (very large changes must be illumination, however fast they occur); there is significant room for improvement.

The most significant case in colour constancy occurs when there are three classes of photoreceptor; others have been studied [?; ?; ?; Maloney and Wandell, 1986; ?], but this is mainly an excuse to do linear algebra.

Finite-dimensional linear models for spectral reflectances can be supported by an appeal to surface physics, as spectral absorbtion lines are thickened by solid state effects. The main experimental justifications for finite-dimensional linear models of surface reflectance are Cohen's [Cohen, 1964] measurements of the surface reflectance of a selection of standard reference surfaces known as **Munsell chips**, and Krinov's [Krinov, 1947] measurements of a selection of natural objects. Cohen [Cohen, 1964] performed a principal axis decomposition of his data, to obtain a set of basis functions, and Maloney [Maloney, 1984] fitted weighted sums of these functions to Krinov's date to get good fits with patterned deviations. The first three principal axes explained in each case a very high percentage of the sample variance (near 99 %), and hence a linear combination of these functions fitted all the sampled functions rather well. More recently, Maloney [Maloney, 1986] fitted Cohen's basis vectors to a large set of data, including Krinov's data, and further data on the surface reflectances of Munsell chips, and concluded that the dimension of an accurate model of surface reflectance was of the order of five or six.

On surfaces like plastics, the specular component of the reflected light is the same colour as the illuminant. If we can identify specular regions from such objects in the image, the colour of the illuminant is known. This idea has been popular for a long time[2]. Recent versions of this idea appear in, for example, [Flock, 1984; D'Zmura and Lennie, 1986; Klinker *et al.*, 1987; Lee, 1986].

There is surprisingly little work on colour constancy that unifies a study of the spatial variation in illumination with solutions for surface colour, which is why we were reduced to ignoring a number of terms in our colour model. There is substantial room for research here, too.

4.7.3 Colour in Recognition

As future chapters will show, it is quite tricky to build systems that use object colour to help in recognition. A variety of effects cause image colours to be poor measurements of surface colour. Uniform colour spaces offer some help here, if we are willing to swallow a fairly loose evolutionary argument: it is worth understanding the colour differences that humans recognise, because they are adapted to measurements that are useful.

4.8 Assignments

Exercises

1. Sit down with a friend and a packet of coloured papers, and compare the colour names that you use. You will need a large packet of papers — one can very often get collections of coloured swatches for paint, or for the Pantone colour system very cheaply. The best names to try are basic colour names

[2] Judd [Judd, 1940] writing in 1960 about early German work in surface colour perception refers to it as "a more usual view".

— the terms red, pink, orange, yellow, green, blue, purple, brown, white, gray and black, which (with a small number of other terms) have remarkable canonical properties that apply widely across different languages [**?**; **?**; **?**]. You will find it surprisingly easy to disagree on which colours should be called blue and which green, for example.

2. Derive the equations for transforming from RGB to CIE XYZ, and back. This is a linear transformation. It is sufficient to write out the expressions for the elements of the linear transformation — you don't have to look up the actual numerical values of the colour matching functions.

3. Linear colour spaces are obtained by choosing primaries and then constructing colourmatching functions for those primaries. Show that there is a linear transformation that takes the coordinates of a colour in one linear colour space to those in another; the easiest way to do this is to write out the transformation in terms of the colourmatching functions.

4. Exercise 3 means that, in setting up a linear colour space, it is possible to choose primaries arbitrarily, but there are constraints on the choice of colour matching functions. Why? What are these constraints?

5. Two surfaces that have the same colour under one light and different colours under another are often referred to as *metamers*. An *optimal colour* is a spectral reflectance or radiance that has value 0 at some wavelengths and 1 at others. Though optimal colours don't occur in practice, they are a useful device (due to Ostwald) for explaining various effects.

   - use optimal colours to explain how metamerism occurs.
   - given a particular spectral albedo, show that there are an infinite number of metameric spectral albedoes.
   - use optimal colours to construct an example of surfaces that look very different under one light (say, red and green) and the same under another.
   - use optimal colours to construct an example of surfaces that swop apparent colour when the light is changed (i.e. surface one looks red and surface two looks green under light one, and surface one looks green and surface two looks red under light two).

6. You have to map the gamut for a printer to that of a monitor. There are colours in each gamut that do not appear in the other. Given a monitor colour that can't be reproduced exactly, you could choose the printer colour that is closest. Why is this a bad idea for reproducing images? Would it work for reproducing "business graphics" (bar charts, pie charts, and the like which all consist of many differernt large blocks of a single colour)?

7. *Volume colour* is a phenomenon associated with translucent materials that are coloured — the most attractive example is a glass of wine. The colouring

comes from different absorption coefficients at different wavelengths. Explain (1) why a glass of sufficiently deeply coloured red wine (a good Cahors, or Gigondas) looks black (2) why a big glass of lightly coloured red wine also looks black. Experimental work is optional.

8. (This exercise requires some knowledge of numerical analysis). In section 4.5.2, we set up the problem of recovering the log-albedo for a set of surfaces as one of minimizing
$$| \mathcal{M}_x \boldsymbol{l} - \boldsymbol{p} |^2 + | \mathcal{M}_y \boldsymbol{l} - \boldsymbol{q} |^2$$
where $\mathcal{M}_x$ forms the x derivative of $\boldsymbol{l}$ and $\mathcal{M}_y$ forms the y derivative (i.e. $\mathcal{M}_x \boldsymbol{l}$ is the x-derivative).
   - We asserted that $\mathcal{M}_x$ and $\mathcal{M}_y$ existed. Use the expression for forward differences (or central differences, or any other difference approximation to the derivative) to form these matrices. Almost every element is zero.
   - The minimisation problem can be written in the form
   $$\text{choose } \boldsymbol{l} \text{ to minimize } (\mathcal{A}\boldsymbol{l} + \boldsymbol{b})^T (\mathcal{A}\boldsymbol{l} + b)$$
   Determine the values of $\mathcal{A}$ and $\boldsymbol{b}$, and show how to solve this general problem. You will need to keep in mind that $\mathcal{A}$ does not have full rank, so you can't go inverting it.

9. In section 4.5.2, we mentioned two assumptions that would yield a constant of integration.
   - Show how to use these assumptions to recover an albedo map.
   - For each assumption, describe a situation where it fails, and describe the nature of the failure. Your examples should work for cases where there are many different albedoes in view.

10. Read the book " Colour: Art and Science", by Lamb and Bourriau, Cambridge University Press, 1995.

Programming Assignments

1. Spectra for illuminants and for surfaces are available on the web (for example `http:whereisit?`). Fit a finite-dimensional linear model to a set of illuminants and surface reflectances using principal components analysis, render the resulting models, and compare your rendering with an exact rendering. Where do you get the most significant errors? why?

2. Fitting a finite-dimensional linear model to illuminants and reflectances separately is somewhat ill-advised, because there is no guarantee that the *interactions* will be represented well (they're not accounted for in the fitting error). It turns out that one can obtain g_{ijk} by a fitting process that sidesteps

the use of basis functions. Implement this procedure (which is described in detail in [Marimont and Wandell, 1992]), and compare the results with those obtained from the previous assignment.

3. Build a colour constancy algorithm that uses the assumption that the spatial average of reflectance is constant. Use finite-dimensional linear models. You can get values of g_{ijk} from your solution to exercise 3.

4. We ignore colour interreflections in our surface colour model. Do an experiment to get some idea of the size of colour shifts possible from colour interreflections (which are astonishingly big). Humans very seldom interpret colour interreflections as surface colour — speculate as to why this might be the case, using the discussion of the lightness algorithm as a guide.

5. Build a specularity finder along the lines described in section **??**.