Digital Image Processing Report

Separating Reflection Components of Textured Surfaces using a Single Image

Indraneil Paul

Romil Aggarwal

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.

Introduction

In a world where we are surrounded by light sources if various spectral characteristics and wavelengths, it rarely can ever be guaranteed that we will be able to produce a glare free image at will, when the situation demands so. Several important and at time mission critical applications, such as medical image processing and surveillance require us to consistently produce glare and artifact free images that can be scrutinized to the finest level of detail. Hence we attempt to approach the problem in a manner that does not necessitate any prior knowledge of the geometry of the object and its chromaticity profile. It is with this in mind that in the following sections we briefly explain the reflection model used, followed by the normalization of the color space so as to implicitly figure of the diffuse colors of a specular surface and restore the surface to the same as accurately as possible.

Reflection Model

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 [Formula 1] which we later modify to a simpler form [Formula 2].

One simplifying assumption made at this stage is that the color profile of the specular component of the total reflection closely mirrors the color profile of the source of the highlight. Further simplifying the model by ignoring camera noise and gain, an image taken by a digital camera can be described as shown in [Formula 3][Formula 4].

One drawback of our method is that to work at its best possible efficiency, the color of the highlight must be effectively white, i.e. the red, green and blue channels of the specular component of the total reflection must be equal. This problem is worked around by transforming the reflection model into a normalized space [Formula 5].

Pipeline Part 1: Specular to Diffuse Mechanism

We can exploit the fact that diffuse reflection scatters the light in all directions with roughly equal intensity, and hence the intensity of diffuse reflection of a light in a certain direction will in most cases be less than that from a specular reflection. Hence we now define a new quantity that allows us to discriminate between a diffuse and highlighted pixel of the same color. This is done by moving the points to a maximum chromaticity (x-axis) vs maximum intensity (y-axis) plane. In this plane all diffuse points will always lie to the right of all the specular points of the same color [Formula 6].

In the aforementioned space the diffuse points of a certain color will form a vertical line and the specular points will form a curve (provided they aren't highlighted by a light of the same color). The intersection of these curves is our solution for the diffuse color [Formula 7][Formula 8][Formula 9][Formula 10].

The final specular-free image is then generated as shown in [Formula 11].

Pipeline Part 2: Intensity Logarithmic Differentiation

We do however, still face a very significant problem. In case we are given a single point of a certain color and radiant intensity, telling if it as a diffuse or specular pixel presents a fundamentally ill-posed problem. However, this can be worked around by converting it into a determinable problem. In this case the trick is to use two spatially adjacent pixels of the same diffuse chromaticity, in which case whether a pixel is diffuse or not is determinable.

The technique revolves about the intensity logarithmic differentiation, that is performed locally according to [Formula 12]. This operation produces equations independent of surface colors for diffuse pixels.

However, our method is still not fool proof as this technique may lead to equations dependent on surface colors for diffuse pixels in some cases, due to camera noise or color boundaries on multicolored surfaces. Hence we also adopt a technique to reliably detect color boundaries.

Pipeline Part 3: Color Boundary Criteria

We define a color boundary criteria as shown in [Formula 13], that determines where the above stated criteria of logarithmic differentiation can and cannot be used without fault.

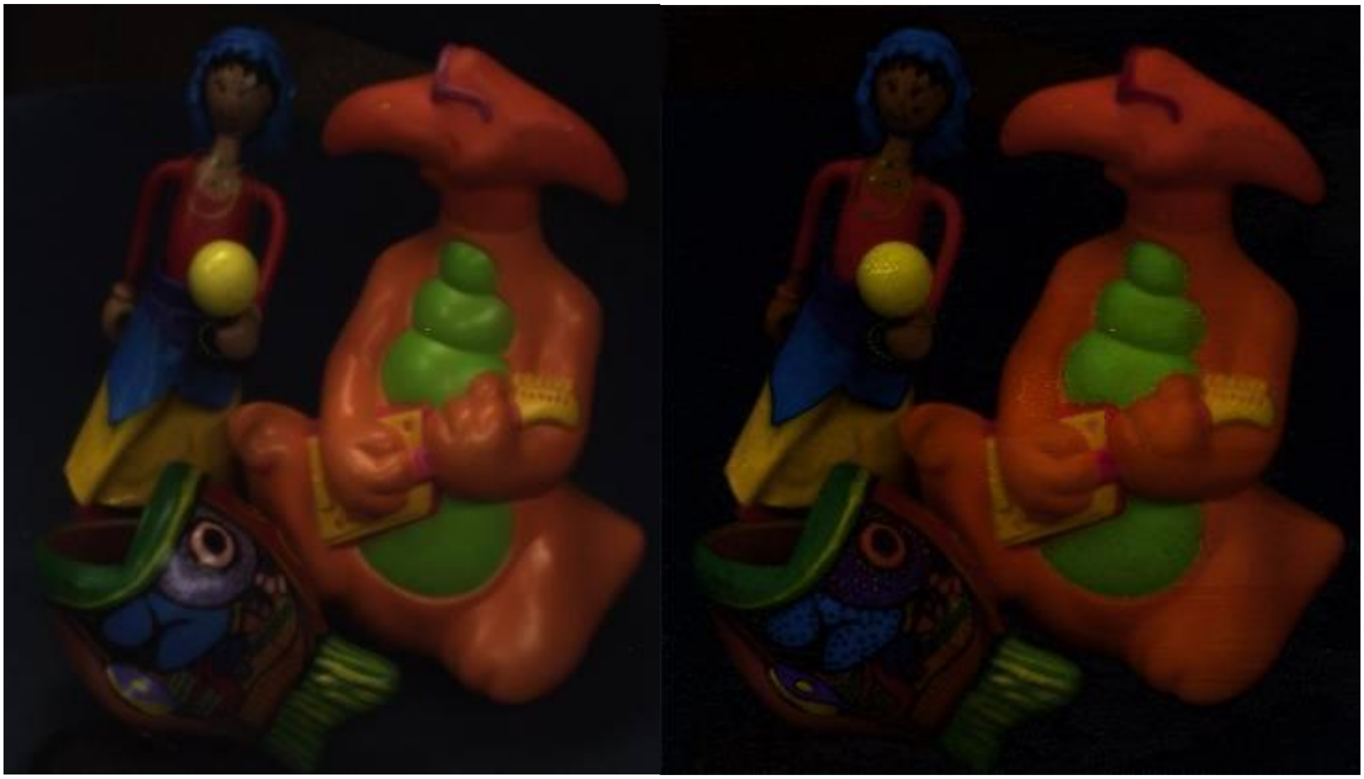
Pipeline Part 4: Iterative Formulation

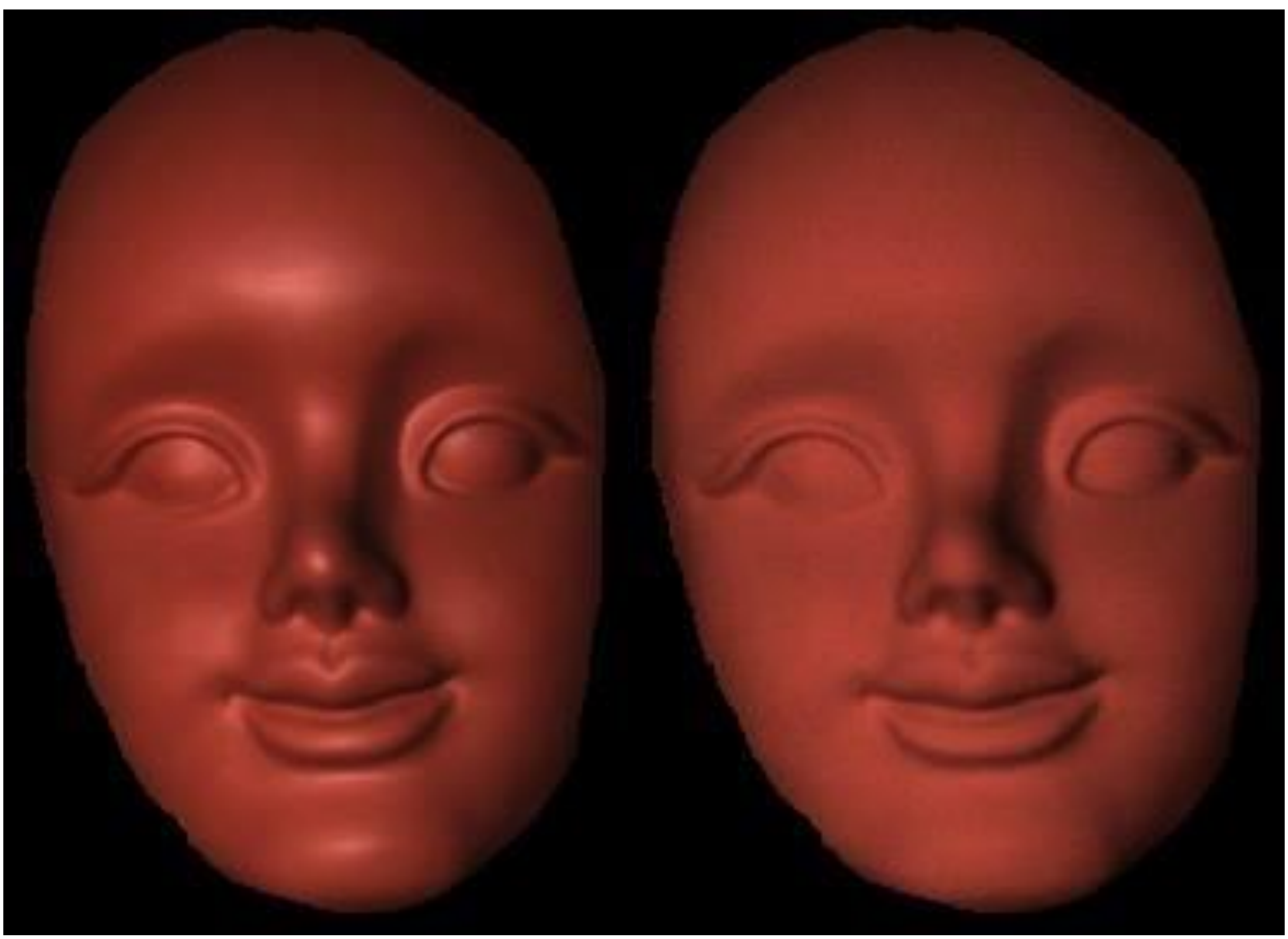
Now we need to determine a reliable termination condition for the whole process. We hence apply a locally iterative method where for every pixel, all the pixels in it's neighborhood are checked and all the pixels are then changed so that their diffuse chromaticity equals the maximum diffuse chromaticity in the neighborhood.

In the case of single color surfaces this is carried on till all the pixels have a common maximum diffuse chromaticity. However, in the case of multi colored surfaces, these are boundary constrained based on the condition shown in the previous section.

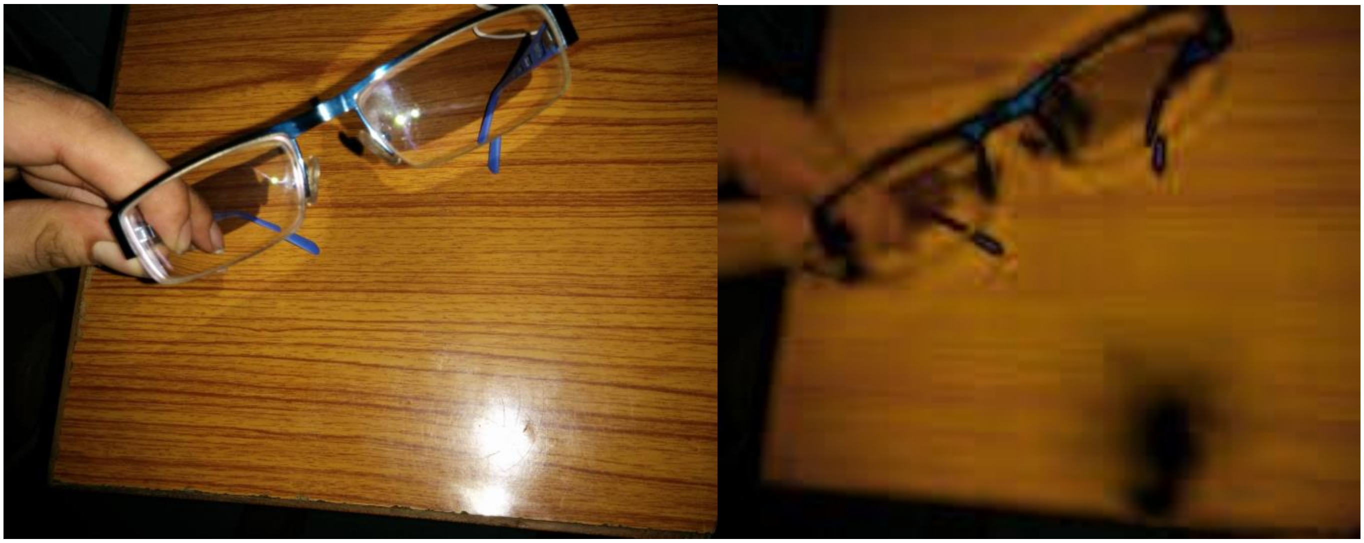
Results

The technique we have shown works very well for multicolored surfaces with arbitrary geometries. It also handles boundaries and noise well and produces results very quickly for moderately sized images.





However, it does suffer from a drawback that it inherits from the reflection model we used. The formation of ghosting cues due to superposition of double reflections from transparent and translucent surfaces causes the algorithm to yield faulty results.



Formulae

Formula 1

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

Formula 2

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

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

Formula 3

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

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 (Ω).

Formula 4

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

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.

Formula 5

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

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.

Formula 6

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

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

Formula 7

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

Formula 8

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

Formula 9

$$m_s(\mathbf{x_1}) = \frac{\Sigma I_i(\mathbf{x_1}) - \Sigma I_i^{diff}(\mathbf{x_1})}{3}$$

Formula 10

$$I_c^{diff}(\mathbf{x_1}) = I_c(\mathbf{x_1}) - m_s(\mathbf{x_1})$$

Formula 11

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

Formula 12

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

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

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

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

Formula 13

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:

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

Paper Referred: Separating Reflection Components of Textured Surfaces using a Single Image by Robbi Tan and Katush Ikeuchi.

http://ljk.imag.fr/membres/Bill.Triggs/events/iccv03/cdrom/iccv03/0870_tan.pdf