well, because manufacturers try to have sensitivities that are reasonably compatible with human receptor sensitivities. They do this so that cameras give about the same responses to colored lights that people do; as a result, cameras tend to have quite similar receptor sensitivities. There are three ways to proceed: install narrow-band filters in front of the lens (difficult to do and seldom justified); apply a transformation to the receptor outputs that makes them behave more like narrow-band receptors (often helpful, if the necessary data are available, Finlayson *et al.* (1994*b*);Barnard *et al.* (2001*a*)); or assume that they are narrow-band receptors and tolerate any errors that result (generally quite successful).

3.4.2 The Specular Term

The specular component will have a characteristic color, and its intensity will change with position. We can model the specular component as

$$g_s(\boldsymbol{x})\boldsymbol{s}(\boldsymbol{x}),$$

where $\boldsymbol{s}(\boldsymbol{x})$ is the unit intensity *image* color of the specular reflection at that pixel, and $g_s(\boldsymbol{x})$ is a term that varies from pixel to pixel, and models the amount of energy specularly reflected. We expect $g_s(\boldsymbol{x})$ to be zero at most points, and large at some points.

The color $\boldsymbol{s}(\boldsymbol{x})$ of the specular component depends on the material. Generally, metal surfaces have a specular component that is wavelength dependent and so takes on a characteristic color that depends on the metal (gold is yellow, copper is orange, platinum is white, and osmium is blue or purple). Surfaces that do not conduct—*dielectric surfaces*— have a specular component that is independent of wavelength (e.g., the specularities on a shiny plastic object are the color of the light). Section 3.5.1 describes how these properties can be used to find specularities, and to find image regions corresponding to metal or plastic objects.

3.5 INFERENCE FROM COLOR

Our color model supports a variety of inferences. It can be used to find specularities (Section 3.5.1); to remove shadows (Section 3.5.2); and to infer surface color (Section 3.5.3).

3.5.1 Finding Specularities Using Color

Specular reflections can have strong effects on an object's appearance. Typically, they appear as small, bright patches, called *highlights* or *specularities*. Highlights have a substantial effect on human perception of a surface properties; the addition of small, highlight-like patches to a figure makes the object depicted look glossy or shiny. Specularities are often bright enough to saturate the camera, so that the color of a specularity can be hard to measure. However, because the appearance of a specularity is quite strongly constrained, there are a number of effective schemes for marking them, and the results can be used as a shape cue.

The dynamic range of practically available albedoes is relatively small. Surfaces with very high or very low albedo are difficult to make. Uniform illumination is common too, and most cameras are reasonably close to linear within their operating range. This means that very bright patches cannot be due to diffuse reflection;

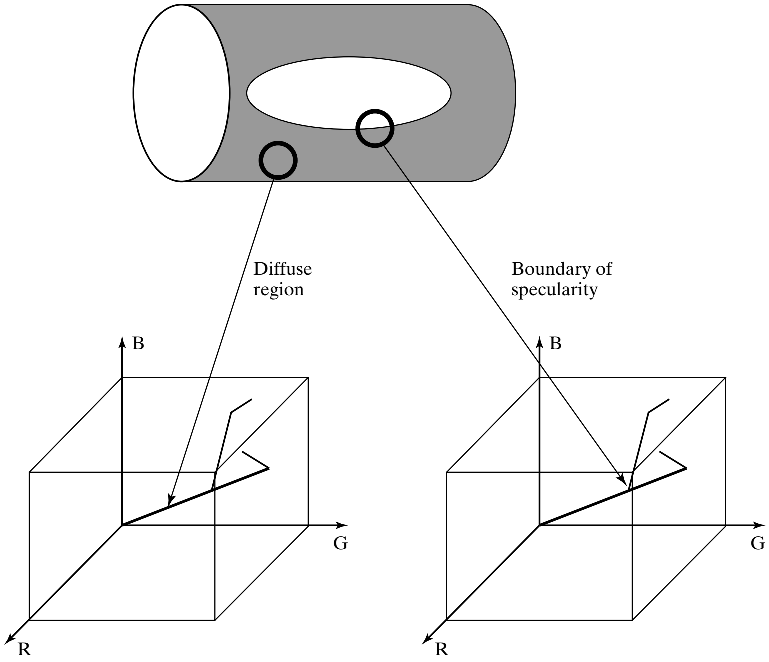

FIGURE 3.18: The linear clusters produced by specularities on plastic objects can be found by reasoning about windows of image pixels. In a world of plastic objects on a black background, a background window produces a region of pixels that are point-like in color space—all pixels have the same color. A window that lies along the body produces a line-like cluster of points in color space, because the intensity varies, but the color does not. At the boundary of a specularity, windows produce plane-like clusters because points are a weighted combination of two different colors (the specular and the body color). Finally, at the interior of a specular region, the windows can produce volume-like clusters, because the camera saturates, and the extent of the window can include both the boundary-style window and saturated points. Whether a region is line-like, plane-like, or volume-like can be determined easily by looking at the eigenvalues of the covariance of the pixels.

they must be either sources (of one form or another—perhaps a stained glass window with the light behind it) or specularities. Furthermore, specularities tend to be small. Thus, looking for small, bright patches can be an effective way to find specularities (Brelstaff and Blake 1988*a*).

An alternative is to use image color. From our model, the color of specularities on dielectric objects is the color of the light source. Assume we can ignore the interreflection term, either because we have an isolated object or because the term doesn't change much over the object we are viewing. Our model gives the image color as a sum of a diffuse term and a specular term. Now consider a patch of surface around a specularity. We expect that this patch is small, because we expect specularities to be small (this will be true on curved surfaces; the approach we are describing might not work for flat surfaces). Because the patch is small, we expect that $\boldsymbol{d}(\boldsymbol{x})$ does not change in the patch; we do not expect to be unlucky, and have a specularity on an albedo boundary. We expect that $\boldsymbol{s}(\boldsymbol{x})$ does not change within the patch, because the color of the specularity will be the color of the light source, and this will not change within a small patch.

On a dielectric object, as we move from a patch with no specular reflection to one with a specular component, the image color will change, because the size of the specular component changes. We can write the image color as

$$g_d(\boldsymbol{x})\boldsymbol{d} + g_s(\boldsymbol{x})\boldsymbol{s},$$

where $\boldsymbol{s}$ is the color of the source and $\boldsymbol{d}$ is the color of the diffuse reflected light, $g_d(\boldsymbol{x})$ is the geometric term that depends on the orientation of the surface, and $g_s(\boldsymbol{x})$ is a term that gives the extent of the specular reflection.

If the object is curved, then $g_s(\boldsymbol{x})$ is small over much of the surface and large only around specularities; $g_d(\boldsymbol{x})$ varies more slowly with the orientation of the surface. We now map the colors produced by this surface in receptor response space and look at the structures that appear there.

The term $g_d(\boldsymbol{x})\boldsymbol{d}$ produces a line that should extend to pass through the origin because it represents the same vector of receptor responses multiplied by a constant that varies over space. If there is a specularity, then we expect to see a second line due to $g_s(\boldsymbol{x})\boldsymbol{s}$. This does not, in general, pass through the origin (because of the diffuse term). This is a line, rather than a planar region, because $g_s(\boldsymbol{x})$ is large over only a small range of surface normals. We expect that, because the surface is curved, this corresponds to a small region of surface. The term $g_d(\boldsymbol{x})$ should be approximately constant in this region. We expect a line, rather than an isolated pixel value, because we expect surfaces to have (possibly narrow) specular lobes, meaning that the specular coefficient has a range of values. This second line might collide with a face of the color cube and get clipped.

The resulting dog-leg pattern leads pretty much immediately to a specularity marking algorithm: find the pattern and then find the specular line. All the pixels on this line are specular pixels, and the specular and diffuse components can be estimated easily. For the approach to work effectively, we need to be confident that only one object is represented in the collection of pixels. This is helped by using local image windows, as illustrated by Figure 3.18. The observations underlying the method hold even if the surface is not monochrome—a coffee mug with a picture on it, for example—but finding the resulting structures in the color space now becomes something of a nuisance and, to our knowledge, has not been demonstrated.

3.5.2 Shadow Removal Using Color

Lightness methods make the assumption that "fast" edges in images are due to changes in albedo (Section 2.2.3). This assumption is usable, but fails badly at shadows, particularly shadows in sunlight outdoors (Figure 3.20), where there can be a large and fast change of image brightness. People usually are not fooled into believing that a shadow is a patch of dark surface, so must have some method to identify shadow edges. Home users often like editing and improving photographs, and programs that could remove shadows from images would be valuable. A shadow removal program would work something like a lightness method: find all edges, identify the shadow edges, remove those, and then integrate to get the picture back.

There are some cues for finding shadow edges that seem natural, but don't work well. One might assume that shadow edges have very large dynamic range (which albedo edges can't have; see Section 2.1.1), but this is not always the case. One might assume that, at a shadow edge, there was a change in brightness but not in color. It turns out that this is not the case for outdoor shadows, because the lit region is illuminated by yellowish sunlight, and the shadowed region is illuminated by bluish light from the sky, or sometimes by interreflected light from buildings, and so on. However, a really useful cue can be obtained by modelling the different light sources.

We assume that light sources are black bodies, so that their spectral energy density is a function of temperature. We assume that surfaces are diffuse. We use the simplified black-body model of Section 3.2.1, where, writing T for the temperature of the body in Kelvins, h for Planck's constant, k for Boltzmann's constant, c for the speed of light, and λ for the wavelength, we have

$$E(\lambda; T) = C\frac{\exp(-hc/k\lambda T)}{\lambda^5}$$

(C is some constant of proportionality). Now assume that the color receptors each respond only at one wavelength, which we write λ_k for the k'th receptor, so that $\sigma_k(\lambda) = \delta(\lambda - \lambda_k)$. If we view a surface with spectral albedo $\rho(\lambda)$ illuminated by one of these sources at temperature T, the response of the j'th receptor will be

$$r_j = \int \sigma_j(\lambda)\rho(\lambda)K\frac{\exp(-hc/k\lambda T)}{\lambda^5}d\lambda = K\rho(\lambda_j)\frac{\exp(-hc/k\lambda_j T)}{\lambda_j^5}.$$

We can form a color space that is very well behaved by taking $c_1 = \log(r_1/r_3)$, $c_2 = \log(r_2/r_3)$, because

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \frac{1}{T}\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

where $a_1 = \log\rho(\lambda_1) - \log\rho(\lambda_3) + 5\log\lambda_3 - 5\log\lambda_1$ and $b_1 = (hc/k)(1/\lambda_3 - 1/\lambda_1)$ (and a_2, b_2 follow). Notice that, when one changes the color temperature of the source, the (c_1, c_2) coordinates move along a straight line. The direction of the line depends on the sensor, *but not on the surface.* Call this direction the **color temperature direction**. The intercept of the line depends on the surface.

Now consider a world of colored surfaces, and map the image colors to this space. There is a family of parallel lines in this space, whose direction is the color temperature direction. Different surfaces may map to different lines. If we change the color temperature of the illuminant, then each color in this space will move along the color temperature direction, but colors will not move from line to line. We now represent a surface color by its line. For example, we could construct a line through the origin that is perpendicular to color temperature direction, then represent a surface color by distance along this line (Figure 3.19). We can represent each pixel in the image in this space, and in this representation the color image becomes a gray-level image, *where the gray level does not change inside shadows* (because a shadow region just has a different color temperature to the non-shadowed

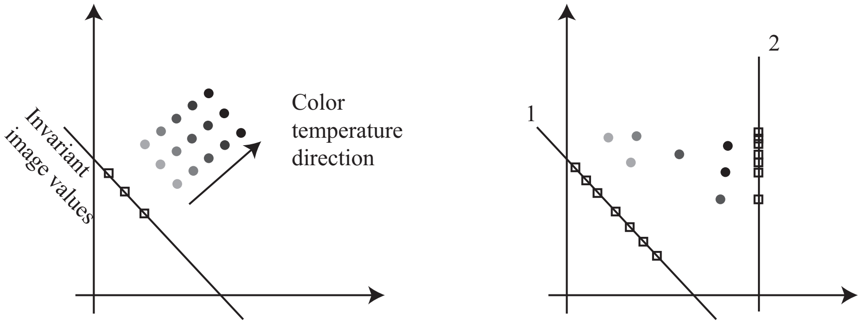

FIGURE 3.19: Changing the color temperature of the light under which a surface is viewed moves the (c_1, c_2) coordinates of that surface along the color temperature direction (**left**; the different gray patches represent the same surface under different lights). If we now project the coordinates along the (c_1, c_2) direction onto some line, we obtain a value that doesn't change when the illuminant color temperature changes. This is the invariant value for that pixel. Generally, we do not know enough about the imaging system to estimate the color temperature direction. However, we expect to see many different surfaces in each scene; this suggests that the right choice of color temperature direction on the **right** is 1 (where there are many different types of surface) rather than 2 (where the range of invariant values is small).

region). Finlayson (1996) calls this the *invariant image*. Any edge that appears in the image but not in the invariant image is a shadow edge, so we can now apply our original formula: find all edges, identify the shadow edges, remove those, and then integrate to get the picture back.

Of course, under practical circumstances, usually we do not know enough about the sensors to evaluate the as and bs that define this family of lines, so we cannot get the invariant image directly. However, we can infer a direction in (c_1, c_2) space that is a good estimate by a form of entropy reasoning. We must choose a color temperature direction. Assume the world is rich in differently colored surfaces. Now consider two surfaces S_1 and S_2. If $\boldsymbol{c}_1$ (the (c_1, c_2) values for S_1) and $\boldsymbol{c}_2$ are such that $\boldsymbol{c}_1 - \boldsymbol{c}_2$ is parallel to the color temperature direction, we can choose T_1 and T_2 so that S_1 viewed under light with color temperature T_1 will look the same as S_2 viewed under light with color temperature T_2. We expect this to be uncommon, because surfaces tend not to mimic one another in this way. This means we expect that colors will tend to spread out when we project along a good estimate of the color temperature direction. A reasonable measure of this spreading out is the *entropy* of the histogram of projected colors. We can now estimate the invariant image, without knowing anything about the sensor. We search directions in (c_1, c_2) space, projecting all the image colors along that direction; our estimate of the color temperature direction is the one where this projection yields the largest entropy. From this we can compute the invariant image, and so apply our shadow removal strategy above. In practice, the method works well, though great care is required with the integration procedure to get the best results (Figure 3.20).

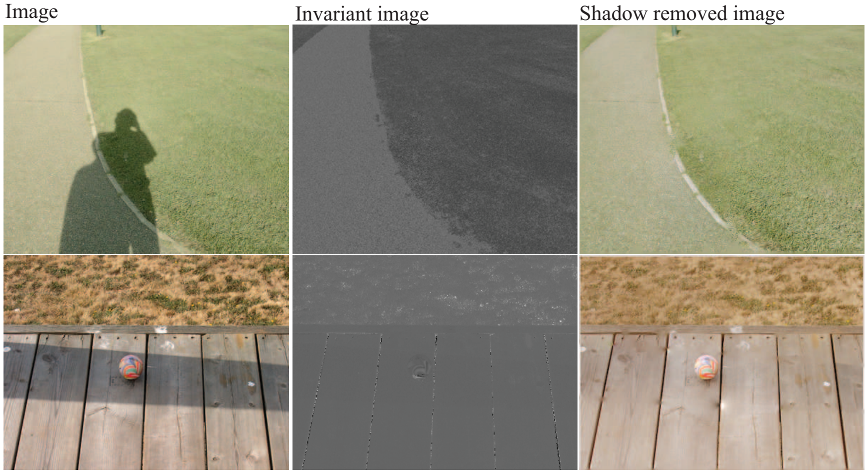

FIGURE 3.20: The invariant of the text and of Figure 3.19 does not change value when a surface is shadowed. Finlayson *et al.* use this to build a shadow removal system that works by (a) taking image edges; (b) forming an invariant image; then (c) using that invariant image to identify shadow edges; and finally (d) integrating only non-shadow edges to form the result. The results are quite convincing. *This figure was originally published as Figures 2 and 4 of "On the Removal of Shadows From Images," G. Finlayson, S. Hordley, C. Lu and M. Drew, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006 © IEEE, 2006.*

3.5.3 Color Constancy: Surface Color from Image Color

In our model, the image color depends on both light color and on surface color. If we light a green surface with white light, we get a green image; if we light a white surface with a green light, we also get a green image. This makes it difficult to name surface colors from pictures. We would like to have an algorithm that can take an image, discount the effect of the light, and report the actual color of the surface being viewed.

This process is called *color constancy.* Humans have some form of color constancy algorithm. People are often unaware of this, and inexperienced photographers are sometimes surprised that a scene photographed indoors under fluorescent lights has a blue cast, whereas the same scene photographed outdoors may have a warm orange cast. The simple linear models of Section 3.3 can predict the color an observer will perceive when shown an isolated spot of light of a given power spectral distribution. But if this spot is part of a larger, more complex scene, these models can give wildly inaccurate predictions. This is because the human color constancy algorithm uses various forms of scene information to decide what color to report. Demonstrations by Land and McCann (1971), which are illustrated in Figure 3.21, give convincing examples of this effect. It is surprisingly difficult to predict what colors a human will see in a complex scene (Fairchild (1998); Helson (1938*a*); (1938*b*); (1934); (1940)). This is one of the many difficulties that make it hard to produce really good color reproduction systems.

Human color constancy is not perfectly accurate, and people can choose to

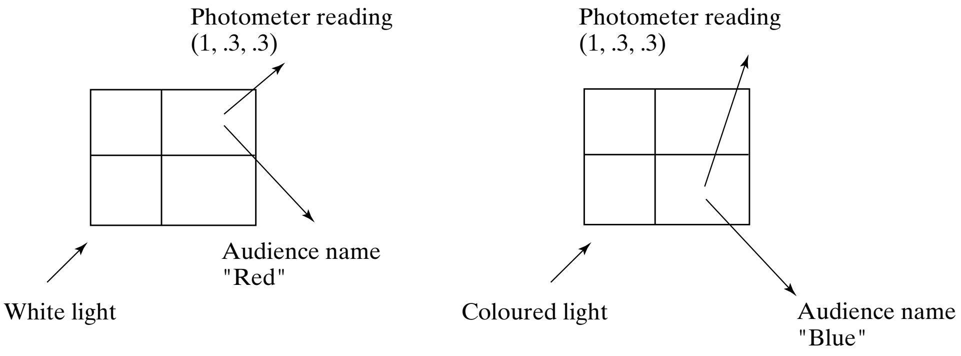

FIGURE 3.21: Land showed an audience a quilt of rectangles of flat colored papers—since known as a Mondrian for a purported resemblance to the work of that artist—illuminated using three slide projectors, casting red, green and blue light respectively. He used a photometer to measure the energy leaving 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. Assume the answer was "red" (on the **left**). 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 color in white light—if the patch looked blue in white light, the answer would be "blue" (on the **right**). In later versions of this demonstration, Land put wedge-shaped neutral density filters into the slide projectors so that the color 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 color.

disregard information from their color constancy system. As a result, people can often report:

- the color a surface would have in white light (often called *surface color*);
- the color of the light arriving at the eye (a useful skill that allows artists to paint surfaces illuminated by colored lighting); and
- the color of the light falling on the surface.

The model of image color in Section 3.4 is

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

We decided to ignore the interreflection term $\boldsymbol{i}(\boldsymbol{x})$. In principle, we could use the methods of Section 3.5.1 to generate new images without specularities. This brings us to the term $g_d(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x})$. 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 colored surfaces. Such worlds are sometimes called *Mondrian worlds*, after the painter. Notice that, under our assumptions, $\boldsymbol{d}(\boldsymbol{x})$ consists of a set of patches of fixed color. We assume that there is a single illuminant that has a constant color over the whole image. This term is a conglomeration of illuminant, receptor, and reflectance information. It is impossible to disentangle these completely in a realistic world. However, current algorithms can make quite usable estimates of surface color from image colors given a well-populated world of colored surfaces and a reasonable illuminant.

Recall from Section 3.4 that 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 is $\rho(\lambda)E(\lambda)$ (multiplied by some constant to do with surface orientation, which we have already decided to ignore). If a linear photoreceptor of the kth type sees this surface patch, its response is:

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

where Λ is the range of all relevant wavelengths, and $\sigma_k(\lambda)$ is the sensitivity of the kth photoreceptor.

Finite-Dimensional Linear Models

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 energy density 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 kth 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).

Inferring Surface Color

The finite-dimensional linear model describes the interaction between illumination color, surface color, and image color. To infer surface color from image color, we need some sort of assumption. There are several plausible cues that can be used.

Specular reflections at dielectric surfaces have uniform specular albedo. We could find the specularities with the methods of that section, then recover surface color using this information. At a specularity, we have

$$p_k = \int \sigma_k(\lambda) \sum_{i=1}^{m} e_i \psi_i(\lambda) d\lambda,$$

and so if we knew the spectral sensitivities of the sensor and the basis functions ψ_i, we could solve for e_i by solving a linear system. Now we know all e_i, and all p_k for each pixel. We can solve the linear system

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

in the unknown r_j to recover reflectance coefficients.

Known average reflectance is another plausible cue. In this case, we assume that the spatial average of reflectance in all scenes is constant and known (e.g., we might assume that all scenes have a spatial average of reflectance that is dull gray). 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 (if the imaging process is linear; see the discussion), and the average of the response of the kth receptor can be written as:

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

We know $\overline{p_k}$ and $\overline{r_j}$, and so have a linear system in the unknown light coefficients e_i. We solve this, and then recover reflectance coefficients at each pixel, as for the case of specularities. For reasonable choices of reflectors and dimension of light and surface basis, this linear system will have full rank.

The **gamut** of a color image is revealing. The gamut is the set of different colors that appears in the image. Generally, it is difficult to obtain strongly colored pixels under white light with current imaging systems. Furthermore, if the picture is taken under strongly colored light, that will tend to bias the gamut. One doesn't see bright green pixels in images taken under deep red light, for example. As a result, the image gamut is a source of information about the illumination. If an image gamut contains two pixel values—call them $\boldsymbol{p}_1$ 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 illuminant information depends on the convex hull of the image gamut. There are now various methods to exploit these observations. There is usually more than one illuminant consistent with a given image gamut, and geometric methods can be used to identify the consistent illuminants. This set can be narrowed down using probabilistic methods (for example, images contain lots of different colors (Forsyth 1990)) or physical methods (for example, the main sources of illumination are the sun and the sky, well modelled as black bodies (Finlayson and Hordley 2000)).

3.6 NOTES

There are a number of important general resources on the use of color. We recommend Hardin and Maffi (1997), Lamb and Bourriau (1995), Lynch and Livingston (2001), Minnaert (1993), Trussell *et al.* (1997), Williamson and Cummins (1983). Wyszecki and Stiles (1982) contains an enormous amount of helpful information. Recent textbooks with an emphasis on color include Velho *et al.* (2008), Lee (2009), Reinhard *et al.* (2008), Gevers *et al.* (2011) and Burger and Burge (2009).

Trichromacy and Color Spaces

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 color receptor in the eye. Work on the genetics of photoreceptors can be interpreted as confirming this hunch (see Nathans *et al.* (1986*a*) and Nathans *et al.* (1986*b*)), although 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 (Mollon 1995).

There is an astonishing number of color spaces and color appearance models available. The important issue is not in what coordinate system one measures color, but how one counts the difference, so color metrics may still bear some thought.

Color 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 color 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 colors are two very different things.

Specularity Finding

The specularity finding method we describe is due to Shafer (1985), with improvements due to Klinker *et al.* (1987), (1990), and to Maxwell and Shafer (2000).

Specularities can also be detected because they are small and bright (Brelstaff and Blake 1988*a*), because they differ in color and motion from the background (Lee and Bajcsy 1992*a*, Lee and Bajcsy 1992*b*, Zheng and Murata 2000), or because they distort patterns (Del Pozo and Savarese 2007). Specularities are a prodigious nuisance in reconstruction, because specularities cause matching points in different images to have different colors; various motion-based strategies have been developed to remove them in these applications (Lin *et al.* 2002, Swaminathan *et al.* 2002, Criminisi *et al.* 2005).

Color Constancy

Land reported a variety of color vision experiments (Land (1959*a*), (1959*b*), (1959*c*), (1983)). Finite-dimensional linear models for spectral reflectances can be supported by an appeal to surface physics as spectral absorption lines are thickened by solid state effects. The main experimental justifications for finite-dimensional linear models of surface reflectance are measurements, by Cohen (1964), of the surface reflectance of a selection of standard reference surfaces known as *Munsell chips*, and measurements of a selection of natural objects by Krinov (1947). Cohen (1964) performed a principal axis decomposition of his data to obtain a set of basis functions, and 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 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 (1986) fitted Cohen's (1964) basis vectors to a large set of data, including Krinov's (1947) data, and further data on the surface reflectances of Munsell chips, and concluded that the dimension of an accurate model of surface reflectance was on the order of five or six.

Finite-dimensional linear models are an important tool in color constancy. There is a large collection of algorithms that follow rather naturally from the approach. Some algorithms exploit the properties of the linear spaces involved (Maloney (1984); Maloney and Wandell (1986); Wandell (1987)). Illumination can be inferred from: reference objects (Abdellatif *et al.* 2000); specular reflections (Judd (Judd 1960) writing in 1960 about early German work in surface color perception refers to this as “a more usual view”; recent work includes (D'Zmura and Lennie 1986, Flock 1984, Klinker *et al.* 1987, Lee 1986)); the average color (Buchsbaum 1980, Gershon 1987, Gershon *et al.* 1986); and the gamut (Forsyth (1990), Barnard (2000), Finlayson and Hordley (1999), (2000)).

The structure of the family of maps associated with a change in illumination has been studied quite extensively. The first work is due to Von Kries (who didn't think about it quite the way we do). He assumed that color constancy was, in essence, the result of independent lightness calculations in each channel, meaning that one can rectify an image by scaling each channel independently. This practice is known as Von Kries' law. The law boils down to assuming that the family of maps consists of diagonal matrices. Von Kries' law has proved to be a remarkably good law (Finlayson *et al.* 1994*a*). Current best practice involves applying a linear transformation to the channels and then scaling the result using diagonal maps (Finlayson *et al.* (1994*a*), (1994*b*)).