1

A Method for Estimating Scene Parameters from Color Histograms

Carol L. Novak   Steven A. Shafer

July 1993

CMU-CS-93-177

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

37P8

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.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

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. Physics-based models of reflection predict that 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.

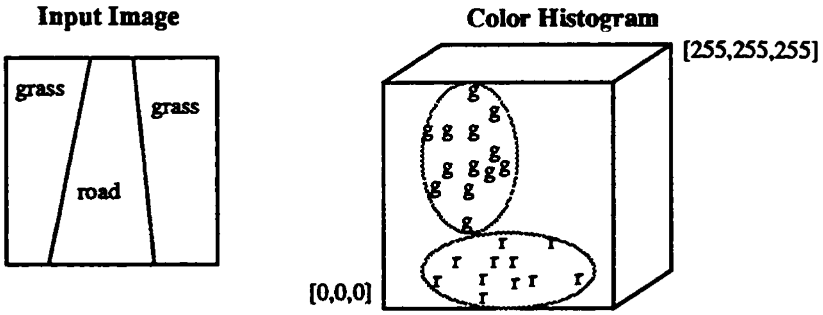
We present here an algorithm for analyzing color histograms that yields estimates of surface roughness, phase angle between the camera and light source, and illumination intensity. These three scene parameters are related to three histogram measurements. However the relationship is complex and cannot 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 present tests of our algorithm on simulated data and the results compare well with the known simulation parameters. We also test our method on real images and the results compare favorably with the actual parameters estimated by other means. 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 [4]. For example, pixels that correspond to grass were modeled as having a canonical color of green with some possible deviation from this color. The color variation was modeled as a probability distribution, so that the further from the characteristic color, the less likely it was that a pixel was grass. Figure 1 shows a diagram of this idea.



g and *r* are pixels in grass and road

Figure 1: Gaussian Histogram

In 1984 Shafer showed that for dielectric materials with highlights, the color histogram associated with a single object forms a plane [15]. 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 a diagram accompanying this idea (Figure 2), he visualized that the histogram would fill out a parallelogram. The paper did not describe how to determine these two vectors. In fact, for any planar histogram, there are many sets of two vectors that could define it.

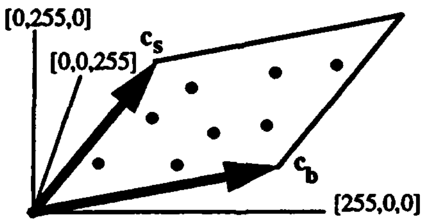


Figure 2: Planar Histogram

In 1987 Klinker and Gershon independently observed that the color histogram forms a T-shape or dog-leg in color space [9], [3]. Figure 3 shows that this color histogram 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 1988, Healey showed that the number of dimensions occupied by the histogram may be used to distinguish metals from dielectrics [5].

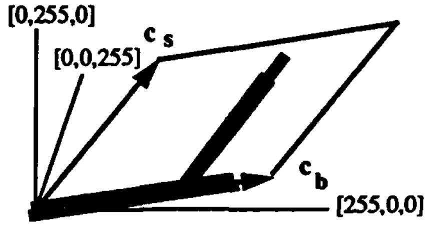


Figure 3: T-shaped Histogram

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 [14]. 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. Section 5 presents our results on real images.

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 [15]:

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

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 magnitude term m that depends only upon (θ_i), (θ_r) and (θ_p):

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

Figure 4 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).

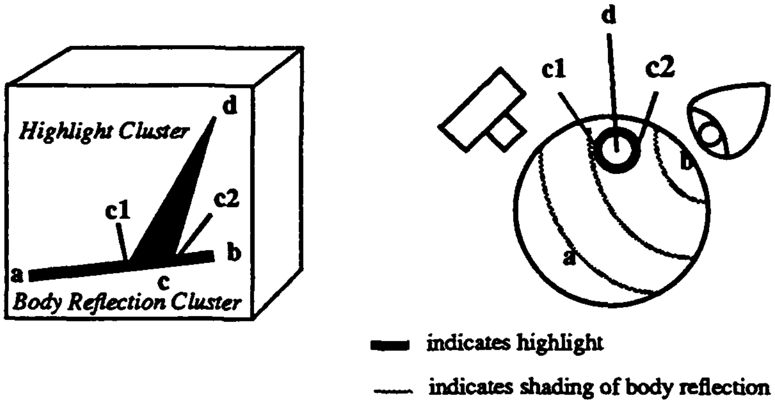


Figure 4: Histogram of an object

If the object has a completely black body color, there will be no body reflection component and hence no body reflection cluster. If the body reflection color is the same hue as the surface reflection color, the body reflection cluster and highlight cluster will be collinear. This is the case for white and gray objects since neither their body color nor surface color imparts any hue to reflected light. Therefore, objects of these colors cannot be analyzed by this type of method.

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 4), 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 $\operatorname{acos}(1/2)$ 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** in Figure 4) 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 [2], [6], [8], [16].

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 $|\overline{ab}|$) 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 [8].

In this presentation we use the Torrance-Sparrow model of scattering [17]. 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 in which we are interested.

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

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

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 α 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 [16] 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 [10]. 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. (It is the fact that highlights may be diffused that causes us to eschew the term "diffuse reflection" and use "body reflection" instead.) 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 object will have a large σ, and so will exhibit a shorter cluster. Figure 5 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** (see Figure 4).

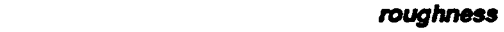


Figure 5: Dependence of highlight cluster length on object roughness

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. It will be assumed that the gain of the camera γ has been calibrated and that the facet size α is known. If not, then only the overall gain $\gamma\alpha B_b$ could be recovered from the histogram.

The graph in Figure 5 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. Figure 6 shows how the length of the highlight cluster changes as the camera and light source are separated by different angles with respect to the object. It demonstrates that the length changes

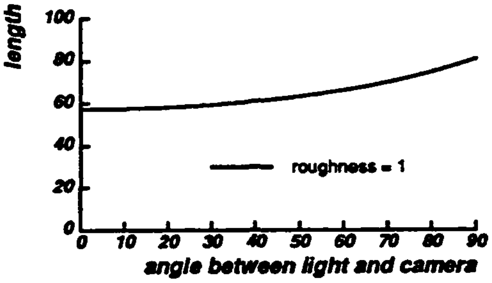


Figure 6: Dependence of cluster length on imaging geometry

slowly as the imaging geometry changes. Figure 7 shows how the length varies with roughness for a variety of viewing geometries. The effect is small but noticeable, so imaging geometry must be considered to make an accurate estimate of surface roughness.

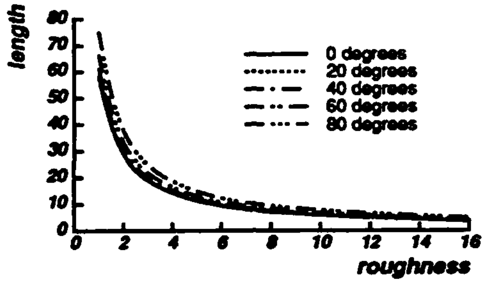


Figure 7: Dependence of length on roughness for different imaging geometries

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 c_1 to point c_2 in Figure 4). 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 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.

This property is independent of object size or shape. It simply shows the variation in surface normals over the area of the highlight. We do not have to fit a surface shape to the image to know how much scattering the object exhibits. It is encoded right there in the histogram.

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 $\overline{c_1 c_2}$

divided by the length of $\overline{ab}$ in Figure 4). A fraction of 0.5 would mean that the highlight cluster's base extended across half the length of the body reflection cluster. Figure 8 shows how the highlight cluster width varies with the surface roughness if a point light source is used.

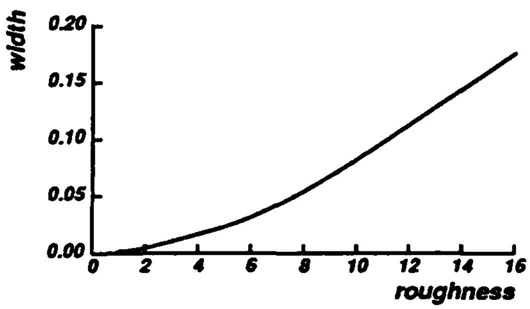


Figure 8: Dependence of highlight cluster width on object roughness

The brightness of the illumination B_s will also have some effect on 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 [12]. 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.

Figure 9 shows how the width of the highlight cluster varies with roughness for a variety of imaging geometries. The angle label is a measure of the angle (in degrees) that separates the light source and camera (with respect to the object). We assume that the object is small enough and far away enough from the camera that this angle is the same for all points on the surface.

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 4, we described the highlight cluster as beginning "some-

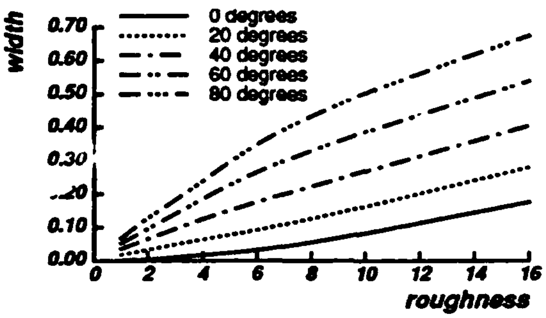


Figure 9: Dependence of highlight width on roughness for different imaging geometries

where" along the body reflection cluster. Klinker derived the "50% heuristic" which stated that for a large range of viewing geometries, the highlight cluster would start somewhere in the upper 50% of the body reflection cluster [8]. 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 4) 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. 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 $\mathrm{acos}\,(1/2)$ 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.

Figure 10 graphically illustrates this phenomenon. The histograms have been projected into the plane defined by the body reflection and surface reflection colors, so that the horizontal direction indicates increasing amounts of body reflection and the vertical direction indicates increasing amounts of surface reflection. This time, the amount of roughness has been held constant at $\sigma = 1$ while the angular separation of the light source and camera has been varied from 0 to 80 degrees. This graph shows how the meeting point decreases as the angle separating the camera and light source increases. Incidentally it also shows how the length and width of the cluster is affected by imaging geometry as described in sections 2.2.1 and 2.2.2.

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 11 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 $\overline{ac}/\overline{ab}$ (see Figure 4).

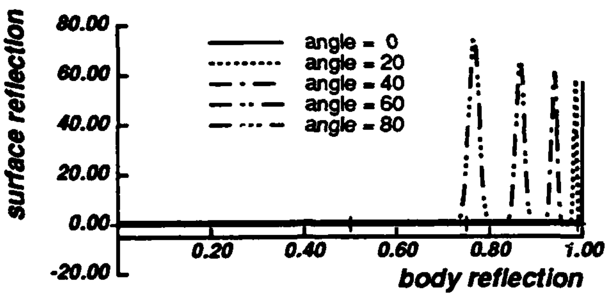


Figure 10: Illustration of how highlight cluster varies with changing imaging geometry

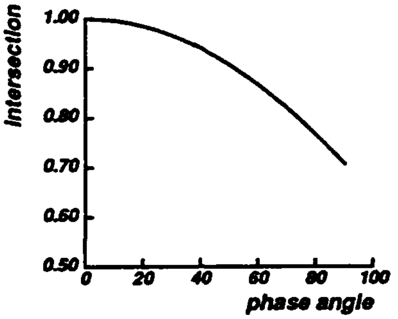


Figure 11: Dependence of intersection point on imaging geometry

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 [17]. The result is that the intersection is very slightly dependent upon the surface roughness [12].

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 in Figure 4). Klinker argued that this vector will usually correspond closely to the surface reflection color [8]. 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 12 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^{o}, 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.

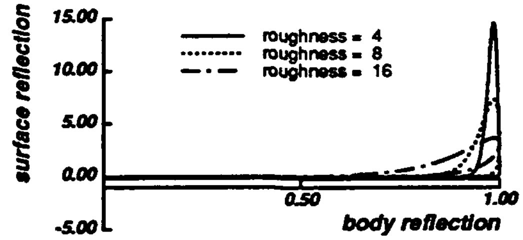


Figure 12: Dependence of highlight cluster direction on object roughness

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 between the two colors, which is not known in advance. If the vector fitted to the highlight 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.

Furthermore, skewing of the highlight cluster will also affect measurements of the cluster's length, width and intersection. Figure 13 shows an enlargement of the histogram for the roughest surface, with key points labeled. The estimate of illumination color obtained by fitting a vector to this highlight cluster will give the direction of the light color as $\overline{cd}$, whereas the correct illumination color is defined by the vertical direction $\overline{c'd}$ (since the histogram was projected into the plane defined by the body and surface reflection colors).

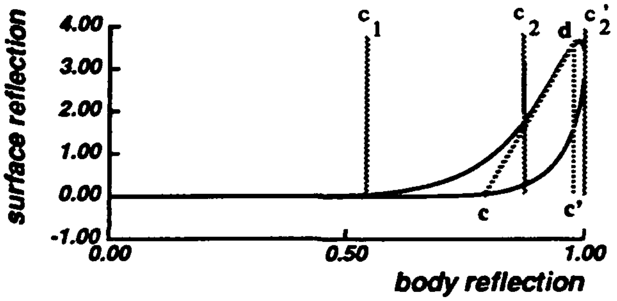


Figure 13: A skewed highlight

When the highlight cluster is skewed from the direction of the illumination color, the cluster will appear longer than it would otherwise be. The amount of surface reflection at the brightest point is the vertical displacement in the graph (the distance from point c$'$ to point d in Figure 13) rather than the distance along the highlight cluster (distance from point c to point d). Similarly, the width of the cluster is determined by projecting all pixels along the illumination color vector (straight down for this graph). Thus the correct measure of highlight width is the variation in body reflectance, which is the distance from c_1 to c_2' rather than the distance from c_1 to c_2. It follows then, that the correct intersection ratio is given by $\overline{ac'}|/|\overline{ab}|$ rather than $\overline{ac}|/|\overline{ab}|$.

The vector fitted to the highlight cluster (from point **c** to point **d** in Figure 4) 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 [20]. Also, some researchers prefer to use the Beckmann-Spizzichino model [1], or a combination of the Beckmann-Spizzichino and the Torrance-Sparrow models [11], 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 Table 1.

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

Table 1: Dependence of histogram features upon scene parameters

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 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. We also assume here that the camera is linear over the range of all possible pixel values that might occur in our images, so that we do not have to worry about "clipping" at some maximum value. (Since clipping is particularly common in pictures with highlights, this assumption will be removed in section 4.) 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 [7]. 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
- $\vec{c}_o$ - object chromaticity under "white" light

For convenience, we have separated the illumination color 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:

$$\vec{c}_o = [d_b^R/c_s^R, d_b^G/c_s^G, d_b^B/c_s^B]$$

The result is normalized to length 1.

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 (σ, θ_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

$$l = \mathrm{MAX}\left[\gamma B_s \frac{FG(\theta_i, \theta_r, \theta_p)\,\alpha}{\sigma \cos(\theta_r)} \exp\left(-\frac{\theta_s^2}{2\sigma^2}\right)\right] \tag{3}$$

over all values of θ_i, θ_r, and θ_s. If it were not for the presence of G in the numerator 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

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

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 14 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. (In this case the best fit line to the cluster would not pass through the brightest point in the highlight cluster.)

The *position* of the dotted line in Figure 14 shows the projection of the brightest highlight pixel onto the body reflection vector *along the best fit vector*, since the length is calculated from the brightest pixel. The "ideal" line 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.

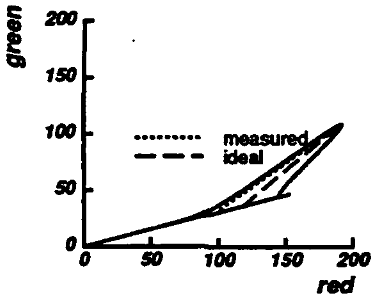


Figure 14: Skewed Length Measurement

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

Figure 15 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.

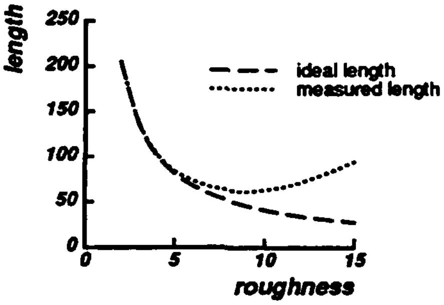


Figure 15: Ideal vs. Measured Length

A similar story applies to the measurement of width and intersection. 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 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?

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 16 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.

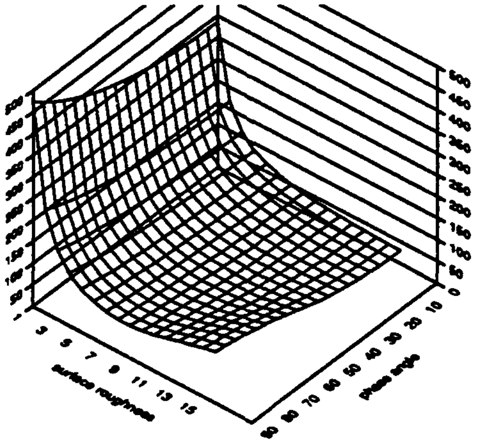


Figure 16: Variation in cluster length with changes in roughness and phase angle

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 graph in Figure 16 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 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. A program can then search through the lookup tables 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 17 shows the space of all possible values for length, width and intersection. The axis coming out of the page encodes the range of highlight cluster lengths; the vertical axis shows the range of widths; and the horizontal axis shows intersection measurements. The surface within this cube 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.

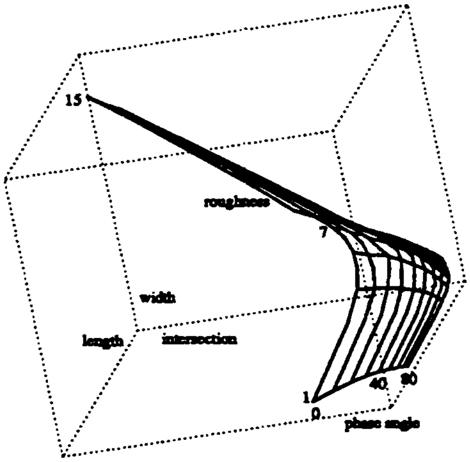


Figure 17: Surface of roughness and phase angles within L-W-I space

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.

3.5. Generating Lookup Tables

A large range of roughness values, phase angles, and illumination intensities is used to create the lookup tables. The ranges are shown in Table 2. 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

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

Table 2: Range of parameters

noise-free camera model. The histogram associated with the object is automatically separated into body reflection and highlight clusters [12]. 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

$$l = \mathrm{MAX}\ (m_s)\ \text{over all}\ \vec{p} \tag{4}$$

$$b = \mathrm{MAX}\ (m_b)\ \text{over all}\ \vec{p} \tag{5}$$

$$i = m_b/b\ \text{for that}\ \vec{p}\ \text{with maximum}\ m_s \tag{6}$$

$$w = (\mathrm{MAX}\ (m_b) - \mathrm{MIN}\ (m_b))/b\ \text{over all}\ \vec{p}\ \text{for which}\ m_s > T \tag{7}$$

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 17 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:

$$\begin{aligned} \sigma &\approx f_n(l, w, i) \\ &\approx A + Bl + Cw + Di + El^2 + Fw^2 + Gi^2 + Hlw + Ili + Jiw + \ldots \end{aligned} \tag{8}$$

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 *n*th 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:

$$\theta_p \approx g_n(l, w, i) \tag{9}$$
$$B_s \approx h_n(l, w, i) \tag{10}$$

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 Table 3.

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

Table 3: Fit of polynomial functions

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 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, 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 [8], 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 Table 2).

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.

Table 4 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 his-

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

Table 4: Results on noise-free data

togram. 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 (see Table 3).

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 Table 2, 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:

$$Skew \approx \Lambda_n(\sigma, \theta_p, B_s) \qquad (11)$$

The coefficients of the *n*th 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 Table 5. The error in estimating skew is the difference between the correct skew angle and the skew angle calculated by our method. The correct skew angle is 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.

Average error	1.73°
Average skew	8.63°
Minimum value	0.01°
Maximum value	27.5°
Number of skews > 1°	69
Cases considered	98/100

Table 5: Results in calculating skew

As an example, the performance of our algorithm on one of the test simulations will be described now. The test image, shown in Color Plate 1, 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 18. 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 unit vector fitted to the direction of 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.

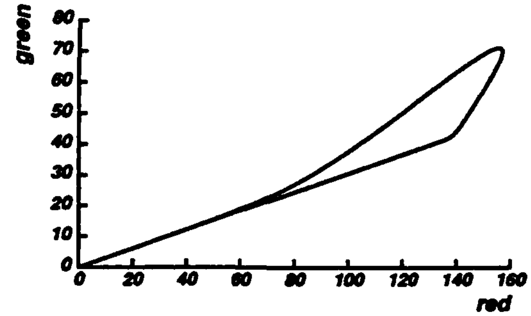


Figure 18: Histogram of simulated image

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 (8), (9) and (10) to the length, width and intersection measurements, the program estimated the roughness value as 11.99°, the phase angle as 67.05°, and the illumination intensity as 89% of maximum. We then applied equation (11) 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.

A picture made by simulating the cylinder with the recovered parameters is shown in Color Plate 2. The difference between this picture and the original is very subtle. The results for this example are summarized in Table 6. The full algorithm is diagrammed in Figure 19.

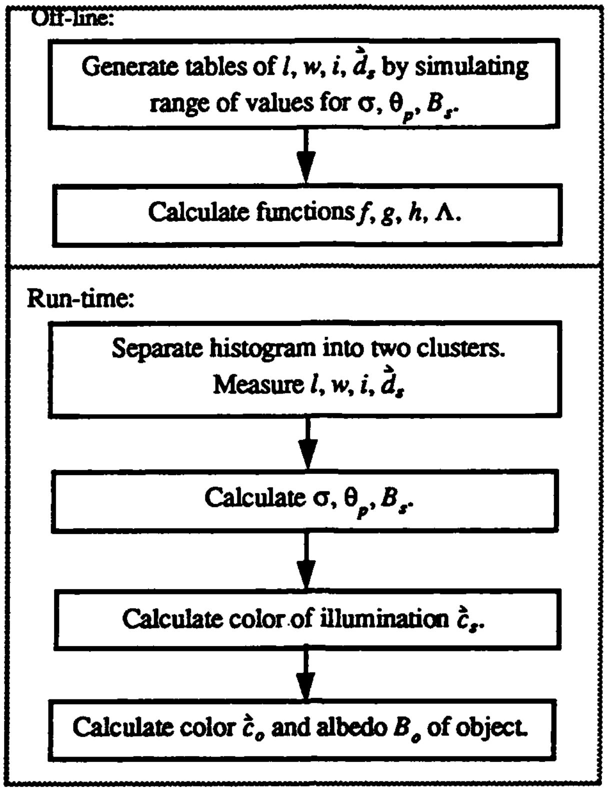

Off-line:
Generate tables of $l, w, i, \vec{d}_s$ by simulating range of values for σ, θ_p, B_s.
Calculate functions f, g, h, Λ.
Run-time:
Separate histogram into two clusters. Measure $l, w, i, \vec{d}_s$
Calculate σ, θ_p, B_s.
Calculate color of illumination $\vec{c}_s$.
Calculate color $\vec{c}_o$ and albedo B_o of object.

Figure 19: Algorithm for calculating scene parameters

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

Table 6: Example results

4. Extending the Analysis to 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 explain our 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. [7]. 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 must consider the effect of Gaussian noise on the histogram's appearance. Under this model, each pixel may be viewed as occupying a . robability 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 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 4) 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, when 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.

Table 7 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 19. Only the lookup tables have been changed, to reflect the noise model of the camera. Comparing the noise-free results in Table 4 and Table 5 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.

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

Table 7: Results on noisy data

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 hir tograms, 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.

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

$$\left.\begin{aligned} \sigma &= f_1(l, w, i) \\ \theta_p &= g_1(l, w, i) \\ B_s &= h_1(l, w, i) \end{aligned}\right\} \text{if histogram is not clipped} \tag{12}$$

$$\left.\begin{aligned} \sigma &= f_2(w_2, w, i) \\ \theta_p &= g_2(w_2, w, i) \\ B_s &= h_2(w_2, w, i) \end{aligned}\right\} \text{if histogram is clipped} \tag{13}$$

The results we have obtained with this technique are shown in Table 8. The results here are similar to those obtained without clipping in Table 7, although there are more problem cases that had to be eliminated from consideration. Techniques that will handle problem cases are a subject of future research.

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

Table 8: Results on data with noise and clipping

5. Application of the Algorithm to Real Images

Section 3 presented a method for analyzing color histograms assuming an ideal imaging system. Section 4 showed how that method may be modified in a systematic way for a more realistic camera system. This section will present some results from applying the algorithm developed in section 4 to real images.

The real images in this paper were taken in the Calibrated Imaging Laboratory (CIL) at Carnegie Mellon University. The CIL provides a controlled setting for taking high-quality images. It also allows for a careful analysis of camera issues that may cause problems for vision algorithms. The equipment and capabilities are described in [19].

5.1. Additional Problems with Real Cameras

Although section 4 modeled some fundamental limitations of real cameras, including noise and clipping, other problems that are commonly encountered in imaging were not modeled. These problems include nonlinearity, color imbalance, chromatic aberration, and blooming. Some problems such as linearity and color balance are easily fixed in software or with simple hardware. Others are more difficult to handle.

The degree of each of these problems is very specific to the particular camera system used. This makes them very difficult to model in a general way. Some systems may have been designed to reduce or even eliminate a specific problem, such as nonlinearity.

5.1.1. Nonlinearity and Color Imbalance

The methods we have developed for analyzing histograms require that the camera have a linear response. This is because they measure specific distances in the color space and assume that these distances are proportional to the amounts of body reflection and surface reflection observed on the object.

Fortunately, linear CCD cameras are commonly available. In fact CCD cameras are inherently linear since they produce a voltage that is proportional to the number of photons; the problem is that some models are designed to drive displays and have added circuitry that makes them nonlinear. In this case, the nonlinearity may be measured using a standard reference chart and corrected with a lookup table [8], [13]. The camera used to take the real images in this paper already has a linear response, so no correction is necessary.

Another common problem with CCD cameras is color imbalance. CCDs are typically much less sensitive to blue wavelengths than they are to green or red wavelengths. In addition, incandescent lights are usually employed when taking pictures in the CIL. Incandescent lights are also strong in the long wavelengths and weak in the short wavelengths. Therefore when incandescent lights are used with CCD cameras, the blue response is extremely poor. If color pictures are taken without some form of color correction, blue objects will be extremely dark, and white objects will appear yellowish.

A simple solution would be to scale the blue band of all color images by an appropriate constant. The problem with this approach is that while scaling up the blue signal, it would also scale up the noise. Therefore it would be better to increase the signal in some other way, such that the values in the blue band are as reliable as those in the red and green bands.

In the CIL, color pictures are taken with a gray-scale camera and a filter wheel to produce the red, green, and blue bands. This allows adjustments to be made individually for each band. In the case of a camera with a fixed exposure time, the aperture is changed between images of the bands, so that the aperture can be much wider to admit more light for the blue band. However, the aperture control method has the disadvantage that the depth of field changes slightly between bands, although this effect is not noticed in scenes with a small range of depths. A different technique is used for cameras that have a variable exposure time. A white test card is used to determine exposure times for each band. In the CIL, the exposure time for the blue band is typically about ten times longer than the red or green exposures. The images shown in this section were taken with this latter technique.

5.1.2. Chromatic Aberration

Another common problem in real images is chromatic aberration. This is an inherent property of lenses, because the index of refraction of optical components varies as a function of wavelength. Chromatic aberration can be partially compensated for during manufacturing by using pairs of lens elements with offsetting dispersion functions. Unfortunately this type of compensation is usually only calculated for two wavelengths (red and blue) and for a small number of points in the image field. Experience in the CIL has shown that even high quality lenses suffer from chromatic aberration that is readily apparent in histogram analysis.

Chromatic aberration is most pronounced where there are sharp edges in the image. For smoother surfaces, the transition between highlight and non-highlight areas will be more sudden, and thus the chromatic aberration will be more pronounced. The effect of chromatic aberration on the histogram is that pixels are displaced from their expected values, causing a scattering effect. The degree of this scattering depends not only upon the sharpness of the transition, but also upon the characteristics of the lens, the color of the object, and where in the image the highlight occurs. (Chromatic aberration is most pronounced in the periphery of the image.) This makes it very difficult to model in a general way.

In our work we take a two-pronged attack upon the problem of chromatic aberration. The first step is to eliminate as much of the chromatic aberration as possible at the time the image is taken. This is done by active lens compensation. Since the CIL allows fine control of the lens position, and since the color bands are imaged separately, the lens is refocused, re-zoomed, and repositioned between the red, green and blue color bands. This approach typically reduces the chromatic aberration by an order of magnitude [13], [18].

Unfortunately, the method does not completely eliminate all chromatic aberration in images with highlights. Also it is time-consuming to calibrate and is not generally applicable to camera setups that do not have a finely controlled lens. Therefore the second approach is to take some chromatic aberration into account when measuring the dimensions of the color histogram.

Figure 20 shows a histogram for a very shiny red pail. The original image is shown in the upper right hand corner of the image. The top of the highlight cluster is fanned out, making it wider than the bottom of the highlight cluster. (The bottom few rows of the original image where blooming has occurred were manually eliminated from consideration.)

This fanning of the highlight cluster is particularly troublesome because it runs completely counter to the model of histogram behavior. The model says that there will be more variation of body reflection amounts at the base of the highlight cluster than at the upper end. A highlight cluster that is wider at the top implies that there are very bright highlight pixels occurring at several different reflection angles, but that for some

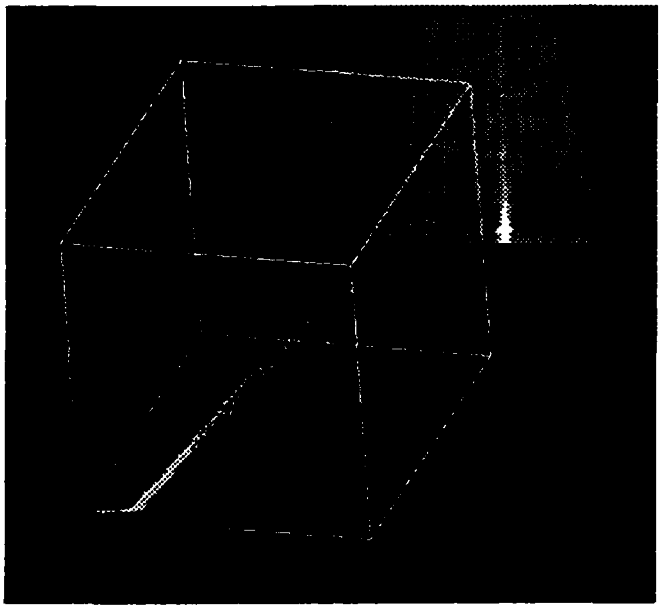


Figure 20: Histogram for a very shiny pail

reason pixels with lesser amounts of highlight occur at only a few reflection angles. If the histogram were to be believed, then the amount of highlight does not fall off as the off-specular angle is increased. This would mean a rather unusual distribution of facet angles. A more plausible explanation of the fan-out of the highlight cluster is that the chromatic aberration causes large hue shifts at points where the highlight is increasing very rapidly.

Since the chromatic aberration makes the highlight cluster wider in regions where the highlight increases rapidly, this would imply that the width measurement is not very trustworthy at the top end of the highlight cluster. A program analyzing the histogram in Figure 20 will measure a very wide highlight cluster, even though it corresponds to a very shiny object. This is because the original algorithm examines the amount of body reflection present in *all* highlight pixels to compute the width measurement. A pixel is classified as a highlight is the amount of surface reflection exceeds the threshold $4\sigma_c$. The surface reflection model predicts that the widest part of the highlight cluster is at the base, but this is not explicitly tested by the program described in section 4.

Although the highlight cluster is plagued by chromatic aberration at the top end, the base of the cluster is still well behaved. Now that chromatic aberration is known to be a problem, the algorithm measures width over all highlight pixels *near* the threshold point, instead of over all pixels that *exceed* the threshold. Equation (7), which gives the width measurement in terms of the computed amounts of body reflection m_b and surface reflection m_s, is modified to

$$w = [\text{MAX}(m_b) - \text{MIN}(m_b)]/b \text{ over all } \vec{p} \text{ for which } T < m_s < T + \sigma_c$$

Similarly, the second width measurement, used when the highlight cluster is clipped, becomes

$$w_2 = [\text{MAX}(m_b) - \text{MIN}(m_b)]/b \text{ over all } \vec{p} \text{ for which } 3T < m_s < 3T + \sigma_c$$

The process for measuring length and intersection remain unchanged.

5.1.3. Blooming

The lower section of the pail image in Figure 20 shows that the brightest section of the highlight has "bloomed" into neighboring pixels. The blooming is most pronounced in the red band of the image, creating a red halo around the highlight. This effect is not chromatic aberration since it is several pixels wide.

Blooming occurs when points on the CCD chip are so saturated by incoming photons that the charge in one well on the chip spills over into neighboring charge wells. Some cameras are designed to be fairly resistant to blooming, an advantage when taking pictures of highlights. The camera used to take the pictures in this paper is very prone to blooming. An interesting thing to note is that it blooms predominantly in the horizontal direction. The picture in Figure 20 is only a mild case of blooming. In some cases the highlight will smear all the way across the image.

Bloomed pixels that saturate the camera do not present a problem for the method developed here. Saturated pixels are easy to detect; any pixel with a value equal to the camera maximum is considered untrustworthy and disregarded. The fact that the object is too bright will be noted and the clipped method is used to calculate the scene parameters. However sometimes blooming will increase the values of nearby pixels *without* causing them to saturate. This means the pixel values may be increased by some arbitrary amount, to some new arbitrary value that is less than the camera maximum. Since the blooming may effect very large areas of the image, it is difficult to handle in an automatic way. In the histogram for Figure 20, *all* pixels that occurred in the same rows as bloomed pixels were eliminated from consideration.

Blooming can be reduced a great deal by manufacturing a camera chip with charge wells further apart. A camera of this type would make is possible to consider all regions of the image when analyzing the histogram, instead of arbitrarily throwing away portions adjacent to saturated pixels. Saturated pixels do not present a problem for our method, *provided* they do not spill charge into neighboring pixels.

This section has described some modifications made to the analysis method along with some techniques for taking high quality images. Taken together, they allow the algorithm to work on real images. The next few sections describe the performance of the algorithm in estimating phase angle, illumination intensity, and roughness from real images.

5.2. Estimating Phase Angle

An experiment was set up in the CIL to test the histogram analysis algorithm at estimating phase angle from real images. A series of images was taken with the camera and light source separated by different phase angles. The angle was measured with a large protractor and strings to indicate the direction of the camera and light source. A diagram of the setup is shown in Figure 21. The angles measured by this method were estimated to be accurate to within 5 degrees. The light was a 250 Watt spotlight; it was estimated to appear a few degrees across at the distance used in the experiment (about 5 feet). Therefore it is only a crude approximation of a point source.

The first image in the sequence is shown in Color Plate 3. This picture was taken when the camera and light source were approximately 10 degrees apart. The phase angle was then increased by 10 degrees between each picture. The last image in the sequence is shown in Color Plate 4. This picture was taken when the phase angle between the camera and light source was 90 degrees.

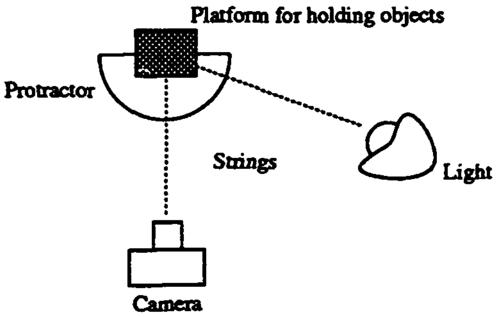


Figure 21: Experiment for estimating phase angle

The program automatically split the color histograms of the objects into two clusters, fit lines to those vectors, and calculated the values of length, width, and intersection. The color histogram of the image in Plate 4 is shown in Color Plate 5. The superimposed white lines show the lines fit by the program.

This process was repeated for each image in the sequence. Figure 22 shows the measured intersection value for each image, plotted against the phase angle that was in effect at the time the image was taken. A comparison of Figure 22 with Figure 11 in section 2 shows that, as predicted, the intersection decreases with increasing phase angle. The sole exception is at $\theta_p = 90°$. A slight error in setting up the light could explain this problem: if the phase angle was actually slightly larger than 90°, the brightest point on the object would be out of view of the camera, throwing off the intersection ratio.

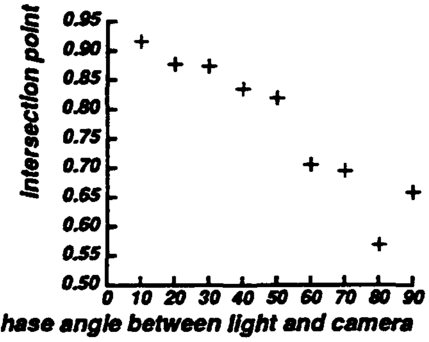


Figure 22: Plot of intersection vs. phase angle

The same polynomial coefficients used in section 4.2 were used to calculate the phase angle from the length, width and intersection measurements. The lookup tables in that section assumed a camera noise $\sigma_c = 1.275$ and clipping at a maximum value of 255. The results are shown in Figure 23. The dotted line shows the correct answer, using the phase angle measured by the protractor as ground truth. The average error in estimating angle is 9.96°.

As the light was moved around, the distance from the object did not remain constant due to the layout of the lab. Therefore the intensity of the illumination was not assumed to be constant. For two of the images in the sequence, the illumination was close enough that the highlight saturated the camera and clipping occurred. This happened at $\theta_p = 60°$ and $\theta_p = 80°$. In these two cases the second set of equations (see equation (13)) for clipping were used. Although this technique produced a reasonable answer for the case when $\theta_p = 80°$, it did not do so well for $\theta_p = 60°$. If this data point is eliminated from consideration, the average error in estimating the phase angle is 7.04°.

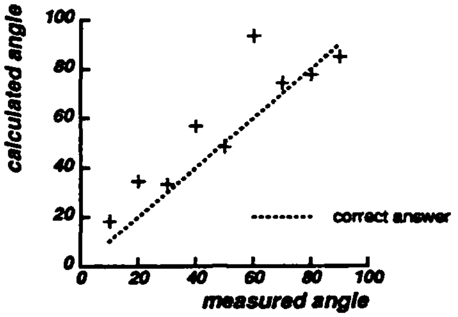


Figure 23: Results for calculating phase angle from real images

Overall the method developed for estimating phase angle from analyzing color histograms works fairly well, especially considering that the ground truth measurement of the phase angle is fairly crude. Also the lookup tables in section 4.2 were calculated without calibrating the simulated images to the conditions in the CIL. In particular, the noise of the camera was not measured precisely and the light source used in the experiments was not a point source as was used in the simulations. Moreover, the Torrance-Sparrow roughness model is an idealized model that assumes isotropic, Gaussian scattering; it may not describe very well the roughness of the real object used in these experiments.

5.3. Estimating Illumination Intensity

A second experiment was performed in the CIL to test the performance of the algorithm at estimating illumination intensity. The spotlight was plugged into a variable voltage supply with a manually operated dial. A sequence of images was taken under increasing levels of illumination, while the imaging geometry and target object were kept constant. Altogether six images were taken. The illumination level was measured with a luminance spot meter aimed at a white card. The spot measurements were estimated to be repeatable to within 5%.

Again, the program analyzed the histograms to produce measurements of length, width and intersection for each image. The polynomial equation to calculate illumination intensity was then applied to these measurements. The results are shown in Figure 24. The horizontal axis shows the luminance (candles per square meter) measured by the spot meter, while the vertical axis shows the intensity estimated by the histogram analysis. The gain of the camera has not been calibrated, so the program gives a relative estimate of intensity. The dotted line shows the best linear fit to the data. If the slope of that line is considered to be the gain of the camera, then the average error in estimating illumination intensity is 5.07%.

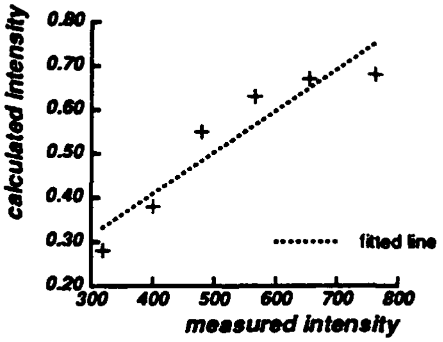


Figure 24: Results for calculating illumination intensity from real images

The results in Figure 24 were computed using only the length, width, and intersection measurements of the highlight cluster. The graph in Figure 25 shows the roughness estimate for each image in the sequence, computed at the same time as the intensity estimates. These measurements are fairly stable across the sequence as would be expected since the imaging geometry and target object were the same in every case. This result is particularly important, since both increasing smoothness and increasing illumination intensity will lengthen the highlight cluster. We also found that the estimates of phase angle made at the same time to be reasonably consistent. This shows that for the most part, the algorithm *does not* confuse changes in the histogram shape due to increasing intensity levels with the effects of roughness and phase angle.

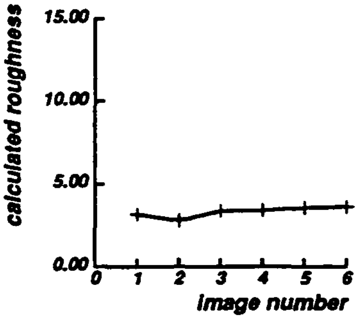


Figure 25: Calculating roughness when intensity changes

The ground truth measurements for this experiment came from a *luminance* meter, which were compared with estimates of *intensity*. Unfortunately these measurements are somewhat different. Luminance is measured by integrating with the human luminous efficiency function $V(\lambda)$. A luminance meter shows how bright a spot will appear to humans, but not necessarily how bright it will appear to a color camera. A narrow band red illumination will have a much lower luminance than a narrow band green one, yet both may have the same intensity as measured by a color camera.

The difference between luminance and intensity is relevant to the results presented here, because as the voltage fed to the spotlight is decreased, the light becomes noticeably redder in hue along with becoming dimmer. This means that luminance may not correlate well with the intensity in this experiment. While the histogram analysis did correctly calculate increasing illumination intensities as the light level was increased, the calculated values do not appear as linear as one might hope. The use of a luminance meter might be causing some of the problems.

5.4. Estimating Roughness

A third experiment was performed to show how the system estimates surface roughness from color histograms. Color Plate 6 show a composite of fives images of different objects. The objects are: a green plastic pool toy in the shape of an alligator; an orange plastic pumpkin for trick-or-treating; a terra-cotta ball (also seen in Plate 3); a red plastic ball; and a red plastic pail (also seen in Figure 20).

Table 9 shows the roughness calculated by the system for each of these objects. The objects are listed in order of decreasing roughness, as estimated by human observation. The calculated roughness number is the standard deviation of facet angles, in degrees.

There is no error measure for these results, since there is no ground truth data for the actual roughness val-

Object	Calculated Roughness
Alligator	10.07°
Pumpkin	8.93°
Terra-cotta ball	3.61°
Red ball	0.40°
Red pail	0.10°

Table 9: Results for estimating roughness

ues. Nevertheless, the roughness ranking from the program agrees with that produced by a human observer. The pumpkin presents a particularly interesting case, since it shows roughness at more than one scale. At the large scale where roughness is judged by touch, it is clearly the roughest object. This roughness is large enough to be considered "texture". At the smaller scale of optical roughness, it is considered to be more shiny than the alligator.

The results in this section demonstrate that the method developed here can be applied to real images. The algorithm produces reasonable estimate of phase angle, illumination intensity, and surface roughness by analyzing the dimensions of color histograms. The method is able to succeed even with such camera problems as noise, clipping, and chromatic aberration. It is likely that the method would perform even better if the lookup tables used measurements from real histograms rather than from simulated ones. Nevertheless, the method works reasonably well, even without prior reference to real data.

6. Conclusions

The color histogram of an image is a rich source of information, but it has not been fully exploited in the past. We have shown that the color histogram of a dielectric object may by characterized by a small number of measurements, which relate directly to many scene properties. We have shown how these histogram measurements may be used to recover estimates of surface roughness, imaging geometry, illumination intensity, and illumination color. These estimates may in turn be used to calculate object color and albedo.

The resulting algorithm is applied to real images, and produces good estimates of phase angle, illumination intensity, and surface roughness. The method is independent of the shape of the object, and works on shapes ranging from a pumpkin to an alligator. The model used to develop the lookup tables is fairly general, and was not calibrated to match the actual imaging conditions such as light source extent, camera noise characteristics, etc. This kind of analysis may be applied to such varied tasks as surface inspection and object recognition.

7. Acknowledgments

The authors would like to thank Reg Willson, Jim Moody and Bill Ross who developed much of the hardware and software that makes picture-taking possible in the Calibrated Imaging Laboratory.

This research was sponsored by the Avionics Lab, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, Arpa Order No. 7597; and by the Air Force Office of Scientific Research under Contract F49620-86-C-0127.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

Bibliography

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

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

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

[4] Haralick, R. M. and Kelly, G. L. "Pattern recognition with measurement space and spatial clustering for multiple images." *Proceedings IEEE*, 57, 1969, 654-665.

[5] Healey, G. "A color reflectance model and its use for segmentation." *International Conference on Computer Vision*, IEEE, 1988, 460-466.

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

[7] Healey, G. and Kondepudy, R. "Modeling and calibrating ccd cameras for illumination insensitive machine vision." In *Proceedings of the SPIE*, No. 1614 SPIE 1991.

[8] Klinker, G. J. *A Physical Approach to Color Image Understanding*, Ph.D. thesis, Carnegie Mellon University, 1988.

[9] Klinker, G. J. and Shafer, S. A. and Kanade, T. "Using a color reflection model to separate highlights from object color." *International Conference on Computer Vision*, IEEE, 1987, 145-150.

[10] Lee, H. C. "Estimating the illuminant color from the shading of a smooth surface."AI Memo 1068 MIT AI Lab. 1988.

[11] Nayar, S. K., Ikeuchi, K. and Kanade, T. "Surface reflection: Physical and geometrical perspectives." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7) 1991.

[12] Novak, C. L. *Estimating Scene Properties by Analyzing Color Histograms with Physics-Based Models*, Ph.D. thesis, Carnegie Mellon University, 1992.

[13] Novak, C. L., Shafer, S. A. and Willson R. G. "Obtaining accurate color images for machine vision research." In *Proceedings of the SPIE.* SPIE, 1990.

[14] Novak, C. L. and Shafer, S. A. "Anatomy of a Color Histogram." In *Computer Vision and Pattern Recognition.* IEEE, 1992.

[15] Shafer, S. A. "Using color to separate reflection components." Computer science department, University of Rochester, 1984.

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

[17] Torrance, K. and Sparrow, E. "Theory for off-specular reflection from roughened surfaces." *J. Opt. Soc. Am.* (57), 1967.

[18] Willson R. G. and Shafer, S. A. "Active lens control for high precision computer imaging." In *Inter-*

national Conference on Robotics and Automation. IEEE, 1991.

[19] Willson R. G. and Shafer, S. A. "Precision imaging and control for machine vision research at Carnegie Mellon University." In *SPIE Conference on High Resolution Sensors and Hybrid Systems*. SPIE, 1992.

[20] Wolff, L. B. "Diffuse reflection." In *Computer Vision and Pattern Recognition*. IF⁼E, 1992.

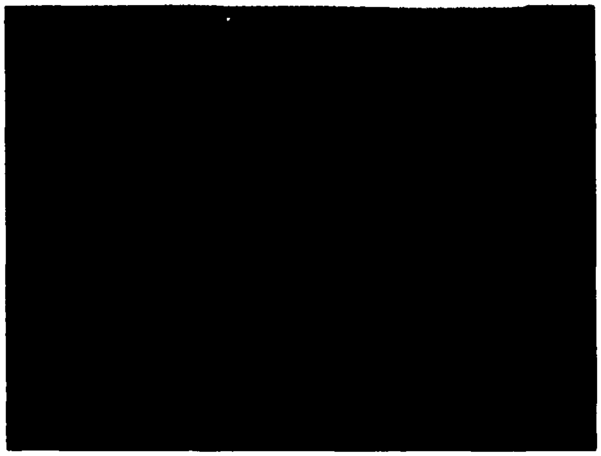


Plate 1: Simulated test image

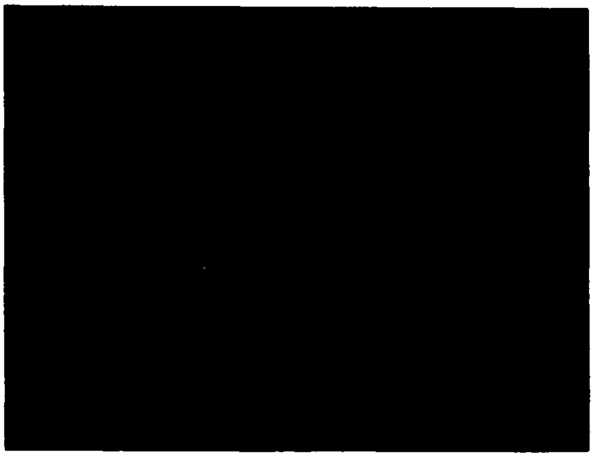


Plate 4: Image with phase angle of 90 degrees

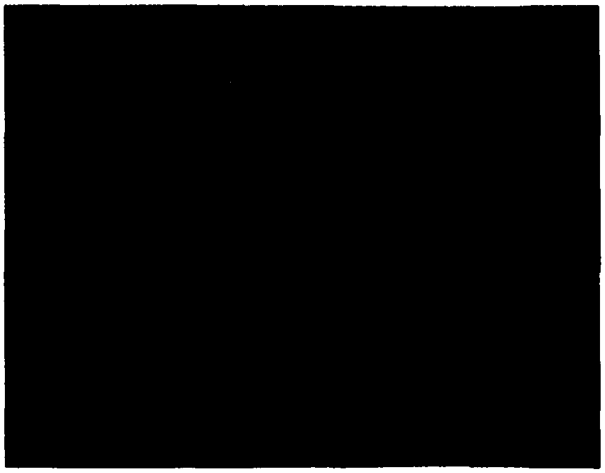


Plate 2: Simulation from recovered parameters

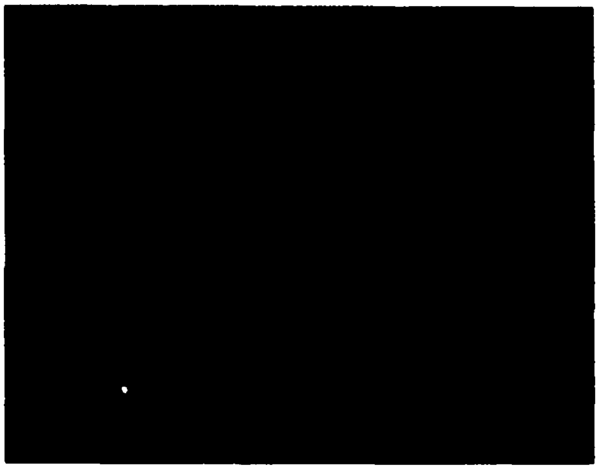


Plate 5: Histogram for phase angle of 90 degrees

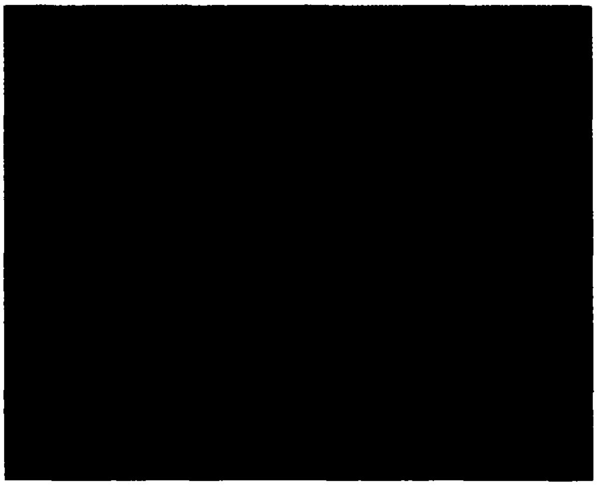


Plate 3: Image with phase angle of 10 degrees

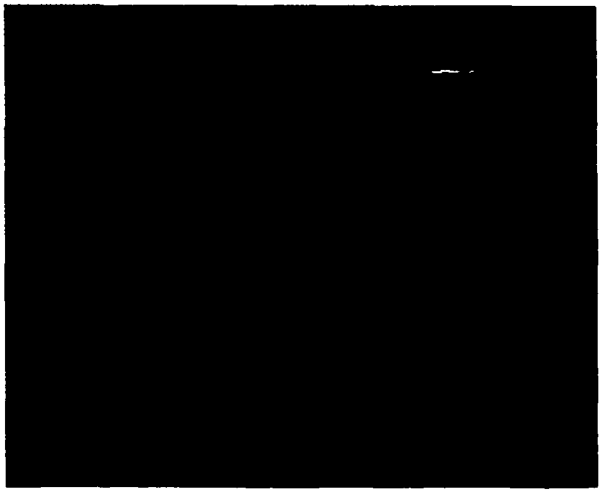


Plate 6: Five objects with different roughnesses