

# Separating Reflection Components of Textured Surfaces using a Single Image

Robby T. Tan

Katsushi Ikeuchi

Department of Computer Science  
The University of Tokyo  
{robby,ki}@cvl.iis.u-tokyo.ac.jp

## Abstract

*The presence of highlights, which in dielectric inhomogeneous objects are linear combination of specular and diffuse reflection components, is inevitable. A number of methods have been developed to separate these reflection components. To our knowledge, all methods that use a single input image require explicit color segmentation to deal with multicolored surfaces. Unfortunately, for complex textured images, current color segmentation algorithms are still problematic to segment correctly. Consequently, a method without explicit color segmentation becomes indispensable, and this paper presents such a method. The method is based solely on colors, particularly chromaticity, without requiring any geometrical parameter information. One of the basic ideas is to compare the intensity logarithmic differentiation of specular-free images and input images iteratively. The specular-free image is a pseudo-code of diffuse components that can be generated by shifting a pixel's intensity and chromaticity nonlinearly while retaining its hue. All processes in the method are done locally, involving a maximum of only two pixels. The experimental results on natural images show that the proposed method is accurate and robust under known scene illumination chromaticity. Unlike the existing methods that use a single image, our method is effective for textured objects with complex multicolored scenes.*

## 1 Introduction

Separating diffuse and specular reflection components is an essential subject in the field of computer vision. Many algorithms in this field assume perfect diffuse surfaces and deem specular reflections to be outliers. However, in the real world, the presence of specular reflection is inevitable, since there are many dielectric inhomogeneous objects which have both diffuse and specular reflections. To properly acquire the diffuse only reflection, a method to separate the two components robustly and accurately is required. Once this separation is done, the specular reflection component becomes advantageous, since it conveys useful information of the surface properties such as microscopic roughness.

Many works have been developed for separating reflec-

tion components. Wolff et al. [19] used a polarizing filter to separate reflection components from gray images. Nayar et al. [12] extended this work by considering colors instead of using the polarizing filters alone. They identified the illumination color vector in RGB space indicated by specular intensity variation produced by the polarizing filter. The combination of polarizing filter and colors is feasible even for textured surfaces; however, utilizing such an additional filter is impractical in some circumstances. Sato et al. [15] introduced a four-dimensional space, temporal-color space, to analyze the diffuse and specular reflections based solely on colors. While this method requires dense input images with variation of illuminant direction, it has the ability to separate the reflection components locally, since each location contains information of diffuse and specular reflections. Recently, instead of using dense images, Lin et al. [11] used sparse images under different illumination positions to resolve the separation problem. They proposed an analytical method that combines the finite dimensional basis functions [14] and dichromatic model to form a closed form equation, by assuming that the sensor sensitivity is narrowband. This method is also able to separate the reflection component locally. Other different methods using multiple images can also be found in the literature [13, 9, 10].

Shafer [16], who introduced the dichromatic reflection model, was one of the early researchers who used a single colored image. He proposed a separation method based on parallelogram distribution of colors in RGB space. Klinker et al. [7] then extended this method by introducing a T-shaped color distribution. This color distribution represents reflectance and illumination color vectors. By separating these color vectors, the reflection equation becomes a closed form equation and directly solvable. Unfortunately, for many real images, this T shape is hardly extractable due to noise, etc. Bajscy et al. [1] proposed a different approach by introducing a three dimensional space composed of lightness, saturation and hue. In their method, the input image has to be neutralized to pure-white illumination using a linear basis functions operation. For every neutralized pixel, the weighting factors of the surface reflectance basis functions are projected into the three-dimensional space, where specular and diffuse reflections can be identifiable, due to the differences of their saturation values.

All above methods that use a single input image require

color segmentation to deal with multicolored images. For non-complex multicolored images, current segmentation algorithms can give reasonably correct results. However, in the real world with complex scene and textured surfaces, these algorithms are still problematic. To overcome this problem, we offer a method that does not require explicit color segmentation; hence, it is applicable for textured surfaces in complex multicolored scenes. Briefly, our method is as follows: given a single colored image, we normalize the illumination color using known illumination chromaticity. The normalization produces an image that has pure-white specular component. Using this image, a pseudo-code of the diffuse component we call *specular-free image* can be generated by simply shifting the intensity and chromaticity of the pixels non-linearly while retaining their hue. The specular-free image has diffuse geometry exactly identical to the diffuse geometry of the input image; the difference is only in the surface color. Thus, by using intensity logarithmic differentiation on both normalized image and its specular-free image, we can determine whether the normalized image contains only diffuse pixels. This ability plays an important role as a termination condition in our iterative framework, which removes specular components step by step until no specular component exists in the image. All processes are done locally, involving only a maximum of two neighboring pixels.

Our method offers several advantages: first, the separation is done without requiring explicit segmentation; second, the method uses simple and hands-on illumination color normalization; third, the specular-free image that has identical geometrical parameters to diffuse components is probably useful for many algorithms in computer vision. In order to separate reflection components correctly, our method requires several assumptions: first, diffuse pixels must occur in one color area, regardless of their quantity; second, illumination chromaticity is known; third, all pixels of the input image must be chromatic pixels ( $R \neq G \neq B$ ).

The rest of the paper is organized as follows. In Section 2, we discuss the dichromatic model, image color formation and normalization. In Section 3, we elaborate the method in detail, describing the derivation of the theory for separating reflection components. We provide a description of the implementation of the method and experimental results for real images in Section 4. Finally, we offer our conclusions in Section 5.

## 2 Reflection Models

Reflection on most inhomogeneous materials is usually described by the dichromatic reflection model, which states that the light reflected from an object is a linear combination of diffuse and specular reflections:

$$\bar{I}(\lambda, \bar{x}) = w_d(\bar{x})S_d(\lambda, \bar{x})E(\lambda, \bar{x}) + w_s(\bar{x})S_s(\lambda, \bar{x})E(\lambda, \bar{x}) \quad (1)$$

where  $\bar{x} = \{r, s, t\}$  is the position of a surface point in a three-dimensional world coordinate system;  $w_d(\bar{x})$  and  $w_s(\bar{x})$  are the weighting factors for diffuse and specular re-

flection, respectively; their values depend on the geometric structure at location  $\bar{x}$ .  $S_d(\lambda, \bar{x})$  is the diffuse spectral reflectance function;  $S_s(\lambda, \bar{x})$  is the specular spectral reflectance function;  $E(\lambda, \bar{x})$  is the spectral energy distribution function of the illumination.

The spectral energy distribution of the specular reflection component is similar to the spectral energy distribution of the incident light [8]. Researchers usually assume that both of them are identical [3, 18], which is named the neutral interface reflection (NIR) assumption by Lee et al. [8]. As a result, we can set  $S_s(\lambda, \bar{x})$  as a constant, and Equation (1) becomes:

$$\bar{I}(\lambda, \bar{x}) = w_d(\bar{x})S_d(\lambda, \bar{x})E(\lambda, \bar{x}) + \tilde{w}_s(\bar{x})E(\lambda, \bar{x}) \quad (2)$$

where  $\tilde{w}_s(\bar{x}) = w_s(\bar{x})k_s(\bar{x})$ , with  $k_s(\bar{x})$  is a constant scalar w.r.t. the wavelength.

**Image Formation.** By ignoring camera noise and gain, an image taken by a digital color camera can be described as:

$$\bar{I}_c(\mathbf{x}) = w_d(\mathbf{x}) \int_{\Omega} S_d(\lambda, \mathbf{x}) E(\lambda) q_c(\lambda) d\lambda + \tilde{w}_s(\mathbf{x}) \int_{\Omega} E(\lambda) q_c(\lambda) d\lambda \quad (3)$$

where  $\mathbf{x} = \{x, y\}$ , the two dimensional image coordinates;  $q_c$  is the three-element-vector of sensor sensitivity and index  $c$  represents the type of sensors (R, G, and B). In this paper, we assume a single uniform illumination color, so that the illumination spectral distribution  $E(\lambda)$  becomes independent from the image coordinate ( $\mathbf{x}$ ). The integration is done over the visible spectrum ( $\Omega$ ).

For the sake of simplicity, Equation (3) is written as:

$$\bar{I}_c(\mathbf{x}) = \bar{m}_d(\mathbf{x})\bar{\Lambda}_c(\mathbf{x}) + \bar{m}_s(\mathbf{x})\bar{\Gamma}_c \quad (4)$$

where  $\bar{m}_d(\mathbf{x}) = w_d(\mathbf{x})L(\mathbf{x})k_d(\mathbf{x})$ , with  $L(\mathbf{x})$  is the spectral magnitude of the surface irradiance on a plane perpendicular to the light source direction;  $k_d(\mathbf{x})$  is the scene radiance to surface irradiance ratio of diffuse surface;  $\bar{m}_s(\mathbf{x}) = \tilde{w}_s(\mathbf{x})L(\mathbf{x})$ ;  $\bar{\Lambda}_c(\mathbf{x}) = \int_{\Omega} s_d(\lambda, \mathbf{x})e(\lambda)q_c(\lambda)d\lambda$ ; with  $s_d(\lambda, \mathbf{x})$  is the normalized surface reflectance spectral function,  $e(\lambda)$  is the normalized illumination spectral energy distribution.  $\bar{\Gamma}_c = \int_{\Omega} e(\lambda)q_c(\lambda)d\lambda$ . The first part of the right side of the equation represents the diffuse reflection component, while the second part represents the specular reflection component.

**Normalization.** In our method, to separate reflection components correctly, the color of the specular component must be pure white ( $\bar{\Gamma}_r = \bar{\Gamma}_g = \bar{\Gamma}_b$ ). However, in the real world, finding a pure white specular component is almost impossible. Most light sources are not wavelength-independent. Moreover, even if the light source is wavelength-independent, because of different sensitivities in color filters, the intensity value of the specular component for every color channel becomes varied, depending

on camera sensitivity. Thus, to obtain a pure white specular component we need to normalize the input image. Here we propose a simple method of normalization without requiring approximated linear basis functions such as in [1].

Mathematically, illumination chromaticity can be written as:  $\psi_c = n \int_{\Omega} e(\lambda) q_c(\lambda) d\lambda$ , where the value of  $n$  is a positive real number ( $0 < n \leq 1$ ). In fact, the exact value of  $n$  is unknown, as the magnitude of incident light is unrecoverable using either chromaticity of a white reference captured by a CCD camera, or current illumination chromaticity algorithms. Fortunately, this magnitude is unnecessary, since we require only the ratio of illumination color. Then, the normalized image is computed using:

$$I_c(\mathbf{x}) = m_d(\mathbf{x})\Lambda_c(\mathbf{x}) + m_s(\mathbf{x}) \quad (5)$$

where  $I_c(\mathbf{x}) = \frac{\bar{I}_c(\mathbf{x})}{\psi_c}$ ;  $\Lambda_c(\mathbf{x}) = \frac{\bar{\Lambda}_c(\mathbf{x})}{\int_{\Omega} e(\lambda) q_c(\lambda) d\lambda}$ ;  $m_d(\mathbf{x}) = \frac{\bar{m}_d(\mathbf{x})}{n}$ ;  $m_s(\mathbf{x}) = \frac{\bar{m}_s(\mathbf{x})}{n}$ . The equation shows that the specular reflection component becomes pure-white color.

Later, when the separation is done for the normalized image, to obtain the actual reflection components, we need to renormalize the separated components, simply by multiplying them with  $\psi_c$ .

### 3 Separation Method

#### 3.1 Specular-to-diffuse mechanism

*Specular-to-diffuse mechanism* bases its techniques on chromaticity and intensity value of specular and diffuse pixels. Chromaticity is usually defined as  $c(\mathbf{x}) = \frac{I_c(\mathbf{x})}{\Sigma I_i(\mathbf{x})}$ . Accordingly, we define *maximum chromaticity* as:

$$\tilde{c}(\mathbf{x}) = \frac{\max(I_r(\mathbf{x}), I_g(\mathbf{x}), I_b(\mathbf{x}))}{\Sigma I_i(\mathbf{x})} \quad (6)$$

where  $\Sigma I_i(\mathbf{x}) = I_r(\mathbf{x}) + I_g(\mathbf{x}) + I_b(\mathbf{x})$ . By assuming a uniformly colored surface lit with a single colored illumination, in a two-dimensional space: chromaticity intensity space, where its  $x$ -axis representing  $\tilde{c}$  and its  $y$ -axis representing  $\tilde{I}$ , with  $\tilde{I} = \max(I_r, I_g, I_b)$ , the diffuse pixels are always located at the right side of the specular pixels, due to maximum chromaticity definition (6). In addition, using either the chromaticity or the maximum chromaticity definition, the chromaticity values of the diffuse points will be constant, regardless of the variance of  $m_d(\mathbf{x})$ . In contrast, the chromaticity values of specular points will vary with regard to the variance of  $m_s(\mathbf{x})$ , as shown in Figure 1. From these different characteristics of specular and diffuse points in the chromaticity intensity space, we devised specular-to-diffuse mechanism. The details are as follows.

When two pixels, a specular pixel  $I_c(\mathbf{x}_1)$  and a diffuse pixel  $I_c(\mathbf{x}_2)$ , with the same diffuse color ( $\Lambda_c$ ) are projected into the chromaticity intensity space, the location of the diffuse point will be at the right side of the specular point. If the color of the specular component is pure white:  $\Gamma_r(\mathbf{x}_1) = \Gamma_g(\mathbf{x}_1) = \Gamma_b(\mathbf{x}_1)$ , by subtracting all channels

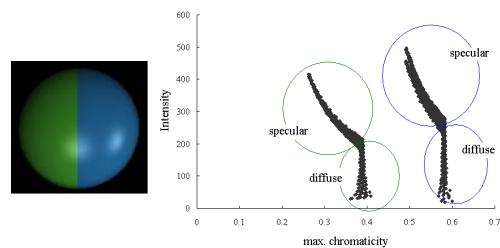


Figure 1: a. Synthetic image b. The projection of the synthetic image pixels into the chromaticity intensity space

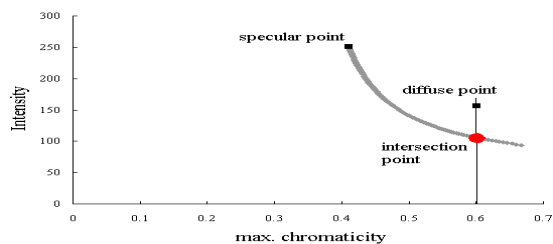


Figure 2: Specular-to-diffuse mechanism. The intersection point equals to the diffuse component of the specular pixel. By knowing diffuse chromaticity of the diffuse pixel, the intersection point can be obtained.

of the specular pixel's intensity using a small scalar number iteratively, and then projecting them into chromaticity intensity space, we will find that the points form a curved line in the space, as shown in Figure 2. This curved line follows this equation (see the Appendix for complete derivation):

$$I_c = m_d(\Lambda_c \Sigma \Gamma_i - \Gamma_c \Sigma \Lambda_i) \left( \frac{c}{c \Sigma \Gamma_i - \Gamma_c} \right) \quad (7)$$

We can observe in Figure 2 that a certain point in the curved line intersects with a vertical line that represents the chromaticity value of the diffuse point. At this intersection,  $m_s$  of the specular pixel equals to zero. Therefore, the intersection point is crucial, because it indicates the diffuse component of the specular pixel.

Mathematically, the intersection point (the diffuse component of the specular pixel) can be calculated as follows. In the previous section,  $\Lambda_c$  and  $\Gamma_c$  have already been defined as integral functions of normalized reflectance spectral, normalized illumination spectral and camera sensitivity. Besides this definition, we can also define both of them in terms of chromaticity. In this term, for all diffuse pixels,  $\Lambda_c = c$  and  $m_d = \Sigma I_i$ . In contrast, for specular pixels, as  $m_d$  cannot be canceled out,  $\Lambda_c \neq c$  and  $m_d \neq \Sigma I_i$ . In addition, for both diffuse and specular pixels we can set  $\Sigma \Gamma_i = \Sigma \Lambda_i = 1$ , and  $\Gamma_c = \frac{1}{3}$  as  $\Gamma_r = \Gamma_g = \Gamma_b$ . Hence, from Equation (7) we can derive the total diffuse intensity of specular pixels as:

$$\Sigma I_i^{diff} = \frac{\tilde{I}(3\tilde{c} - 1)}{\tilde{c}(3\tilde{\Lambda} - 1)} \quad (8)$$

where  $\tilde{\Lambda} = \max(\Lambda_r, \Lambda_g, \Lambda_b)$ , and  $I_c^{diff}$  is the diffuse component.

Thus, to calculate  $\Sigma I_i^{diff}(\mathbf{x}_1)$ , the value of  $\tilde{\Lambda}(\mathbf{x}_1)$  is required, which can be obtained from the diffuse pixel; since if the two pixels have the same surface color, then  $\tilde{\Lambda}(\mathbf{x}_1) = \tilde{\Lambda}(\mathbf{x}_2)$ . Having obtained  $\Sigma I_i^{diff}(\mathbf{x}_1)$ , the specular component is calculated using:

$$m_s(\mathbf{x}_1) = \frac{\Sigma I_i(\mathbf{x}_1) - \Sigma I_i^{diff}(\mathbf{x}_1)}{3} \quad (9)$$

Finally, by subtracting the specular pixel intensity with the specular component, the diffuse component is obtainable:

$$I_c^{diff}(\mathbf{x}_1) = I_c(\mathbf{x}_1) - m_s(\mathbf{x}_1) \quad (10)$$

To compute  $I_c^{diff}$  correctly, the mechanism needs linearity between the camera output and the flux of incoming light intensity.

In the case of the above two pixels, the mechanism can successfully obtain reflection components because the diffuse chromaticity ( $\tilde{c}(\mathbf{x}_2)$ ) is known. Unfortunately, given a multicolored image, the diffuse chromaticity for each color is unknown, which in fact, is the main problem of separating reflection component using a single multicolored image.

Nevertheless, although it cannot directly separate the reflection component, from the mechanism, we know that the diffuse component of a specular pixel lies somewhere in the curved line. This phenomenon is important in our separating method and is useful for generating a pseudo-code of diffuse component.

**Specular-Free Image.** To generate a pseudo-code of diffuse component or, as we call *specular-free image*, we simply set the diffuse chromaticity ( $\tilde{\Lambda}$  in Equation (8)) equal to an arbitrary scalar value ( $0 < \tilde{\Lambda} \leq 1$ ), for all pixels regardless their color. For instance, Figure 3.a shows an image of a multicolored scene. By setting  $\tilde{\Lambda} = 0.5$  for all pixels, we can obtain an image that is geometrically identical to the diffuse component of the input image (Figure 3.b). The difference of both is in their surface colors.

This technique can successfully remove highlights mainly because the saturation values of all pixels are made constant with regard to the maximum chromaticity, while retaining their hue [2, 1]. However, in the case of two adjacent objects that have the same hue but different saturation, using this technique will cause the color boundary of the two objects to disappear.

Formally, the specular-free image can be described as:

$$\hat{I}_c(\mathbf{x}) = m_d(\mathbf{x})\hat{\Lambda}_c(\mathbf{x}) \quad (11)$$

where  $m_d$  is exactly identical to  $m_d$  in Equation (5). In order to avoid negative values of  $\hat{I}_c(\mathbf{x})$ , the arbitrary scalar value of maximum diffuse chromaticity ( $\tilde{\Lambda}$ ) is obtained from the smallest maximum chromaticity of the input image.



Figure 3: a. Normalized image after removing achromatic pixels, below-camera-dark pixels and saturated pixels. b. Specular-free image by setting  $\tilde{\Lambda} = 0.5$ . The specular components are perfectly removed; the difference is only in the surface color.

Generating the specular-free image using the specular-to-diffuse mechanism is a one-pixel-based operation that requires only a single colored image without any segmentation process. Therefore, it is simple and probably useful for many applications in computer vision that do not need actual surface color but suffer from highlights.

### 3.2 Intensity Logarithmic Differentiation

Given only one colored pixel, to determine whether it is diffuse or specular pixel is an ill posed problem. Since, in a linear equation such as Equation (5), only from a single value of  $I_c$ , we cannot determine whether  $m_s$  is equal to zero. Thus, instead of using a single pixel we use two pixels that are spatially adjacent. We will show that using only two adjacent pixels that have the same diffuse chromaticity ( $\Lambda_c$ ), whether they are diffuse is determinable. This kind of local operation is indispensable for our method to deal with textured surface.

Our technique is principally based on intensity logarithmic differentiation of the normalized image and the specular free image. First, by assuming uniform color pixels ( $\Lambda_c$  becomes independent from  $\mathbf{x}$ ), we apply the logarithm and spatial differentiation operations on Equation (5):

$$\frac{d}{dx} \log(I_c(\mathbf{x})) = \frac{d}{dx} \log(m_d(\mathbf{x})\Lambda_c + m_s(\mathbf{x})) \quad (12)$$

For diffuse pixels where  $m_s = 0$ , the equation becomes:

$$\frac{d}{dx} \log(I_c(\mathbf{x})) = \frac{d}{dx} \log(m_d(\mathbf{x})) \quad (13)$$

The partial differentiation is applied w.r.t. both  $x$  and  $y$ ; yet the operations are done independently. For diffuse pixels

( $m_s = 0$ ), the logarithmic differentiation produces an equation that is independent from the surface color ( $\Lambda_c$ ), while for specular pixels ( $m_s \neq 0$ ), the dependence still prevails.

If we apply the same assumption and operations on the specular-free image (Equation (11)), which  $m_s$  for all pixels equal to zero, then we will obtain an identical result to Equation (13). Therefore, by computing the difference between the logarithmic differentiation of the normalized image and the logarithmic differentiation of the specular-free image, whether two neighboring pixels are diffuse pixels is determinable.

$$\Delta(\mathbf{x}) = d\log(I_c(\mathbf{x})) - d\log(\hat{I}_c(\mathbf{x})) \quad (14)$$

$$\Delta(\mathbf{x}) \begin{cases} = 0 & : \text{diffuse} \\ \neq 0 & : \text{specular or boundary} \end{cases} \quad (15)$$

Furthermore, by examining every pixel of an image, the logarithmic differentiation can be used to determine whether the image contains only diffuse pixels.

In multicolored surfaces, as shown in Equation (15), there is still an ambiguity between specular and boundary pixels. Since using two neighboring pixels that have different surface color, the difference of the logarithmic differentiation does not equal to zero, even though the pixels are diffuse pixels. Theoretically, by extending the number of pixels into at least four neighboring pixels, it is possible to distinguish them. However, in real images, camera noise and surface noise (surface variance) [5] make such identification become error-prone; thus, we need another more robust analysis, which will be described in the next subsection.

### 3.3 Color Boundary

A number of methods have been proposed to solve the color boundary problem, which is also known as the problem of material changes [6, 4]. Unlike those methods, we use a simple chromaticity-based method to handle the problem. We define  $r = \frac{I_r}{I_r + I_g + I_b}$  and  $g = \frac{I_g}{I_r + I_g + I_b}$ , and apply the below decision rule to solve the problem:

$$(\Delta r(\mathbf{x}) > thR \quad \text{and} \quad \Delta g(\mathbf{x}) > thG) \begin{cases} 1: \text{boundary} \\ 0: \text{non-boundary} \end{cases} \quad (16)$$

where  $thR$  and  $thG$  are the small scalar numbers.  $\Delta r = r_{in} - r_{spec}$  and  $\Delta g = g_{in} - g_{spec}$ , with  $r_{in}$  and  $r_{spec}$  are red channel chromaticity of normalized input image and specular-free image. While,  $g_{in}$  and  $g_{spec}$  are green channel chromaticity of normalized input image and specular-free image.

This simple chromaticity thresholding is sufficient for two neighboring pixels. Since if two neighboring pixels have the same surface color, the chromaticity difference is small, even for specular pixels. This is one of the advantages of our local, two-neighboring-pixels operation. Moreover, the above thresholding can also solve the problem of two adjacent objects that have the same hue but different saturation, as long as the saturation difference is not less

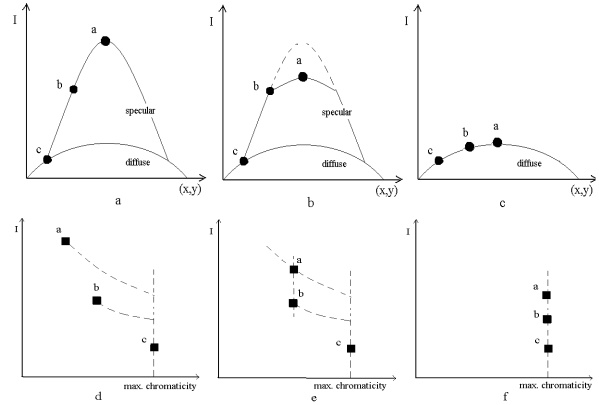


Figure 4: Basic idea of the iterative framework using local 2-pixels operation. Top row, spatial-intensity space: a. Initial condition b. First looping c. Final condition; Bottom row, chromaticity intensity space: d. Initial condition e. First looping f. Final condition

than the thresholds. Fortunately, in practice, even if the saturation difference is less than the thresholds, it does not affect the result much; since it means that the objects almost have the same color, so that it is unnecessary to distinguish them. In addition, we have no problem even if the above thresholding wrongly deems the shadow boundary as a color boundary, since we have nothing to do with shadow.

### 3.4 Iterative Framework

Figure 4 illustrates the basic idea of our iterative framework. For the sake of simplicity, the illustration assumes uniform surface color of three neighboring pixels: one diffuse pixel ( $c$ ) and two specular pixels ( $a$  and  $b$ ), as shown in Figure 4.a. If the pixels are transformed into the chromaticity intensity space, we will obtain the distribution of the maximum chromaticity as illustrated in Figure 4.d.

In considering a two-pixel operation, the iteration begins with comparing the maximum chromaticity of point  $a$  and point  $b$ . From the maximum chromaticity definition in Equation (6), it is known that the smaller  $m_s$  the bigger maximum chromaticity value. In other words, point  $b$  is more diffuse than point  $a$ . Thus, by shifting point  $a$  using the specular-to-diffuse mechanism w.r.t the maximum chromaticity of point  $b$ , the more diffuse pixel  $a$  can be obtained, i.e., the intensity of pixel  $a$  becomes decreased and its chromaticity becomes identical to point  $b$ 's, as illustrated in Figure 4.b and 4.e respectively. Using the same process, in the second iteration, the maximum chromaticity of point  $b$  and point  $c$  are compared and then shifted. When the maximum chromaticity of point  $b$  equals to the maximum chromaticity of point  $c$ , the intensity of pixel  $b$  becomes equal to its diffuse component. The same operation is done for all pixels iteratively until their maximum chromaticity becomes the same (Figure 4.f), which as a result, produces the diffuse components of the three pixels (Figure 4.c).

However, the above termination condition: looping until the maximum chromaticity of all pixels are the same, is workable only for a uniform colored surface. In multicolored surfaces, such termination condition will produce incorrect separation results. Thus, instead of using same maximum chromaticity condition, we use the logarithmic differentiation, as explained in the subsection 3.2, to verify whether the image contains only diffuse pixels. Algorithm 4.1 shows the pseudo-code of the iteration method for both uniform and multicolored surface; a detailed explanation will be provided in Section 4.

## 4 Implementation and Results

**Algorithm 4.1:** ITERATION( $N, S, \epsilon$ )

**comment:** N=normalized-image; S= specular-free-image  
 $\Delta = \text{delta}(N, S, \epsilon);$   
**while**  $\text{any}(\Delta(\mathbf{x}) > \epsilon)$   
  **for**  $\mathbf{x} \leftarrow 0$  **to**  $\text{sizeof}(N)-1$   
    **if**  $\mathbf{x}.\text{flag} == \text{diffuse}$   
      **then**  $\text{next}(\mathbf{x});$   
    **if**  $\text{IsBoundary}(\mathbf{x}, \mathbf{x} + 1) == \text{true}$   
      **then**  $\begin{cases} \mathbf{x}.\text{flag} = \text{boundary}; \\ (\mathbf{x} + 1).\text{flag} = \text{boundary}; \\ \text{next}(\mathbf{x}); \end{cases}$   
    **if**  $\tilde{c}(\mathbf{x}) == \tilde{c}(\mathbf{x} + 1)$   
      **then**  $\begin{cases} \mathbf{x}.\text{flag} = \text{noise}; \\ (\mathbf{x} + 1).\text{flag} = \text{noise}; \\ \text{next}(\mathbf{x}); \end{cases}$   
     $N(\mathbf{x}) = \text{Specular2Diffuse}(I_c(\mathbf{x}), I_c(\mathbf{x} + 1));$   
     $\text{next}(\mathbf{x});$   
   $\Delta = \text{delta}(N, S, \epsilon);$   
**return** ( $N$ )  
**comment:** N = normalized diffuse component

Algorithm 4.1 begins with executing function  $\text{delta}(N, S, \epsilon)$ , which calculates the difference of the intensity logarithmic differentiation of the normalized image ( $N$ ) and the specular-free image ( $S$ ). In a discrete operation, the logarithmic differentiation is done using:  $d\log(I(\mathbf{x})) = \log(\Sigma I_i(\mathbf{x} + 1)) - \log(\Sigma I_i(\mathbf{x}))$ . After calculating the difference of  $d\log$ , the function labels the pixels of the normalized image: for pixels that have  $\Delta$  more than  $\epsilon$  ( $\approx 0$ ), they are labeled "specular"; otherwise "diffuse".

Then, if there are some pixels labeled "specular", for each of them, the algorithm examines whether the pixel and its neighbor are boundary pixels. If so, they are labeled "boundary"; else, at least one of them must be a specular pixel.

At this point, ideally we can start to use the specular-to-diffuse mechanism. However, before we apply the mechanism to both pixels, additional checking is useful, i.e., whether both pixels' maximum chromaticity are the same. If they are the same, then the pixels are labeled "noise". The

reason that they are noise and not specular pixels is because specular pixels never have the same maximum chromaticity.

After verifying that the pixels are not noise, using the specular-to-diffuse mechanism, the intensity and maximum chromaticity value of the pixel that have smaller  $\tilde{c}$  is shifted w.r.t. the pixel with bigger  $\tilde{c}$ . This is applied to all pixels, and produces a more diffuse normalized image. Using this image, function  $\text{delta}(N, S, \epsilon)$  is executed once again. This time, pixels labeled "boundary" and "noise" are ignored. Finally, if there is still any  $\Delta$  larger than  $\epsilon$ , then the iteration continues; if not, the separation terminates, which as a result, produces a diffuse component of the normalized image.

**Algorithm 4.2:** CONTROLLEDTHRESHOLD( $N, S$ )

**comment:** N=normalized-image; S= specular-free-image  
 $\text{stepTH} = \text{InitialThreshold};$   
**while**  $\text{stepTH} > \epsilon$   
   $\Delta = \text{delta}(N, S, \epsilon);$   
  **if**  $\text{any}(\Delta(\mathbf{x}) > \text{stepTH})$   
    **then**  $\text{Iteration}(N, S, \text{stepTH});$   
   $\text{stepTH} = \text{stepTH} - \delta;$   
   $\text{ResetAllLabels}();$   
   $\text{Renormalization}(N);$   
**return** ( $N$ );  
**comment:** N=actual diffuse component

In our implementation, we define  $\epsilon = 0$ . For color boundary thresholds ( $thR$  and  $thG$ ), we set both of them with the same number ranging from 0.05 to 0.1. The numbers are chosen by considering camera noise, illumination color variance, ambient light (some considerably small interreflections) and surface color variance (although human perception deems that the color surface is uniform, in fact there is still color variance due to dust, imperfect painting, etc.). In computing the normalized specular reflection components, practically we remove pixels that do not have pure-white color, since they are possibly produced by influence of noise.

For a more stable and robust algorithm we add an algorithm that controls the decrease of the threshold of  $\Delta$  step-by-step, as described in Algorithm 4.2. In function  $\text{Iteration}(N, S, \epsilon)$ ,  $\text{stepTh}$  will replace  $\epsilon$ , which in our implementation its initial value equals to 0.5. In fact, the initial should be set as large as possible, yet by considering the computational time the number is chosen. To obtain more accurate results, the smaller subtracting number ( $\delta$ ) is better and, in our implementation, we set it equal to 0.01. To avoid saturated pixels, HDR (High Dynamic Range) images can be used.

The experiments were conducted using a CCD camera: SONY DXC-9000 (a progressive 3 CCD digital camera) by setting the gamma off; while the illumination chromaticity was estimated using a color constancy algorithm [17]. Figure 5 shows the separation result of Figure 3.a., where the objects were lit with solux halogen lamp. Figure 6 to Figure



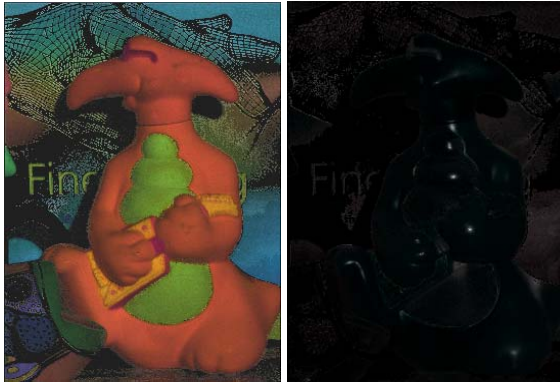


Figure 5: a. Diffuse component of Figure 3.a b. Specular component of Figure 3.a

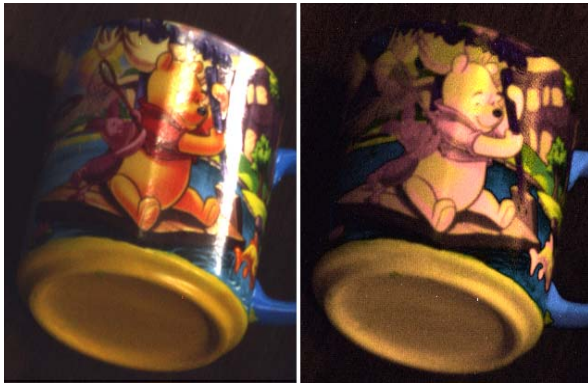


Figure 6: a. a complex textured surface of a cup lit with fluorescent lights. b. the specular-free image by setting  $\bar{\Lambda} = 0.5$

9 show complex textured images, specular-free images, and separation results. All images were taken under fluorescent lamps in uncontrolled-indoor environment.

## 5 Conclusion

We have proposed a novel method to separate diffuse and specular reflection components. The main insight of our method is in the chromaticity-based iteration with regard to the logarithmic differentiation of the specular-free image. Using our method, the separation problem in textured surfaces with complex multicolored scene can be resolved, as long as each color area has diffuse pixels. There are at least three factors that are crucial and could become the main contributions of our method, i.e., the specular-to-diffuse mechanism, the specular-free image, and the logarithmic differentiation-based iteration framework. The experimental results on complex textured images show that the proposed method is accurate and robust.

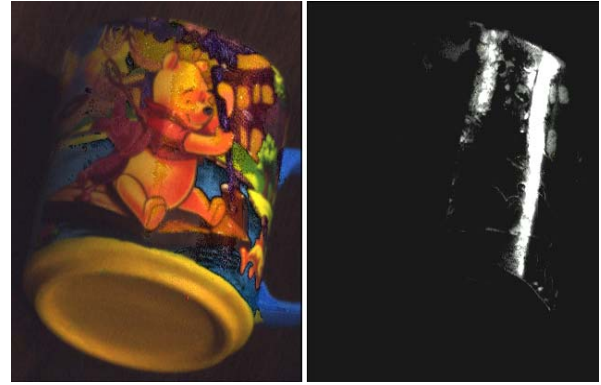


Figure 7: a. diffuse component b. specular component

## Acknowledgements

This research was, in part, supported by Japan Science and Technology (JST) under CREST Ikeuchi Project.

## Appendix

A complete derivation of the correlation between illumination chromaticity and image chromaticity is described as follows:

$$\begin{aligned} I_c(\mathbf{x}) &= m_d(\mathbf{x})\Lambda_c(\mathbf{x}) + m_s(\mathbf{x})\Gamma_c \\ c(\mathbf{x}) &= \frac{I_c(\mathbf{x})}{I_r(\mathbf{x}) + I_g(\mathbf{x}) + I_b(\mathbf{x})} \\ c(\mathbf{x}) &= \frac{m_d(\mathbf{x})\Lambda_c(\mathbf{x}) + m_s(\mathbf{x})\Gamma_c}{m_d(\mathbf{x})\Sigma(\Lambda_i(\mathbf{x})) + m_s(\mathbf{x})\Sigma(\Gamma_i)} \end{aligned}$$

For local (pixel based) operation the location ( $\mathbf{x}$ ) can be removed:

$$\begin{aligned} m_s(c\Sigma\Gamma_i - \Gamma_c) &= m_d\Lambda_c - cm_d\Sigma\Lambda_i \\ m_s &= \frac{cm_d\Sigma\Lambda_i - m_d\Lambda_c}{\Gamma_c - c\Sigma\Gamma_i} \end{aligned}$$

Substituting  $m_s$  in the definition of  $I_c$  with  $m_s$  in the last equation:

$$I_c = m_d(\Lambda_c\Sigma\Gamma_i - \Gamma_c\Sigma\Lambda_i)\left(\frac{c}{c\Sigma\Gamma_i - \Gamma_c}\right)$$

## References

- [1] R. Bajscy, S.W. Lee, and A. Leonardis. Detection of diffuse and specular interface reflections by color image segmentation. *International Journal of Computer Vision*, 17(3):249–272, 1996.
- [2] M. D’Zmura and P. Lennie. Mechanism of color constancy. *Journal of Optics Society of America A.*, 3(10):1162–1672, 1986.
- [3] G.D. Finlayson and G. Schaefer. Solving for color constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision*, 42(3):127–144, 2001.



Figure 8: a. a complex multicolored scene lit with fluorescent lights. b. The specular-free image by setting  $\tilde{\Lambda} = 0.5$

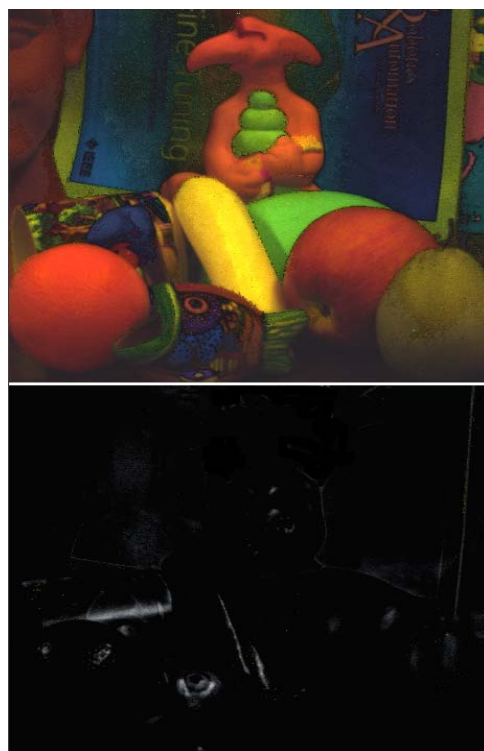


Figure 9: a. diffuse component b. specular component

- [4] R. Gershon, A.D. Jepson, and J.K. Tsotsos. Ambient illumination and the determination of material changes. *Journal of Optics Society of America A.*, 3(10):1700–1707, 1986.
- [5] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.
- [6] J.M. Rubin and W.A. Richard. Color vision: representing material changes. *AI Memo 764*, MIT Artificial Intelligence Lab. Cambridge, Mass., 1984.
- [7] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlights in color images. *International Journal of Computer Vision*, 2:7–32, 1990.
- [8] H.C. Lee, E.J. Breneman, and C.P. Schulte. Modeling light reflection for computer color vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(4):402–409, 1990.
- [9] S.W. Lee and R. Bajcsy. Detection of specularities using color and multiple views. *Image and Vision Computing*, 10:643–653, 1990.
- [10] S. Lin, Y. Li, S. B. Kang, X. Tong, and H.Y. Shum. Diffuse-specular separation and depth recovery from image sequences. In *proceeding of Europe Conference of Computer Vision*, pages 210–224, 2002.
- [11] S. Lin and H.Y. Shum. Separation of diffuse and specular reflection in color images. In *proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 341–346, 2001.
- [12] S.K. Nayar, X.S. Fang, and T. Boult. Separation of reflection components using color and polarization. *International Journal of Computer Vision*, 21(3), 1996.
- [13] J.S. Park and J.T. Tou. Highlight separation and surface orientation for 3-d specular objects. In *proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, 1990.
- [14] J.P.S. Perkkinen, J. Hallikainen, and T. Jasskelainen. Characteristic spectra of munsell colors. *Journal of Optics Society of America A.*, 6(2):318–322, 1989.
- [15] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *Journal of Optics Society of America A.*, 11(11):2990–3002, 1994.
- [16] S. Shafer. Using color to separate reflection components. *Color Research and Applications*, 10:210–218, 1985.
- [17] R. T. Tan, K. Nishino, and K. Ikeuchi. Illumination chromaticity estimation using inverse-intensity chromaticity space. *proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pages 673–680, 2003.
- [18] S. Tominaga and B.A. Wandell. Standard surface-reflectance model and illumination estimation. *Journal of Optics Society of America A.*, 6(4):576–584, 1989.
- [19] L.B. Wolff and T. Boult. Constraining object features using polarization reflectance model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):635–657, 1991.