Estimating Scene Properties from Color Histograms

Carol L. Novak Steven A. Shafer

November 1992

CMU-CS-92-212

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597. C. L. Novak was supported by a fellowship from the Air Force Office of Scientific Research under Contract F49620-86-C-0127.

Keywords: vision and scene understanding, color, modeling and recovery of physical attributes

Abstract

One of the key tools in applying physics-based models to machine vision has been the analysis of color histograms. In the mid-1980s it was recognized that the color histogram for a single inhomogeneous surface with highlights will have a planar distribution in color space. It has since been shown that the colors do not fall randomly in a plane, but form clusters at specific points. The shape of the histogram is related not only to the illumination color and object color, but also to such non-color properties as surface roughness and imaging geometry.

We present here an algorithm for analyzing color histograms that yields estimates of surface roughness, phase angle between the camera and light source, and illumination intensity. These three scene parameters are related to three histogram measurements. However the relationship is complex and cannot be solved analytically. Therefore we have developed a method for estimating these properties by interpolating between histograms that come from images of known scene properties.

We have also shown how roughness and imaging geometry affect the recovery of illumination color from highlight analysis. Our estimates of the roughness and phase angle allow us to make a better estimate of illumination color. This in turn will give better estimates of object colors to achieve color constancy. Our method for estimating scene properties is very fast, and requires only a single color image.

1 Introduction

Color histograms have long been used by the machine vision community in image understanding. Color is usually thought of as an important property of objects, and is often used for segmentation and classification. Unfortunately color is not uniform for all objects of a given class, nor even across a single object. Color variation has come to be expected in images, and vision researchers have been working on modeling this variation.

The earliest uses of color histograms modeled the histogram as a Gaussian cluster in color space [Haralick and Kelly, 1969]. In 1984 Shafer showed that for dielectric materials with highlights, the color histogram associated with a single object forms a plane [Shafer, 1984]. This plane is defined by two color vectors: a body reflection vector and a surface reflection vector. Every pixel's color is a linear combination of these two colors. In 1987 Klinker and Gershon independently observed that the color histogram forms a T-shape or dog-leg in color space [Klinker *et al.*, 1987], [Gershon, 1987]. They showed that it is composed of two linear clusters, one corresponding to pixels that exhibit mostly body reflection and one corresponding to pixels that exhibit mostly surface reflection. This T-shape made it possible to identify characteristic body reflection and illumination colors.

In previous work, we showed that color histograms have identifiable features that depend in a precise mathematical way upon such non-color scene properties as surface roughness and imaging geometry [Novak and Shafer, 1992]. In this paper we show that three scene properties—the illumination intensity, the roughness of the surface, and the phase angle between camera and light source—may be recovered from three measurements of the histogram shape.

However, the functions that relate the scene properties to the histogram measurements are interdependent and highly non-linear, involving trigonometric and exponential functions. Since an analytic solution is not possible, we have developed a method that interpolates between data from a lookup table. The lookup table covers a range of scene parameters, and records measurements of the histograms that result from each combination of those scene parameters. Then a polynomial function is fitted to the data to approximate the relationships between scene parameters and histogram shape.

Our work has also shown how the colors observed in a highlight depend upon more than just the scene colors, but also upon the surface roughness and the imaging geometry. Our estimates of these scene parameters allow us to improve our initial estimate of the illumination color. This along with the estimate of the illumination intensity allows us to discount the effect of the illuminant to recover estimates of the object's reflectance. Section 2 briefly explains the relationship between the color histogram features and the various scene parameters. Section 3 presents an algorithm to compute estimates of these parameters from the histogram. It describes the development of our algorithm for an ideal camera. Section 4 shows how the algorithm may be extended to accommodate more realistic camera characteristics.

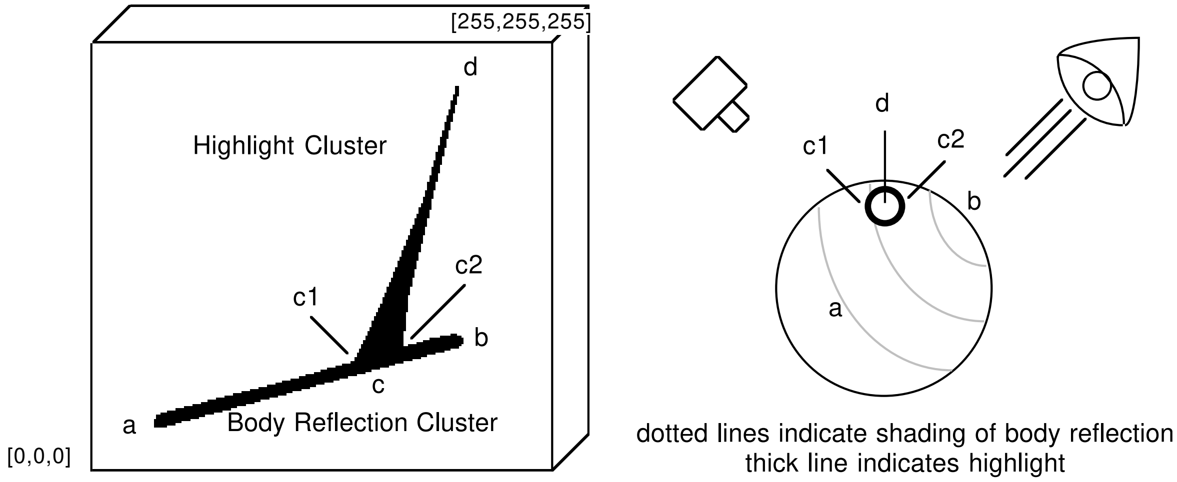

Figure 1: Histogram of an object

2 Understanding Color Histograms

When we talk about the color histogram, we mean a distribution of colors in the three-dimensional RGB space. For a typical imaging system with 8 bits for each color band, there are 256^3 "bins" into which a pixel may fall. In this work, we only consider whether a bin is full or empty. We do not use a fourth dimension to display the number of pixels which have a particular RGB value. A fourth dimension would be difficult to visualize, but more significantly would also be dependent on such things as object size and shape.

In our work we use the "Dichromatic Reflection Model" which states that the light L reflected from a dielectric object is the sum of two components: a surface component L_s and a body component L_b [Shafer, 1984].

$$L(\lambda, \theta_i, \theta_r, \theta_p) = L_b(\lambda, \theta_i, \theta_r, \theta_p) + L_s(\lambda, \theta_i, \theta_r, \theta_p)$$

Each of the two components is a function of the wavelength of light (λ) and the angles of incidence (θ_i), reflection (θ_r), and phase angle (θ_p). The Dichromatic Reflection Model further states that each component in turn may be separated into a color term c that depends only on λ, and a geometric term m that depends only upon θ_i, θ_r, and θ_p.

$$L(\lambda, \theta_i, \theta_r, \theta_p) = m_b(\theta_i, \theta_r, \theta_p)c_b(\lambda) + m_s(\theta_i, \theta_r, \theta_p)c_s(\lambda)$$

Figure 1 contains a sketch of a typical color histogram for a dielectric surface illuminated by a single light source. As labeled, the histogram has two linear clusters of pixels: the body reflection cluster and the highlight cluster. The first of these clusters extends from the black corner of the cube (point **a**) to the point of maximum body reflection (point **b**). The other cluster starts somewhere along the body reflection cluster (point **c**) and extends to the highlight maximum (point **d**).

2.1 The Body Reflection Cluster

The linear cluster that we call the body reflection cluster corresponds to pixels that exhibit mostly body reflection with very little surface reflection. If there is no ambient illumination in the scene, this cluster begins at the black point of the color cube (point **a** in figure 1), corresponding to points on the surface whose normal is 90 degrees or more away from the direction of the illumination. The point at the other extreme of the body reflection cluster (point **b**), corresponds to the largest amount of body reflection seen anywhere on the object. If we assume that the body reflection component is Lambertian, the magnitude term will obey the relation

$$m_b \;=\; \gamma B_b \cos(\theta_i) \tag{1}$$

where θ_i is the angle of illumination incidence. The gain of the camera in converting photons measured by the CCD array into pixel values is represented by γ. The brightness of the body reflection is represented by the term B_b. This factors in both the reflectance of the object (albedo) and the intensity of the light.

In the Lambertian model the magnitude of the body reflection is proportional to the cosine of the incidence angle, so pixels located half-way along the body reflection cluster would correspond to surface points with normals $\cos^{-1}(1/2)$ or 60 degrees away from the illumination direction. If the object exhibits all possible surface normals, the body reflection cluster will be full length and densely filled. If the object is composed of a small number of flat surfaces, there will be gaps in the body reflection cluster. For this paper we will assume that objects we are looking at have a broad, continuous distribution of surface normals.

A vector fitted to the body reflection cluster (from point **a** to **b**) will point in the direction of the body reflection color, which is the product of the object color and the illumination color. Once the illumination color has been determined from analysis of the highlight, the object color alone may be calculated by dividing out the influence of the illumination, as proposed in some color constancy methods [D'Zmura and Lennie, 1986],[Healey and Binford, 1987],[Klinker, 1988],[Tominaga and Wandell, 1989].

If we assume that there is some point on the object which has a surface normal pointing directly at the light source, and which is visible to the camera, then at that point $\cos(\theta_i) = 1$. This means that the length of the fitted vector (the magnitude $|\vec{ab}|$) corresponds to the gain γ times the object's apparent brightness B_b. If the intensity of the illumination could also be recovered from highlight analysis, then the albedo of the object could be separated out from the object's apparent brightness (assuming that the gain of the camera had been calibrated). Otherwise it will be impossible to tell a bright light shining on a dark surface from a dim light shining on a bright surface. Fortunately highlights provide an invaluable clue to distinguishing between these cases.

2.2 The Highlight Cluster

The cluster of pixels we call the highlight cluster corresponds to pixels that show a non-negligible amount of surface reflection. This corresponds exactly to the area of the image that we would call the highlight. In the histogram, the highlight cluster starts where it intersects with the body reflection cluster (point **c**) and extends upwards from there to the brightest point of the highlight (point **d**). For many shiny objects, the highlight is so bright that the highlight cluster is clipped at the white point of the color cube where the highlight has saturated the camera [Klinker, 1988].

In this presentation we use the Torrance-Sparrow model of scattering [Torrance and Sparrow, 1967]. This models a surface as a collection of tiny facets, each of which may have a local surface normal that is different from the global surface normal. The distribution of facet normals is modeled as Gaussian, with σ describing the standard deviation. Smooth surfaces will have a small standard deviation while rougher surfaces will have a larger standard deviation. The facets are larger than the wavelength of visible light, but too small to be seen as texture. We will assume that the facet size is a constant for the surfaces we are interested in.

The equation that we use for scattering gives the amount of surface reflection as

$$m_s = \gamma B_s \frac{F\, G(\theta_i, \theta_r, \theta_p)\, C}{\sigma \cos(\theta_r)} \exp\left(-\frac{\theta_s^2}{2\sigma^2}\right) \qquad (2)$$

where θ_s is the off-specular angle and θ_r is the angle of reflectance. B_s is the intensity of the illumination, γ is the camera gain, and C is a constant that includes the facet size (a variable in the original Torrance-Sparrow model). G is an attenuation factor that depends upon geometry and which comes into play at grazing angles. (G is a complicated function of incidence angle, reflectance angle, and phase angle and we will not reproduce it here; see [Torrance and Sparrow, 1967] for details).

F is the Fresnel coefficient that describes the percentage of the light that is reflected at the interface; it is a function of geometry, wavelength, polarization state, and index of refraction of the material in question. However it is very weakly dependent on incident angle and wavelength (over the visible range), so we will follow the Neutral Interface Reflection model and assume that it is constant for a given material [Lee, 1988]. Furthermore, for a wide range of plastics and paints, the indices of refraction are very nearly identical. Henceforth we will assume that materials have an index of refraction of 1.5, corresponding to 4.0% Fresnel reflectance.

2.2.1 Length of Highlight Cluster

When looking at highlights on a variety of surfaces, we quickly observe that highlights are brighter and sharper on some surfaces, while they are dimmer and more diffused on other surfaces. Very shiny surfaces exhibit only a tiny amount of scattering of the surface reflection, whereas very matte surfaces have a great deal of scattering. This scattering of surface reflection is a result of the optical roughness of the surface.

We see from equation 2 that the sharpness of the peak is determined by the standard deviation σ, and that the height of the peak is inversely proportional to σ. Intuitively this makes sense, since surface reflection scattered over a very small area will be more “concentrated.” A smooth object will have a small standard deviation of facet slopes, σ, resulting in a long highlight cluster. A rough

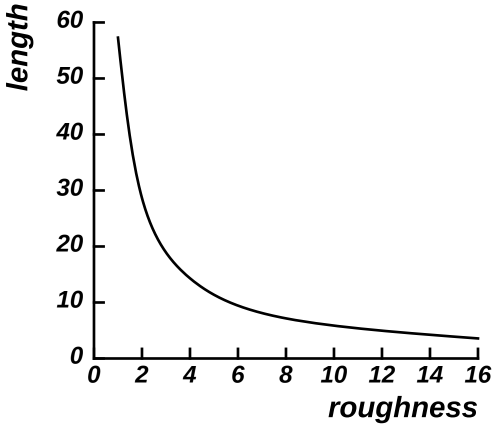

Figure 2: Dependence of highlight cluster length on object roughness

object will have a large σ, and so will exhibit a shorter cluster. Figure 2 shows a plot of the length of the highlight cluster vs. the object's roughness for simulated images where all other factors have been held constant. The horizontal axis indicates the standard deviation of facet angles (in degrees) which is our measure of roughness. The vertical axis indicates the Euclidean distance in RGB space from point **a** to **b**.

Equation 2 indicates that the intensity of the light source B_s also affects the magnitude of the surface reflection, and thus the length of the highlight cluster. It is obvious from equation 2 that the length is directly proportional to this brightness. Since both the brightness of the light source and the roughness of the surface affect the length of the highlight cluster, a question arises whether it is possible to distinguish a bright light shining on a rough surface from a dim light shining on a smooth surface. A feature of the histogram that allows us to distinguish between these two cases will be described in section 2.2.2.

The graph in figure 2 was calculated for the imaging geometry where the light source and camera are separated by zero degrees. However, equation 2 predicts that the imaging geometry will have an effect upon highlight magnitude, as indicated by the $\cos(\theta_r)$ term in the denominator and the attenuation term G in the numerator. The length of the highlight cluster increases as the phase angle between the camera and light source is increased. The effect is small but noticeable, so imaging geometry must be considered to make an accurate estimate of surface roughness.

2.2.2 Width of Highlight Cluster

Another difference between histograms for smooth and rough surfaces is the width of the highlight cluster where it meets the body reflection cluster (the distance from point $\mathbf{c}_1$ to point $\mathbf{c}_2$ in figure 1). The highlight cluster will be wider for rougher surfaces, and narrower for smoother surfaces. This is because rougher objects will scatter surface reflection more widely, over a larger number of reflectance angles.

In the color histogram, a noticeable amount of surface reflection results in pixels that are

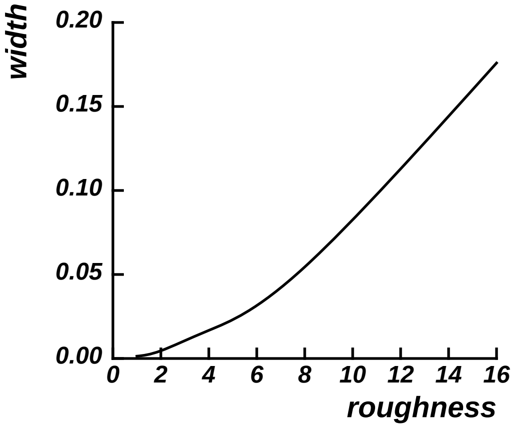

Figure 3: Dependence of highlight cluster width on object roughness

displaced from the body cluster in the direction of the illumination color. If we take any highlight pixel and project along the surface color vector onto the body reflection vector, we can tell how much body reflection is present in that pixel. If we consider all the pixels in the highlight area of the image and look at how much body reflection is in each of them, we will obtain some range of body reflection magnitudes. If the surface is very smooth with a sharp, bright highlight, that range will be small. However if we consider a rougher object with a more diffused highlight, the range of body reflection magnitudes will be larger since the highlight is spread over a larger number of surface normals.

We simulated objects with different roughness values and identified those pixels that showed surface reflection. We calculated the body reflection for each of these points and computed the variation. This variation was divided by the overall length of the body reflection vector to yield a fraction (the length of $\overline{c_1c_2}$ divided by the length of $\overline{ab}$ in figure 1). A fraction of 0.5 would mean that the highlight cluster's base extended across half the length of the body reflection cluster. Figure 3 shows how the highlight cluster width varies with the surface roughness if a point light source is used.

The brightness of the illumination B_s will also have some effect on the the width of the highlight cluster. As the light intensity is increased, points on the surface that had amounts of surface reflection too small to be noticed may become bright enough to be included with the highlight pixels. Clearly the width will grow as the light intensity grows. However the growth is very slow. Although roughness affects both the length and the width of the highlight cluster, changes in illumination intensity affect primarily the length. Therefore it should be possible to distinguish a bright source illuminating a rough object from a dim source illuminating a shiny one.

Although the width of the highlight cluster does not depend upon the object's size and shape, it does depend upon the imaging geometry. To see why this is so, imagine a highlight that spreads 15 degrees in every direction from its maximum. If the camera and light source are separated by 30 degrees, the perfect specular angle will be at 15 degrees with respect to the illumination direction. The highlight will spread over points with surface normals ranging from 0 degrees to 30 degrees. (For ease of explanation, we will ignore the influence of the $1/\cos(\theta_r)$ term.) The amount of body reflection at these points will vary from $cos(0) = 1.0$ to $cos(30) = 0.87$, a width of 0.13. If the camera and light source are separated by 90 degrees, the perfect specular angle will be at 45 degrees, with the highlight spreading from 30 degrees to 60 degrees. Then the amount of body reflection will vary from $cos(30) = 0.87$ to $cos(60) = 0.50$, a width of 0.37.

For the case of highlight cluster width, the measurement is very sensitive to different viewing geometries, so the phase angle between the camera and light source must be known or estimated somehow. Thus it is particularly fortuitous that such an estimate can be made right from the histogram itself, as we will now describe.

2.2.3 Intersection of Clusters

When we introduced the diagram in figure 1, we described the highlight cluster as beginning "somewhere" along the body reflection cluster. Now we will show how to pinpoint the location. The distance along the body reflection cluster where the two clusters intersect (the length of $\overline{ac}$ divided by the length of $\overline{ab}$ in figure 1) shows the amount of body reflectance at those points on the surface that are highlighted. Assuming that body reflection is Lambertian, the amount of body reflection is proportional to the cosine of the incidence angle θ_i. If the two clusters intersect at the maximum point on the body reflection cluster, it means the highlight occurs at those points that have the maximum amount of body reflection, where surface normals point directly at the light source. If the two clusters meet halfway along the body reflection cluster, the highlight must occur at points with surface normal pointing $\cos^{-1}(1/2)$ or 60 degrees away from the illumination direction.

If the body reflection is Lambertian, it does not depend in any way upon the angle from which it is viewed. Thus the body reflection does not tell us anything about the camera direction. However, the surface reflection is dependent upon both the illumination and camera directions. If we ignore for a moment the $1/\cos(\theta_r)$ term in equation 2, we see that the maximum amount of surface reflection will occur at those points on the surface where the angle of incidence equals the angle of reflection. Thus if the highlight occurs at a point where the surface normal faces 10 degrees away from the light source direction, the light source and camera must be 20 degrees apart with respect to that point on the surface.

It does not matter whether the object has one highlight or many. If the object is small compared to the distance to the light source and camera, the highlight or highlights will always occur at points with the same surface normal for a given imaging geometry. Figure 4 shows what happens when we graph intersection vs. imaging geometry. The horizontal axis shows the phase angle between the light source and the camera with respect to the object. The vertical axis is the intersection ratio $\overline{ac}/\overline{ab}$.

The $1/\cos(\theta_r)$ term in equation 2 means that the maximum amount of surface reflection will not always occur precisely at the perfect specular angle. This is particularly true of rougher surfaces where the highlight is spread over a wide range of reflectance angles so that $1/\cos(\theta_r)$ varies significantly. This causes the "off-specular peaks" described in [Torrance and Sparrow, 1967]. The

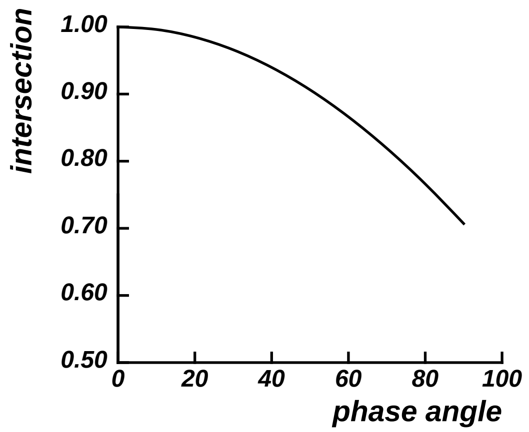

Figure 4: Dependence of intersection on imaging geometry

result is that the intersection is very slightly dependent upon the surface roughness.

2.2.4 Direction of Highlight Cluster

The highlight cluster is usually long and narrow in shape and a vector can be fitted to it (from point **c** to **d**). Klinker argued that this vector will usually correspond closely to the surface reflection color [Klinker, 1988]. This is true for smooth objects where the highlight has a small area, and for imaging geometries where the body reflection changes slowly over that area. In this case, the amount of body reflection at the base of the highlight cluster and the amount at the tip varies by a small amount.

On the other hand, if the object is optically rough and the highlight occurs on a part of the object where the cosine of the incidence angle changes more rapidly, then the amount of body reflection at the base of the highlight cluster may vary significantly from the amount at the tip. This has the effect of skewing the highlight cluster away from the direction of the illumination color, toward the body reflection color. The estimate of the illumination color made from fitting a vector to this cluster will be somewhat inaccurate.

We can visualize this phenomena by projecting the histogram into the plane defined by the body reflection color and surface reflection color. We simulated dichromatic reflection for objects with the same body color but with different roughness values. Figure 5 shows a cross section of the histograms that result. The horizontal direction is the direction of increasing amounts of body reflection; all three histograms fall right on this line for body reflection pixels. The vertical direction is defined by increasing amounts of surface reflection. For surfaces with a standard deviation facet angles of 4°, a vector fitted to the highlight cluster will point exactly in this direction. A vector fitted to the highlight cluster when σ equals 8 will deviate slightly from the vertical direction. In the extreme case, where σ equals 16, the vector will deviate dramatically.

In the color histogram, the vectors describing the body reflection color and the illumination color are not generally perpendicular. The angle between them depends upon the hue difference

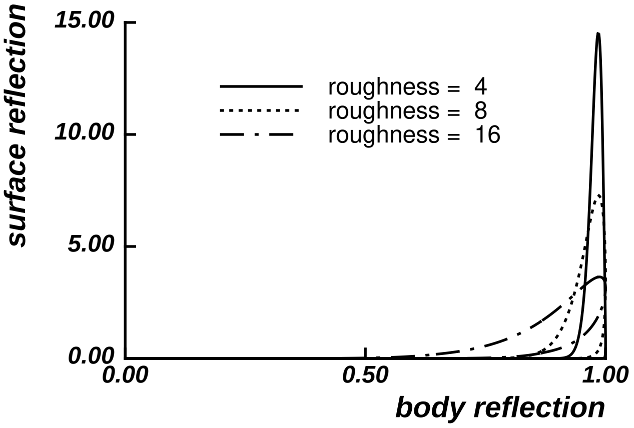

Figure 5: Dependence of highlight cluster direction on object roughness

between the two colors, which is not known in advance. If the vector fitted to the high light cluster does not point exactly in the direction of increasing amounts of surface reflection, the estimate of illumination color will be off by some amount. This in turn will bias the estimate of the object color which is obtained by dividing the body reflection color by the illumination color.

Automatic fitting of a vector to highlight clusters to estimate illumination color has been demonstrated [Klinker, 1988]. The vector fitted to the highlight cluster (from point **c** to point **d**) is a good first estimate of the illumination color, but we now know that it may be skewed. If we know the surface roughness and imaging geometry, we can calculate the amount of skewing and compensate for it.

Recently Wolff has shown that the Lambertian model is somewhat inaccurate for body reflection from many real surfaces, particularly at extreme reflectance angles; he has proposed a different model to replace it [Wolff, 1992]. Also, some researchers prefer to use the Beckmann-Spizzichino model [Beckmann and Spizzichino, 1963], or a combination of the Beckmann-Spizzichino and the Torrance-Sparrow models [Nayar *et al.*, 1991], to describe surface scattering. While a choice of different models would change the exact positions of the graphs in this section, the general relationships would remain the same. These dependencies of the color histogram upon scene parameters are a consequence of the different geometric distributions as well as the different hues of the two components of reflection.

In sections 2.2.1 and 2.2.2 we showed that the roughness of the object affects the length and width of the highlight cluster, but that there is some dependence on imaging geometry. Then in section 2.2.3 we showed that the imaging geometry determines the intersection of the two clusters, but that there is some dependence upon roughness. Furthermore, the intensity of the illumination affects the length and width of the highlights cluster as well, although in different ways. The degree of dependence of each histogram measurement upon the scene parameters is characterized in figure 6.

In this section we have shown that the direction of the highlight cluster can be skewed away from the direction of the illumination color, depending upon the amount of roughness and the

	Roughness	Phase Angle	Illumination Intensity
Length	strong	weak	strong
Width	strong	strong	weak
Intersection	weak	strong	none

Figure 6: Dependence of histogram features upon scene parameters

imaging geometry. The highlight cluster length, width and intersection are defined with respect to the cluster direction, so measurements of these features also depend upon the estimate of the illumination color. If that direction changes, the other histogram measurements will too.

Obviously these factors are all interdependent. Therefore we propose to solve for the roughness, phase angle, and illumination intensity simultaneously, based on the measurements of the histogram.

3 Analyzing Color Histograms

Section 2 described the relationship between the shape of the color histogram and various scene properties. Understanding the relationship is the first step in analyzing color histograms. The next step is figuring out how to actually exploit the histogram to recover quantitative measures of the scene properties.

In this section we assume an ideal sensor that has highly desirable properties. This kind of camera, which is unobtainable in practice, shows the limit of what can be obtained under the best possible imaging conditions. In section 4 we will show how the requirements may be relaxed for the case of a more realistic camera.

We assume here that our camera has a linear response, so that increasing amounts of light entering the camera will correspond to proportional increases in the pixel values. In practice, many cameras are reasonably linear in their response. Even when they are not, a linear correction is easy to apply once the non-linearity has been measured [Novak *et al.*, 1990].

In this section we also assume that the camera is linear over the range of all possible pixel values that might occur in our images. Unfortunately no real camera has this property; they all have some maximum value that can be measured. After this maximum has been reached, increasing the brightness of a point will not cause an increase in the corresponding pixel value. The result is that the pixel value is "clipped" at the maximum value of the camera. Since it is precisely pictures with highlights that we are interested in, the assumption of unlimited dynamic range is much less realistic than the linearity assumption. In section 4, we will show how this assumption may be removed.

A final assumption made about the ideal sensor is that it does not suffer from any noise problems. Shot noise, dark current noise, and amplifier noise are three sources of noise that occur in real CCD cameras [Healey and Kondepudy, 1991]. In section 4 we will show how these effects might be modeled and taken into consideration in the analysis of histograms.

3.1 Knowns and Unknowns

The known image parameters which can be measured from the color histogram are the body reflection cluster's length and direction; the highlight cluster's length, width, and direction; and the intersection point of the two clusters. This gives four scalar values and two vector quantities. They will be referred to by the following variables:

- l - length of highlight cluster
- w - width of highlight cluster
- i - intersection of two clusters

- b - length of body reflection cluster
- $\vec{d_s}$ - direction of highlight cluster
- $\vec{d_b}$ - direction of body reflection cluster

The unknown scene parameters which we would like to recover from the histogram can also be divided into scalar values and vector quantities. The scalar values are the surface roughness, the phase angle between the light source and camera directions, the illumination intensity, and the surface albedo. The vector quantities are the chromaticity of the illumination and that of the object. These variables are:

- σ - optical roughness
- θ_p - phase angle
- B_s - illumination intensity
- B_o - object albedo
- $\vec{c_s}$ - chromaticity of light source
- $\vec{c_o}$ - chromaticity of object under "white" light

For convenience, we have separated the illumination and object reflectance into intensity components and chromatic components. As we mentioned in section 2.1 the object's chromaticity $\vec{c_o}$ may be recovered in a straightforward way from the direction of the body reflection cluster $\vec{d_b}$ and the color of the light source $\vec{c_s}$. The red component of the body reflection vector is divided by the red component of the light source color, the green component divided by the green component, and the blue component divided by the blue component; the result is normalized to length 1.

$$\vec{c_o} = \frac{[d_b^R/c_s^R,\ d_b^G/c_s^G,\ d_b^G/c_s^B]}{||\,[d_b^R/c_s^R,\ d_b^G/c_s^G,\ d_b^G/c_s^B]\,||}$$

The albedo of the object B_o is also recovered in a straightforward manner from the length of the body reflection cluster b and the illumination intensity B_s. From equation 1 the body reflection magnitude is

$$m_b = \gamma B_b \cos(\theta_i)$$

where B_b is the object's apparent brightness. The body cluster length b is equal to the maximum value of m_b, when $cos(\theta_i) = 1$. Therefore $b = \gamma B_b$. Since the apparent brightness of the object is the product of the object albedo B_o and the illumination intensity B_s, the albedo is

$$B_o = B_b/B_s = b/(\gamma B_s)$$

For the remaining unknowns the situation is not so simple. From section 2 we know that the length l is related to surface roughness and illumination intensity, but is also dependent upon imaging geometry. The remaining knowns $(l, w, i, \vec{d_s})$ and unknowns $(\sigma, \theta_p, B_s, \vec{c_s})$ will be examined in detail in the next few sections.

3.2 Analytic Solution

Equations 1 and 2 describe the amounts of body and surface reflection m_b and m_s as a function of imaging geometry and light intensity B_s. Equation 2 also shows how the amount of surface reflection at a given point varies with the roughness of the surface σ.

Unfortunately it is not possible to directly solve for these scene parameters from the histogram measurements. The length of the highlight cluster indicates the maximum amount of surface reflection seen anywhere on the object. For given values of σ, θ_p, and B_s the length l may be calculated

$$l \;=\; MAX\left[\gamma B_s \frac{F\,G(\theta_i, \theta_r, \theta_p)\,C}{\sigma\cos(\theta_r)} \exp\left(-\frac{\theta_s^2}{2\sigma^2}\right)\right] \text{ over all values of } \theta_i,\, \theta_r,\, \theta_s \tag{3}$$

If it were not for the presence of G in the denominator and $\cos(\theta_r)$ in the denominator, the length would be easily derived. If they disappeared from the equation, the maximum value of the surface reflectance would occur at the perfect specular angle, when θ_s is zero.

However, even if all terms besides σ were assumed to be constant and equal to 1, the equation would reduce to

$$l = \frac{1}{\sigma}\exp(-\frac{1}{\sigma^2})$$

which has no analytic solution for σ. Clearly a direct solution of the complete surface reflection equation cannot be obtained.

3.3 Ideal vs. Measured

Although equation 3 has no analytic solution, it might be possible to solve it iteratively, through some sort of search (for example by gradient descent). However that assumes the length l can be accurately measured from the histogram. In section 2.2.1, the length is measured from the tip of the highlight cluster to its base *along the direction of the highlight color*. Unfortunately the highlight color is not typically known in advance. As described in section 2.2.4, the highlight color may be estimated by fitting a vector to the pixels that form the highlight cluster. However that cluster may be skewed.

This is shown in figure 7 which is the histogram of a simulated rough object illuminated with white light. The dotted “measured” line shows the direction calculated for the best fit vector to the highlight cluster, and what happens when we project from the brightest highlight pixel down along that vector. The “ideal” indicates the direction of the actual illumination color. The skewing causes the measured length to be longer than it would have been if the correct illumination color had been known.

This contrasts the concept of “ideal” length that we would like to obtain from the histogram, with the “measured” length that can actually be recovered without a priori knowledge about the

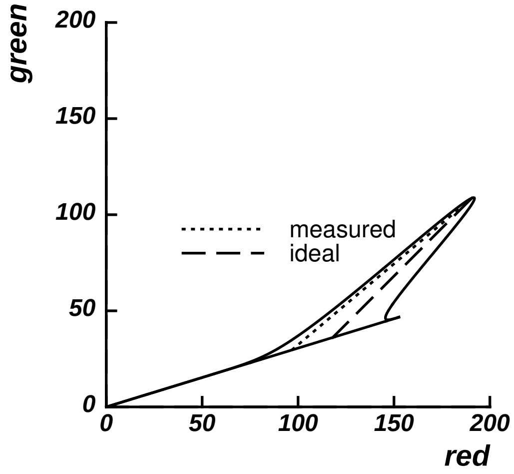

Figure 7: Skewed Length Measurement

illumination color. All the graphs presented in section 2 that relate histogram measurements to scene parameters used were obtained by knowing the illumination color.

Figure 8 shows how both the “ideal” length and “measured” length vary with roughness for the case where the camera and light source are separated by 20 degrees. For smooth surfaces both values show an inverse relationship between length and roughness. As the roughness increases, the deviation between the two curves grows; eventually the measured length begins *increasing* again. If only the measured length value is available, it would be difficult to tell a very smooth surface from a very rough one.

A similar story applies to the measurement of width and intersection point. When the highlight is skewed, the values measured from the histogram will be different from the ideal values shown in the graphs in section 2. The intersection point is measured by taking the brightest pixel in the highlight cluster and projecting it onto the body reflection vector. This projection is done along the vector that is assumed to be the highlight color. The point of intersection with the body reflection cluster is supposed to represent the amount of body reflection at the maximum highlight point. However, if the highlight cluster is skewed towards the body reflection cluster, the highlight pixel will be projected too far down, yielding an intersection estimate that is too small.

By the same token, the width measurement is also affected by the skewing. The width measurement represents the range of body reflectance values which show a highlight compared to the range of all body reflectance values. However, a wrong estimate of the illumination color will give incorrect estimates of the amount of body reflection at each point. The calculated variation may be larger or smaller than than the correct width value.

In the absence of *a priori* information, we can only measure what is available in the histogram. How do we derive the ideal values from the ones that are measured? Once that is done how do we recover the image parameters that we are interested in?

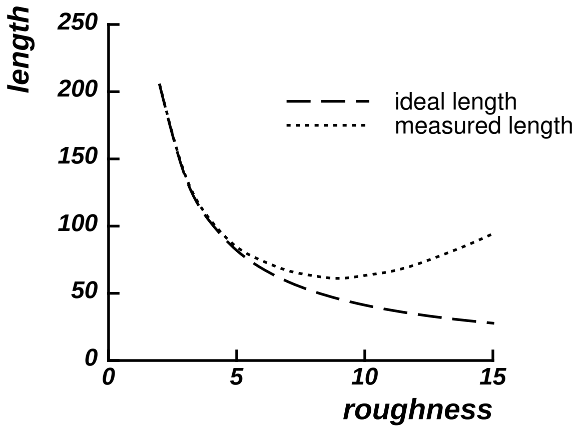

Figure 8: Ideal vs. Measured Length

3.4 Approximate Solution

Our approach is to recover scene parameters by an approximate method, *directly* from the initial histogram measurements. Therefore, we do not need to recover "ideal" histogram values from the "measured" ones. The ideal values of the histogram are a useful abstraction since their relationship to scene parameters is easy to explain. However they cannot be obtained from the histogram without knowledge of the illumination color.

Figure 9 shows the variation in the measured length as roughness and phase angle are changed. Each value of length describes a contour within the space of roughness and phase angles. Given a length measurement from a histogram, the associated scene parameters must lie somewhere on that contour. When illumination intensity is considered along with roughness and phase angle, these three scene parameters form a three dimensional parameter space. A length measurement would then describe a two dimensional surface within that space, showing the possible roughness, phase angle, and light intensity values that could give rise to a histogram with that length highlight cluster.

The intersection and width measurements will also describe surfaces within the parameter space. The hope is that the surfaces for each of the histogram measurements will intersect at a single point in the parameter space, making it possible to recover unique values for surface roughness, phase angle, and illumination intensity. So obvious questions are: how can we generate these contours of equal-length, equal-width, and equal-intersection; and do these contours intersect to give unique solutions?

Section 3.2 pointed out that there is no analytic solution to generate the contours. However, the

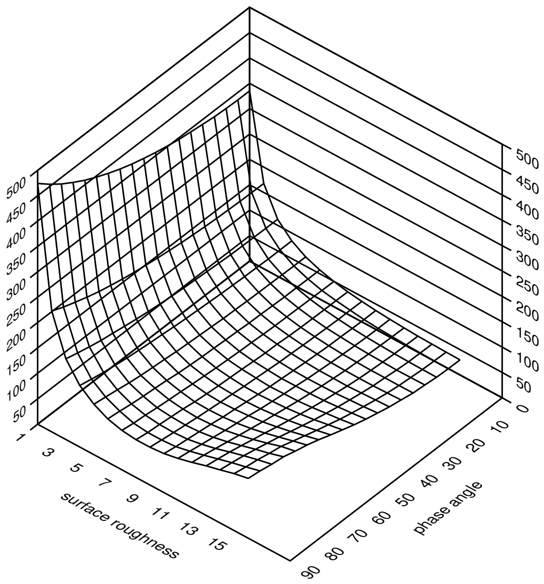

Figure 9: Variation in cluster length with changes in roughness and phase angle

graph in figure 9 shows how the highlight cluster length varies with roughness and phase angle at discrete points. These values come from simulating an object with those parameters and then measuring the length, width and intersection of the highlight cluster in the the resulting color histogram. By simulating a large range of roughness values, phase angles, and illumination intensities, we create lookup tables of length, width, and intersection measurements. The lookup table is searched to find the scene parameters that correspond to a given set of histogram measurements.

The more interesting question is whether a unique solution exists for a given triple (l, w, i). If some triple has more than one solution, that means that different combinations of scene parameters can give rise to identical histogram measurements. It also means that a search through the contours in parameter space cannot be guaranteed to converge.

To visualize the distribution of possible (l, w, i) triples, figure 10 shows the space of all possible values for length, width and intersection. The x-axis encodes the range of highlight cluster lengths, the y-axis shows the range of widths, and the z-axis shows intersection. The surface within it is defined by lines of equal roughness in one direction and lines of equal phase angle in another direction. This surface shows which combinations of length-width-intersection are possible for histograms within the simulated range. Points in the space not falling on the surface correspond to histograms that do not make sense.

Although the surface curves around in the L-W-I space, it does not intersect itself anywhere. This means that any triple of length-width-intersection that falls on the surface is associated with a unique set of surface roughness and phase angle values. The only remaining problem is to determine from these histogram measurements where on the surface they will lie.

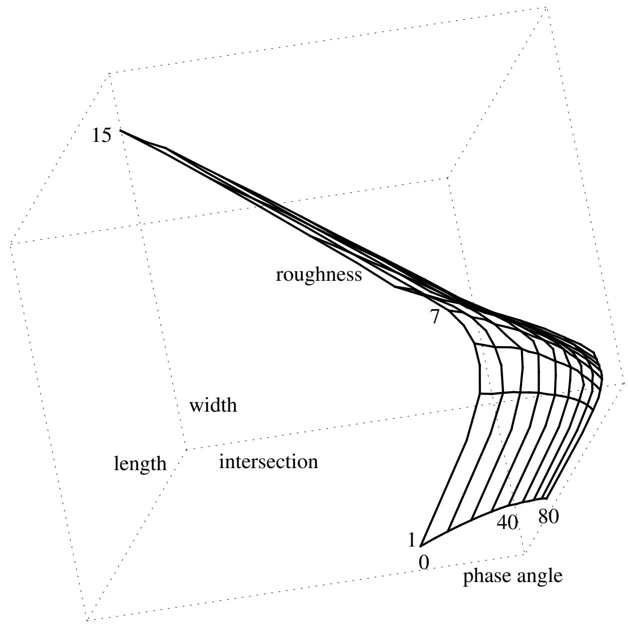

Figure 10: Surface of roughness and phase angles within L-W-I space

	Minimum	Maximum	Increment	Total Used
Roughness	1°	15°	2°	8
Phase Angle	0°	90°	10°	10
Intensity	50 %	100 %	10 %	6
Overall				480

Figure 11: Range of scene parameters used to generate lookup table

3.5 Generating Lookup Tables

A large range of roughness values, phase angles, and illumination intensities are used to create the lookup tables. The ranges are shown in the table in figure 11. The roughness value is the standard deviation of facet angles with respect to the global surface normal, in degrees. The phase angle is the angle between the camera and light source with respect to the object, also in degrees. The light intensity is a percentage of a hypothetical light's maximum output.

For each set of roughness, phase angle, and light intensity values, a simulated object is generated, using the noise free camera model. The histogram associated with the object is automatically separated into body reflection and highlight clusters. The technique is similar to that described in [Klinker, 1988]. Vectors are fitted to each of the clusters.

Once the direction of the highlight cluster has been measured, the vector $\vec{d_s}$ is used to project all highlight pixels onto the body reflection vector $\vec{d_b}$. These projections determine the relative contributions of the vectors in each pixel. Each color pixel $\vec{p}$ in the histogram can then be defined as

$$\vec{p} = m_s \vec{d_s} + m_b \vec{d_b}$$

This is essentially the dichromatic equation, although the highlight cluster direction $\vec{d_s}$ may differ from the actual highlight color. The histogram measurements are then defined simply as

$$\begin{aligned} l &= MAX(m_s) \text{ over all } \vec{p} \\ b &= MAX(m_b) \text{ over all } \vec{p} \\ i &= m_b/b \text{ for that } \vec{p} \text{ with maximum } m_s \\ w &= [MAX(m_b) - MIN(m_b)]/b \text{ over all } \vec{p} \text{ for which } m_s > T \end{aligned}$$

The threshold T can be set to any arbitrary small value for the case of a noise-free sensor. The value will be set more carefully in section 4.1.

3.6 Calculating Roughness, Phase Angle, and Illumination Intensity

The surface shown in figure 10 shows contours of equal roughness and phase angle within the L-W-I space. Once length, width and intersection measurements have been obtained from a histogram, the problem is to determine which contours they fall on. It is unlikely that a random histogram will have measurements that fall exactly on the contours shown. Therefore some sort of interpolation is needed.

Our work uses a polynomial approximation to the surface in L-W-I space. We assume that the roughness can be approximated as a polynomial function of length, width, and intersection measurements of the histogram.

$$\begin{aligned} \sigma &\approx f_n(l, w, i) \\ &\approx A + Bl + Cw + Di + El^2 + Fw^2 + Gi^2 + Hlw + Ili + Jwi + \ldots \end{aligned} \tag{4}$$

The lookup table provides the means for calculating the coefficients of the polynomial. It provides almost 500 sets of histogram measurements and the associated roughness values. Least squares estimation is used to calculate the best fit nth degree polynomial to the data. A fourth degree polynomial is used in our experiments.

Similarly, the phase angle and illumination intensity are also approximated as polynomial functions of the histogram length, width, and intersection.

$$\theta_p \approx g_n(l, w, i) \tag{5}$$

$$B_s \approx h_n(l, w, i) \tag{6}$$

Least squares is used to calculate the coefficients for these polynomials also. The fit of the least squares calculation to the data is shown in the table in figure 12.

Generating the lookup table is obviously very time consuming (about 8 hours on a SPARC II) since it involves calculating almost 500 graphics simulations. However, the table generation and

Function	R squared
$\sigma = f_n(l, w, i)$	0.877
$\theta_p = g_n(l, w, i)$	0.939
$B_s = h_n(l, w, i)$	0.712

Figure 12: Fit of polynomial functions

coefficient calculation only need to be done once and can be done ahead of time. At run-time our system takes a histogram from an image with unknown parameters and automatically separates it into two clusters and measures their dimensions. The polynomial equations are then applied to quickly estimate the roughness, phase angle, and illumination intensity. The run-time portion is very quick, taking less than 3 seconds on a histogram containing about 3600 pixels. If the histogram has already been split into clusters in the process of segmenting the image [Klinker, 1988], the time to calculate the scene parameters is less than 1 second.

To test the polynomial approximations, one hundred test images were simulated and then analyzed by our method. These test images were noise-free, generated with the same idealized camera model used to generate the lookup table. The surface roughness, phase angle, and illumination intensity values used in the test images were chosen by a pseudo-random number generator. The test values were constrained to lie within the ranges used in the lookup table (see figure 11).

The calculated values of σ, θ_p, and B_s were compared with the original values used to generate the image. In almost all cases the calculated values were close to the original ones. However, for 2% of the cases, the values were very obviously wrong. For example, a negative value of roughness or illumination intensity is clearly unreasonable. Fortunately, bad values can be detected automatically by checking to see if recovered values are within the allowable range. Recovered values that fall outside that range indicate that a different method should be used to recover the scene parameters. For example, a lower degree polynomial approximation could be used for these cases. Although the fourth degree polynomial is better at approximating the function over its whole range, it may deviate somewhat at the extremes of the function. In fact the 2% problem cases occur when the roughness is very low (between 1 and 2 degrees) and the phase angle is very large (greater than 75 degrees). The problem disappears if a third degree polynomial is used instead, although the overall error on all cases is slightly higher.

The table in figure 13 shows the results for the remaining 98 cases where the fourth degree polynomial produced reasonable estimates. It shows the average error in recovering the parameter, and also reiterates the step sizes used in the table. The errors are lower than the table resolution, showing the interpolation method is fairly effective.

The results for calculating roughness and phase angle are very good. They show that these non-color parameters may be calculated with reasonably high accuracy just by considering the shape of the color histogram. The error in calculating illumination intensity result is a bit higher, although it still provides a useful estimate. This error is not too surprising, since the R-squared fit of the function to calculate illumination intensity is somewhat worse than for the other two functions

Parameter	Average Error	Resolution of Table
Roughness	1.20°	2°
Phase Angle	4.40°	10°
Intensity	8.18 %	10 %
Cases Considered		98/100

Figure 13: Results on noise free data

(see figure 12).

3.7 Calculating Illumination Chromaticity

As we pointed out in section 2.2.4, the highlight cluster may be somewhat skewed from the direction of the highlight color. The skew is particularly pronounced at certain imaging geometries and when the surface is rough. These two factors determine how much the body reflection changes over the area of the highlight.

Therefore, if we know—or can calculate—the surface roughness and the imaging geometry, we can in turn calculate the amount of highlight skewing. Once the skew is known, its effect can be subtracted from the direction of the highlight cluster to give the true color of the illumination.

Section 3.6 showed how to estimate the roughness and phase angle from the color histogram. These estimates will now be used to estimate the skewing. A similar lookup table approach is used. When the simulations are performed to calculate length, width, and intersection for the range of scene parameters given in figure 11, the skewing of the highlight is also calculated. In the graphics simulation the correct illumination color is obviously known in advance, so the angle between this color and the vector fitted to the highlight cluster is calculated and stored in the lookup table along with the values for length, width, and intersection. Then a polynomial function is used to calculate the skew angle as a function of roughness, phase angle, and illumination intensity:

$$Skew \quad \approx \quad \Lambda_n(\sigma, \theta_p, B_s) \tag{7}$$

The coefficients of the nth degree polynomial function are calculated using a least squares fit to the data in the lookup table. A third degree polynomial was tested and gave an R-squared fit of 0.993.

Once the skew has been calculated, the highlight color $\vec{c_s}$ may be calculated from the measured direction $\vec{d_s}$ using the calculated skew angle. Obviously, if the polynomial functions described in section 3.6 produce bogus estimates of the roughness, phase angle, or illumination intensity, there is little point in plugging them into the equation for calculating *Skew*. In those 2% of the 100 test cases, the program did not attempt to calculate the illumination color. For the remaining 98% test cases, the skew angle was used to calculate the illumination color. The results are shown in the table in figure 14. The error in estimating skew is the difference—in degrees—between the correct skew angle and the skew angle calculated by our method. The correct skew angle is

Average Error	1.73°
Average Value	8.63°
Minimum Value	0.01°
Maximum Value	27.5°
Number of Skews $>$ 1°	69
Cases Considered	98/100

Figure 14: Results in Calculating Skew

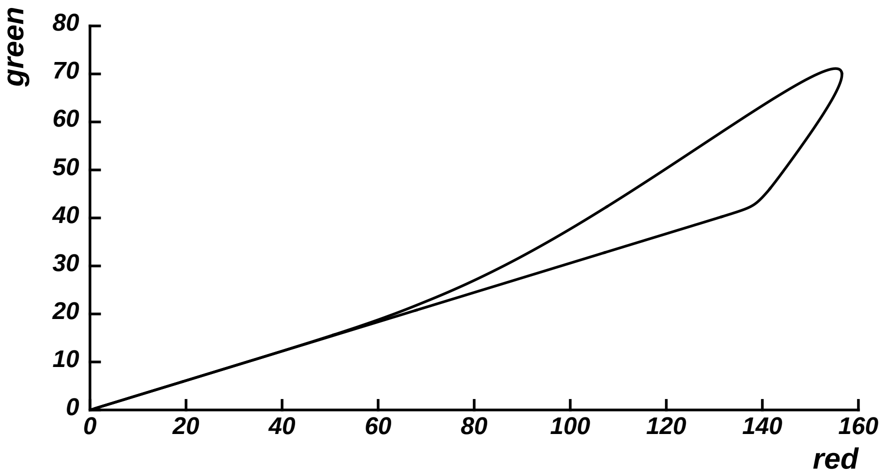

Figure 15: Histogram of simulated image

easily calculated from the illumination color used to generate the test picture. The table shows the average error over the 98 cases considered. It also shows the minimum, maximum and average of the actual skew values. For 69 of the test images, the scene parameters were such that the highlight was skewed by 1° or more.

An example of our algorithm on one of the test simulations is shown here. The test image is a red cylinder under white light ($\vec{c_s} = [0.58, 0.58, 0.58]$). The surface roughness is 12.06° standard deviation of the facet angles; the phase angle between camera and light source with respect to the object is 63.30°; and the illumination intensity is 90% of maximum.

The histogram associated with this image is shown in figure 15. The graph shows the projection of the color histogram into the red-green plane. The program automatically divided the histogram into body reflection and highlight clusters. The vector fitted to the highlight cluster is $\vec{d_s} = [0.81, 0.45, 0.37]$. The length of the highlight cluster was measured as 74.1; the width of the highlight cluster was measured as 0.47 (the highlight cluster extends over slightly less than half the body cluster); the intersection of the two clusters was measured as 0.51 (the brightest point in the highlight cluster was projected onto the body cluster just above the halfway point). These measurements were all obtained using the vector $\vec{d_s}$ to calculate the amount of body reflection and surface reflection.

Simulated Image	$\vec{c_s} = [0.58, 0.58, 0.58]$	$\sigma = 12.06°$	$\theta_p = 63.30°$	$B_s = 90\%$
Histogram Measurements	$\vec{d_s} = [0.81, 0.45, 0.37]$	$l = 74.1$	$w = 0.47$	$i = 0.51$
Recovered Parameters	$\vec{c_s} = [0.59, 0.57, 0.57]$	$\sigma = 11.99°$	$\theta_p = 67.05°$	$B_s = 89\%$

Figure 16: Example results

The direction fitted to the highlight cluster is significantly skewed away from the direction of the actual illumination color. It represents a much redder color than the white illumination color, and so would be a poor estimate of the illumination color. It would also yield an inaccurate estimate of the object color when the influence of the illumination color is divided out of the body reflection color.

Applying polynomial equations 4, 5 and 6 to the length, width and intersection measurements, the program estimated the roughness value as 11.99°, a phase angle as 67.05°, and the illumination intensity as 89% of maximum. We then applied equation 7 to these estimates of σ, θ_p, and B_s, and estimated the skew between the highlight cluster direction and the actual illumination color to be 18.75°. Applying this skew to the cluster direction $\vec{d_s}$ produced an estimate of the light source chromaticity as $\vec{c_s} = [0.59, 0.57, 0.57]$. This is very close to the original white color.

The results for this example are summarized in the table in figure 16. The full algorithm is diagrammed in figure 17.

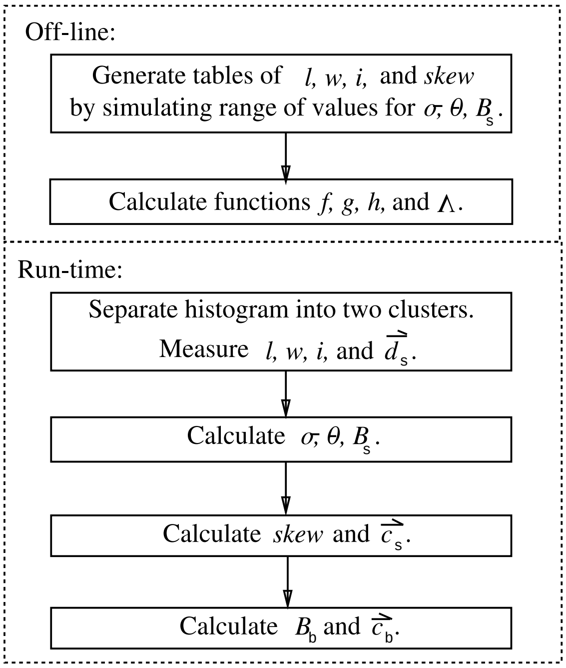

Figure 17: Algorithm for calculating scene parameters

4 Analyzing Histograms with a Realistic Sensor Model

In section 3 we described how to estimate scene parameters from a color histogram if an ideal sensor is used. Specifically, we assumed that the camera used to view objects is noise-free and linear over the range of all possible intensity values. The possibility of clipping at some maximum camera value was not considered in section 3.

However these are not very realistic assumptions. Some forms of noise exist in all real camera systems. And while linearity of response over a limited range is common in many cameras, all real cameras have some maximum value. Clipping at the maximum value is particularly common in pictures with highlights since they are so much brighter than other parts of the scene.

Since we aim to analyze pictures with highlights, and since all real imaging systems have noise, we need to deal with these issues in order to make the algorithm work on real images. Section 4.1 will describe how we have adapted the algorithm to deal with noise, while section 4.2 will propose a method for dealing with the problem of clipping.

4.1 Camera Noise

Real cameras present many limitations for taking accurate images. CCD cameras, which are used by most vision researchers, present many problems such as dark current, shot noise, amplifier noise, line jitter, etc. [Healey and Kondepudy, 1991]. Some of these problems, such as shot noise, are an inherent property of CCD cameras and cannot be removed even with high-quality manufacturing.

We have modeled camera noise as having a Gaussian distribution with a mean of zero. The standard deviation of camera noise σ_c was modeled as 1.275. The camera is assumed to measure eight bits per pixel, giving a range from 0 to 255.

Since our method estimates scene parameters by analyzing color histograms, we consider the effect of Gaussian noise on the histogram's appearance. Under this model, each pixel may be viewed as occupying a probability cloud in color space. This cloud represents the pixel's ideal color value plus some uncertainty due to noise. The size of the cloud depends upon the standard deviation of the noise. If the population of pixels is large, the cloud will be filled according to the distribution of noise. For a Gaussian distribution, the modal pixel will coincide with the ideal pixel value, with two-thirds of the pixels falling within one standard deviation of the ideal value.

One of the effects of noise on the histogram is that it will make the body reflection cluster thicker. A uniformly colored object imaged with an ideal camera would have a body reflection cluster that is a line only one pixel thick. Noise added to ideal pixels will correspond to some pixels that are displaced away from the ideal line. This will result in a thicker body reflection cluster. However, even in the ideal case we needed a threshold for distinguishing highlight pixels from pixels on the body reflection cluster, although this threshold could be arbitrary (see section 3.5). With an explicit noise model, the threshold should be based on the standard deviation of the camera

Parameter	Average Error	Resolution of Table
Roughness	1.08°	2°
Phase Angle	4.56°	10°
Intensity	6.85 %	10 %
Skew	1.39°	-
Cases Considered		99/100

Figure 18: Results on noisy data

noise. We have chosen to set the threshold T at $4\sigma_c$, so over 99% of the noisy body reflection pixels will fall within the body reflection cluster.

Noise added to body reflection pixels will also push some pixels up or down along the direction of increasing amounts of body reflection. Noise added to those pixels which are at the tip of the body reflection cluster (point **b** in figure 1) will result in a cluster that is slightly longer. This will affect the measurement of the length of the body reflection cluster b and the subsequent calculation of the object's albedo B_o (see section 3.1). Fortunately the effect is very small, so we will ignore it.

Noise added to pixels in the highlight cluster will also displace those pixels, causing the highlight cluster to be slightly wider and longer than for the noise-free case. In this case we cannot be so cavalier about the effect, for we are using small differences in the width of the cluster to predict differing values of roughness. Therefore in dealing with noise in the highlight pixels we will take a two pronged attack: we will explicitly account for the noise in generating the lookup tables, and we will try to remove some noise when analyzing each histogram.

When we generated the lookup tables for section 3 we simulated noise-free images and measured the resulting histograms. Now, for the noisy camera model, we simulate images again, this time adding pseudo-random Gaussian noise. The resulting histograms are analyzed, and new values of length, width and intersection are recorded in the lookup table. These measurements will reflect the slight increases in the length and width of the two histogram clusters.

Furthermore, in measuring the histograms we use local smoothing. Each pixel from the simulated image is averaged with its four nearest neighbors before being plotted in the histogram. For pixels falling in the highlight or at its edge, this will have the effect of smoothing out the highlight somewhat, but will also tend to average out the noise. The effect of highlight smoothing on the length, width and intersection measurements will also be recorded in the lookup table.

The table in figure 18 presents the results when the method is tested on 100 simulated images with noise. The algorithm is the same as the one presented in figure 17. Only the lookup tables have been changed, to reflect the noise model of the camera. Comparing the noise-free results in figures 13 and 14 with the results here, we see that the algorithm actually did slightly better with noisy data, except when calculating phase angle. This is a surprising result, but we believe it is caused by the averaging that is done in on the noisy data, which has the effect of smoothing the histogram slightly.

4.2 Camera Clipping

Like noise, clipping is an inherent property of real cameras since all real cameras have some maximum value that they can measure. Many real images with highlights suffer from clipping since highlights are often very bright. When taking real pictures, researchers can often adjust the aperture, light level, or exposure time so that the brightest point in the image is within the dynamic range of the camera. However, highlights on shiny objects can be an order of magnitude brighter than surrounding non-highlight areas, so that if the camera is adjusted to measure highlights properly, the non-highlight areas will be so dim as to be lost in the camera noise.

Therefore any algorithm that hopes to work on real pictures with highlights should take clipping into account. The effect of clipping on the color histogram is easy to model. Any pixels that would have exceeded the maximum value (255 for an 8 bit camera) will be held to that value. As a result, the highlight cluster will not be as long as it otherwise would have been. Clipping usually occurs when the surface roughness has a standard deviation of facet angles of 3° or less.

In analyzing color histograms, clipping is easy to detect. If a pixel has the maximum value in any band, it was almost certainly clipped. However, dealing with the clipping presents a greater problem since the effect upon the length may be dramatic. Fortunately the width measurement is not affected at all, as long as there are some pixels at the edge of the highlight that do not saturate the camera. The intersection measurement can be affected, since it relies upon the projection of the brightest highlight pixel. In the case of clipping, the brightest *non-saturated* pixel is used to calculate the intersection point. Fortunately, the difference is very small.

The length measurement is an important parameter in calculating the scene parameters, along with the width and intersection measurements. In order to calculate the three unknown values of surface roughness, phase angle and illumination intensity, three known histogram measurements are needed. If we are forced to throw out the length measurement, we need to replace it with some other measurement.

We propose to replace the length measurement with another measurement of the cluster shape. The width measurement and the length measurement are a way of characterizing the shape of the highlight cluster. We want to capture this shape since it is related to the relative distributions of body and surface reflection, which in turn are determined by the scene parameters we want to estimate.

When the length measurement is untrustworthy due to clipping, we propose to use a second width measurement. The first width measurement tells how wide the highlight cluster is at the threshold distance T from the body reflection cluster. The second width measurement will show how wide the highlight cluster is at a distance of $3T$ from the body reflection cluster.

When the lookup tables are calculated, we record the length and also whether or not it is clipped. We also record the second width measurement, as well as the first width and intersection measurements previously described. After the lookup table is generated *two* sets of polynomial functions are fitted: one for analyzing histograms that are not clipped, and one set for histograms

Parameter	Average Error	Resolution of Table
Roughness	1.00°	2°
Phase Angle	6.07°	10°
Intensity	6.76 %	10 %
Skew	1.69°	-
Cases Considered		96/100

Figure 19: Results on noisy data when clipping is present

that are. If the second width measurement is referred to as w_2 then the scene parameters may be calculated by

$$\left.\begin{array}{rcl}\sigma & = & f_1(l,w,i)\\ \theta_p & = & g_1(l,w,i)\\ B_s & = & h_1(l,w,i)\end{array}\right\}\quad\text{if the histogram is not clipped}$$

$$\left.\begin{array}{rcl}\sigma & = & f_2(w_2,w,i)\\ \theta_p & = & g_2(w_2,w,i)\\ B_s & = & h_2(w_2,w,i)\end{array}\right\}\quad\text{if the histogram is clipped}$$

The preliminary results we have obtained with this technique are shown in the table in figure 19. The results here are similar to those obtained without clipping in figure 18, although there are more problem cases that had to be eliminated from consideration. We are presently working on techniques that will handle those cases as well.

In order to make the algorithm work well on real images, it would be desirable to obtain the lookup table entries from histograms of real test objects. Images of objects with known roughness values could be used to calibrate the system, since it is possible that surface roughness in real objects is somewhat different from the models.

5 Conclusions

The color histogram of an image is a rich source of information, but it has not been fully exploited in the past. The variation of color was once modeled as a random process with a Gaussian distribution. More recently, the model has become more structured, showing that the distribution of colors on an inhomogeneous object will form a plane in RGB space, and that the pixels fall within a parallelogram defined by the body color and illumination color. Experiments have since shown that the parallelogram is not uniformly filled; rather, the color histogram will show two distinct clusters that meet in RGB space.

We have gone further to show that the direction, length, width, and intersection of these two clusters may by characterized by numerical measurements. The resulting measurements are not just of academic interest, but relate directly to many scene properties. The color variation described by the histogram actually depends on several non-color parameters.

We have shown how these histogram measurements may be used to recover estimates of surface roughness, imaging geometry, illumination intensity, and object albedo. Once these non-color properties have been estimated, they may be used to further refine estimates of illumination color and object color. We have developed a method that interpolates histogram measurements from idealized image data, and shown how that method may be extended for more realistic camera characteristics. Further work is now under way to apply the methodology to real laboratory images. This kind of analysis may be applied to such varied tasks as surface inspection and object recognition.

References

[Beckmann and Spizzichino, 1963] P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. The Macmillan Company, 1963.

[D'Zmura and Lennie, 1986] M. D'Zmura and P. Lennie. Mechanisms of color constancy. *J. Opt. Soc. Am.*, 3(10):1622–1672, October 1986.

[Gershon, 1987] R. Gershon. *The Use of Color in Computational Vision*. PhD thesis, University of Toronto, 1987.

[Haralick and Kelly, 1969] R. M. Haralick and G. L. Kelly. Pattern recognition with measurement space and spatial clustering for multiple images. *Proc. IEEE*, 57:654–665, April 1969.

[Healey and Binford, 1987] G. Healey and T. O. Binford. The role and use of color in a general vision system. In *Proceedings of the ARPA Image Understanding Workshop*, pages 599–613. DARPA, 1987.

[Healey and Kondepudy, 1991] G. Healey and R. Kondepudy. Modeling and calibrating ccd cameras for illumination insensitive machine vision. In *SPIE No. 1614 Conference on Optics, Illumination, and Image Sensing for Machine Vision VI*. SPIE, 1991.

[Klinker *et al.*, 1987] G. J. Klinker, S. A. Shafer, and T. Kanade. Using a color reflection model to separate highlights from object color. In *International Conference on Computer Vision (ICCV)*, pages 145–150, London, June 1987. IEEE.

[Klinker, 1988] Gudrun J. Klinker. *A Physical Approach to Color Image Understanding*. PhD thesis, Carnegie Mellon University, May 1988.

[Lee, 1988] H. C. Lee. Estimating the illuminant color from the shading of a smooth surface. A.I. Memo 1068, Massachusetts Institute of Technology Artificial Intelligence Laboratory, August 1988.

[Nayar *et al.*, 1991] S.K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):611–633, July 1991.

[Novak and Shafer, 1992] C. L. Novak and S. A. Shafer. Anatomy of a color histogram. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1992.

[Novak *et al.*, 1990] C. L. Novak, S. A. Shafer, and R. G. Willson. Obtaining accurate color images for machine vision research. *Proc. SPIE*, 1990.

[Shafer, 1984] S. A. Shafer. Optical phenomena in computer vision. In *Proceedings CSCSI-84*, London, Ontario, May 1984. Canadian Society for Computational Studies of Intelligence.

[Tominaga and Wandell, 1989] S. Tominaga and B. A. Wandell. Standard surface-reflectance model and illuminant estimation. *J. Opt. Soc. Am. A*, 6(4):576–584, April 1989.

[Torrance and Sparrow, 1967] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, September 1967.

[Wolff, 1992] L. B. Wolff. Diffuse reflection. In *Computer Vision and Pattern Recognition*, pages 472–478. IEEE, June 1992.