

A PDE Approach to Specularity Removal in Images

Comp 766: Shape Analysis in Computer Vision

Milena Scaccia

April 27, 2011

1 Introduction

Specular reflections are exhibited by a wide range of materials whose reflectance can be described as a linear combination of specular and diffuse components [5]. There are several benefits to separating an image into the two components. By isolating the diffuse component (which is often well-described by the *Lambertian model*), powerful Lambertian-based tools for tracking, reconstruction and recognition (e.g. shape-from-shading) can be more widely applied to real-world, non-Lambertian scenes. Specular reflectance itself plays an evident role in human perception. Based on this, several computer vision algorithms have been designed to successfully infer shape solely from specularities [3].



Figure 1: Specularities from a basket of fruit [5]

Recovering a diffuse component and a monochromatic specular component from a single three-channel RGB image is an ill-posed problem. In this work, we explore a framework by Mallick et al., which uses a partial differential equation (PDE) approach for separating the two components in images [5].

Their solution proposes to first transform the image into the SUV color space, which conveniently provides a partial separation of the diffuse and specular components. This holds true for surfaces which are well-represented by Shafer's dichromatic model (where the spectral distribution of the specular component is similar to that of the light source color, and the diffuse component relies on the material properties of the surface [5]). The separation will then be completed using spatio-temporal information by evolving a PDE that iteratively erodes the specular component locally at each pixel.

We evaluate this approach using images of textured and untextured surfaces. We observe that the erosion process for each of these cases essentially mimics a grassfire flow [12], but with different speed terms.

The remainder of this report is organized as follows. Section 2 provides background on the reflectance model, the SUV color space, and upwind derivatives. Section 3 presents Mallick et al.'s specularity removal

algorithm. Our results along with implementation details are presented in Section 4. We conclude with a discussion about advantages and limitations of the algorithm in Section 5.

2 Background

2.1 Shafer's Dichromatic Model of Reflectance

Surfaces well-represented by the dichromatic model have a specular component whose spectral distribution is similar to that of the light source color (illuminant) while the diffuse component heavily relies on the material properties of the surface.

The bidirectional reflectance distribution function (BRDF) for the dichromatic model can be expressed as

$$f(\lambda, \Theta) = g_d(\lambda)f_d + f_s(\Theta), \quad (1)$$

where λ is the wavelength of light, $\Theta = (\theta_i, \phi_i, \theta_r, \phi_r)$ parameterizes directions of incoming irradiance and outgoing radiance, g_d is the *spectral reflectance*, and f_d and f_s are the *diffuse* and *specular* BRDFs, respectively.

An RGB color vector contains three measurements, defined as $\mathbf{I} = [I_1, I_2, I_3]^T$, with

$$I_k = (D_k f_d + S_k f_s(\Theta)) \hat{\mathbf{n}} \cdot \hat{\mathbf{I}}, \quad (2)$$

where $D_k = \int C_k(\lambda)L(\lambda)g_d(\lambda)d\lambda$ is the effective *albedo* in the k^{th} channel and $S_k = \int C_k(\lambda)L(\lambda)d\lambda$ is the effective *source strength* as measured by the k^{th} sensor channel. Note that $L(\lambda)$ is the *spectral power distribution* of the light source, and $C_k(\lambda)$ is the *camera sensitivity function*. The terms $\hat{\mathbf{n}}$ and $\hat{\mathbf{I}}$ denote the surface *normal* and the light source *direction*, respectively. The illuminant is defined as source vector $\mathbf{S} = [S_1, S_2, S_3]^T$. It is a unit vector with base at the origin and has an azimuthal and elevation angle [Figure 2].

2.2 SUV Color Space

The SUV color space is an illuminant-dependent color space. It is defined as a rotation of the RGB space where one axis (S) is aligned with the illuminant color and thus contains the complete specular component, while the other two channels (U and V) are purely diffuse. Let \mathbf{I}_{RGB} denote an image in RGB space, \mathbf{I}_{SUV} the image in SUV space, and \mathbf{R} the rotation matrix with the following properties:

$$\mathbf{I}_{SUV} = \mathbf{R}\mathbf{I}_{RGB}, \quad \mathbf{R}\mathbf{S} = [1 \ 0 \ 0]^T. \quad (3)$$

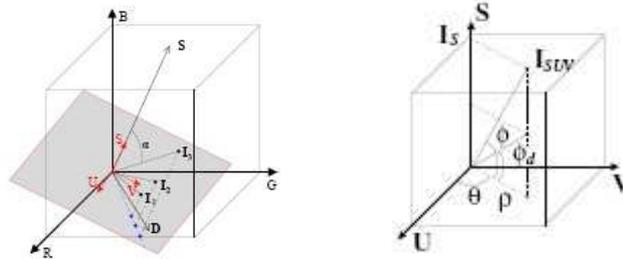


Figure 2: (a) RGB to SUV Transformation (b) SUV parameterization into ρ , θ and ϕ . [5]

Expanding (3) using equation (2), we obtain

$$\mathbf{I}_{SUV} = (\mathbf{R}\mathbf{D}f_d + \mathbf{R}\mathbf{S}f_s(\Theta))\hat{\mathbf{n}} \cdot \hat{\mathbf{1}}. \quad (4)$$

If we let \mathbf{r}_i^T denote the i^{th} row of \mathbf{R} , we observe that the expressions for the U and V channels are devoid of the specular component due to the property $\mathbf{R}\mathbf{S} = [1 \ 0 \ 0]^T$ from equation (3):

$$I_U = \mathbf{r}_2^T \mathbf{D}f_d \hat{\mathbf{n}} \cdot \hat{\mathbf{1}}, \quad I_V = \mathbf{r}_3^T \mathbf{D}f_d \hat{\mathbf{n}} \cdot \hat{\mathbf{1}} \quad (5)$$

The S channel contains the full specular component (plus some diffuse information, hence being a partial separation):

$$I_S = \mathbf{r}_1^T \mathbf{D}f_d \hat{\mathbf{n}} \cdot \hat{\mathbf{1}} + f_s(\Theta) \cdot \hat{\mathbf{1}}. \quad (6)$$

2.3 Upwind Derivatives

Non-linear PDEs are defined at points where partial derivatives exist. During the evolution of a PDE, a discontinuity known as a *shock* may develop. An example occurs in the grassfire flow [12] which is the evolution of a curve C in the direction of the inner normal $C_t = \Gamma \hat{N}$, whose level-set form is given by $\Psi_t = \Gamma \|\nabla \Psi\|$ with speed $\Gamma = 1$. To mitigate this problem, standard derivatives can be replaced by one-sided derivatives by implementing what is known as an *upwinding scheme* [11].

Given speed function Γ , and time step Δt , the level sets can be approximated by a *first-order space convex*:

$$\Psi^{m+1} = \Psi^m - \Delta t [\max(\Gamma, 0) \nabla^+ + \min(\Gamma, 0) \nabla^-], \quad (7)$$

where

$$\nabla^+ = [\max(\Psi_x^-, 0)^2 + \min(\Psi_x^+, 0)^2 + \max(\Psi_y^-, 0)^2 + \min(\Psi_y^+, 0)^2]^{1/2} \quad (8)$$

$$\nabla^- = [\max(\Psi_x^+, 0)^2 + \min(\Psi_x^-, 0)^2 + \max(\Psi_y^+, 0)^2 + \min(\Psi_y^-, 0)^2]^{1/2}, \quad (9)$$

and

$$\begin{aligned} \Psi_x^+ &= \frac{\Psi(i + \Delta x, j) - \Psi(i, j)}{\Delta x}, & \Psi_x^- &= \frac{\Psi(i, j) - \Psi(i - \Delta x, j)}{\Delta x} \\ \Psi_y^+ &= \frac{\Psi(i, j + \Delta y) - \Psi(i, j)}{\Delta y}, & \Psi_y^- &= \frac{\Psi(i, j) - \Psi(i, j - \Delta y)}{\Delta y} \end{aligned} \quad (10)$$

are forward and backward spatial derivatives, respectively, (see [11], equation(4.7)).

The above extends to *second-order space convex*:

$$\nabla^+ = [\max(A, 0)^2 + \min(B, 0)^2 + \max(C, 0)^2 + \min(D, 0)^2]^{1/2} \quad (11)$$

$$\nabla^- = [\max(B, 0)^2 + \min(A, 0)^2 + \max(D, 0)^2 + \min(C, 0)^2]^{1/2}, \quad (12)$$

where

$$A = \Psi_x^- + \frac{\Delta x}{2} m(\Psi_{xx}^{--}, \Psi_{xx}^{+-}) \quad (13)$$

$$B = \Psi_x^+ - \frac{\Delta x}{2} m(\Psi_{xx}^{++}, \Psi_{xx}^{+-}) \quad (14)$$

$$C = \Psi_y^- + \frac{\Delta y}{2} m(\Psi_{yy}^{--}, \Psi_{yy}^{+-}) \quad (15)$$

$$D = \Psi_y^+ - \frac{\Delta y}{2} m(\Psi_{yy}^{++}, \Psi_{yy}^{+-}) \quad (16)$$

$$(17)$$

and the switch function (which turns itself off when a shock is detected) is defined as

$$m(x, y) = \begin{cases} \begin{cases} x & |x| \leq |y| \\ y & |x| > |y| \end{cases} & xy \geq 0 \\ 0 & xy < 0. \end{cases} \quad (18)$$

3 Algorithm

3.1 Mallick et al.'s Specularity Removal Algorithm

To complete the separation between specular and diffuse components, a family of PDEs is derived as follows. The input image is first transformed to SUV space using equation (3) above. According to [4] we can take $\mathbf{R} = [\mathbf{R}_G(\phi_S)][\mathbf{R}_B(-\theta_S)]$, where (ϕ_S, θ_S) are the elevation and azimuthal angles of the source vector \mathbf{S} in the RGB coordinate system. Note that \mathbf{S} is the effective illuminant color, and the same illuminant color is assumed over the entire image.

The SUV image is then reparametrized using cylindrical and spherical coordinates:

$$\rho = \sqrt{I_U^2 + I_V^2} \quad \theta = \tan^{-1}\left(\frac{I_U}{I_V}\right), \quad \phi = \tan^{-1}\left(\frac{I_S}{\rho}\right), \quad (19)$$

where $\phi = \phi_d + \phi_s$ is a linear combination of specular and diffuse components, and ρ, θ are purely diffuse (since they depend only on the U and V channels) and contain hue and shading information, respectively [5].

The problem of computing the separation is *reduced* to that of estimating $\phi_d(x, y)$ at each image point. Once known, the RGB diffuse component is recovered by inverting equations (19) and (3) with ϕ replaced with ϕ_d .

The multi-scale erosion of ϕ by a structuring set $B \subseteq \mathbb{R}^2$, at scale t (where t is time), is given as

$$\Psi(\mathbf{x}, t) = (f \ominus tB)(\mathbf{x}) \triangleq \inf\{\phi(\mathbf{x} + \Delta\mathbf{x}) : \Delta\mathbf{x} \in tB\}, \quad (20)$$

where B is compact, $tB \triangleq \{tb : b \in B\}$, and $\mathbf{x} = (x, y)$. According to [5] [1], this multi-scale erosion is computed by evolving the PDE

$$\Psi_t = \lim_{\Delta t \rightarrow 0} \frac{\inf\{\nabla\Psi^T \Delta\mathbf{x} : \Delta\mathbf{x} \in \Delta t B\}}{\Delta t}, \quad (21)$$

where $\nabla\Psi$ is the two-dimensional spatial gradient ¹ of Ψ evaluated at time t .

We will take ϕ as our initial Ψ . Basically, when we set $\Psi(\mathbf{x}, 0) = \phi(\mathbf{x})$, and evolve the PDE to iteratively erode the specular contribution to ϕ , the value of ϕ at each image point is replaced by the minimum value within a neighborhood established by B . Since $\phi_d \leq \phi$, Mallick et al. claim that when the image contains at least one image point for which $\phi_s = 0$ (purely diffuse), the process will converge to ϕ_d as t is sufficiently large.

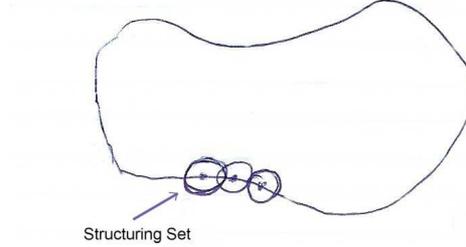


Figure 3: Disk-like Structuring Set

This PDE is considered for different cases. In the case of an untextured surface with homogeneous diffuse color, a disk-like structuring set can be used to erode the specular component equally in all directions (i.e. diffuse color information can be shared in all directions) [Figure 3]. In this case, [5][1] show that equation (21) takes the form

$$\Psi_t = -\|\nabla\Psi\|, \quad (22)$$

which is essentially a grassfire flow described in Section 2.3.

For more complicated scenes having regions of distinct colors or texture, the above may cause “color bleeding” or unwanted blurring of the diffuse texture. Therefore, we need to take into account an appropriate structuring set, as well as a stopping function g to attenuate the erosion process and thus prevent “bleeding”:

$$\Psi_t = -g(\rho, \nabla\rho)(\nabla\Psi^T \mathbf{M} \nabla\Psi)^{1/2}, \quad (23)$$

where the matrix \mathbf{M} determines the shape of the structuring set for the erosion process.

For textureless surfaces having large regions of distinct uniform diffuse color, a disk structuring set ($\mathbf{M} = \mathbf{I}_{2 \times 2}$) is used, and the stopping function is defined as

$$g(\rho, \nabla\rho) = \left(\frac{1 - e^{-\rho}}{1 + e^{-\rho}} \right) \frac{e^{-(\|\nabla\rho\| - \tau)}}{1 + e^{-(\|\nabla\rho\| - \tau)}}, \quad (24)$$

where τ is a threshold on $\|\nabla\rho\|$, above which erosion is attenuated [5]. Erosion will attenuate when $\|\nabla\rho\|$ is large, as this indicates a boundary between areas of distinct colors, across which information should not be shared [5]. This process is known as *Isotropic Erosion*:

$$\Psi_t = -g(\rho, \nabla\rho)(\nabla\Psi^T \mathbf{I}_{2 \times 2} \nabla\Psi)^{1/2} = -g(\rho, \nabla\rho)\|\nabla\Psi\|. \quad (25)$$

In textured regions, the above would blur the diffuse texture. We want to instead erode ϕ *anisotropically*, and therefore use a linear structuring set aligned with the iso-contours of θ (which is independent of both specular

¹Upwind derivatives will be used for this matter in the implementation.

and shading information and thus provides uncorrupted surface color information). A local predictor is defined for the direction in which ϕ_d is constant:

$$\nabla\hat{\theta} = \begin{cases} \nabla\theta/|\nabla\theta| & \|\nabla\theta\| > 0 \\ 0 & \|\nabla\theta\| = 0. \end{cases} \quad (26)$$

The PDE becomes

$$\Psi_t = -g(\rho, \nabla\rho)(\nabla\Psi^T(\mathbf{I}_{2\times 2} - \nabla\hat{\theta}\nabla\hat{\theta}^T)\nabla\Psi)^{1/2}. \quad (27)$$

Note that when the diffuse color is constant, (i.e. $\nabla\hat{\theta} = [0 \ 0]^T$), this equation reduces to isotropic erosion (25). So this PDE can be used for either textured or non-textured surfaces. Mallick et al. go on to describe cases for more complicated scenes requiring a 2D ellipse structuring set and also discuss the extension to specular removal in videos, but for the purpose of the project we will limit our study to the above cases.

In the next section, we discuss implementation details and present our results.

4 Experimental Results

4.1 Implementation Details

The above algorithm was implemented using MATLAB 7.6. There were subtleties involved in the implementation, especially in the transformation from RGB to SUV space, as the algorithm was sensitive to the type of image data.

We have chosen to work with JPEG images downloaded from the internet, where some were taken from the authors' paper. We did not have access to the original high-quality images the authors have acquired in their laboratory under known illuminant color, but to the low-dynamic range ones available on the internet in JPEG format. In [5], it is suggested that a *standard gamma correction* of 2.2 is applied to JPEG images. Gamma correction is a nonlinear operation used to code and decode luminance values in image systems [9]. JPEG images contain *gamma-encoded* values and not linear intensities. Before carrying out the separation, a gamma correction of 2.2 was applied to the image using MATLAB's `imadjust` function. The inverse correction was then applied when recovering the diffuse RGB component. At times, these corrections may not recover the original brightness exactly, but they come close. Without the correction, the resulting diffuse images resulted in being too dark or at times corrupted.

Because we are processing JPEG images, we needed to make an assumption about the illuminant color: [5] suggested assuming the illuminant color is white when it is unknown. We assumed the same illuminant color over the entire image. With this, the transformation matrix was computed using equation (3), and applied to each pixel of the RGB image. The images ρ , θ and ϕ were computed according to equation (19).

Erosion was applied directly on the level curves of the image ϕ , where all curves simultaneously obey. Upwind derivatives (equations (7)-(10)) were used to compute spatial derivatives of Ψ , while standard central differences (see [11], equation (4.7)) were used for ρ and θ .

A first-order upwind scheme was used. With the second-order upwind scheme, resulting images were comparably similar, only *slightly* less smooth. Because results were similar, we favored the first-order scheme in order to avoid the extra computational cost of the second-order scheme.

Because intensity values are located only at each pixel in images, a spatial step size of 1 was chosen in both the x and y directions. The stopping function in equation (24) is expressed in terms of ρ and relies on a parameter τ which controls the erosion. We found that setting τ to half the maximum value taken on by ρ , with a time step between 0.1 and 0.5 effectively reduced specularities and prevented color bleeding in our results. If the threshold τ is decreased to allow less erosion, the time step should be increased accordingly, and if it is increased to allow more erosion, then the time step should be decreased to avoid instabilities.

We have chosen to visualize the behavior of one level curve at a height (or intensity) of 0.8 or 0.9. This was visualized by slicing the family of curves at a particular height using MATLAB's `contour` function, and overlaying the isocurve on the `pcolor` plot of the family of images. Using a crossing detector such as MATLAB's `edge` function with the `'zerocross'` option is another alternative in visualizing an isocurve.

The resulting RGB diffuse component was recovered by inverting equations (3) and (19) with ϕ replaced with ϕ_d . The specular component was displayed by visualizing ϕ_s .

4.2 Results

4.2.1 Isotropic Erosion

Isotropic erosion is applied to an 8-bit JPEG version of Mallick et al.'s image of a fruit basket, for which gamma is assumed to be 2.2. Because the illuminant color is unknown, it is assumed to be white [5]. Although these conditions introduce noise, the algorithm successfully recovers the diffuse and specular components as seen in Figures 4 to 6.

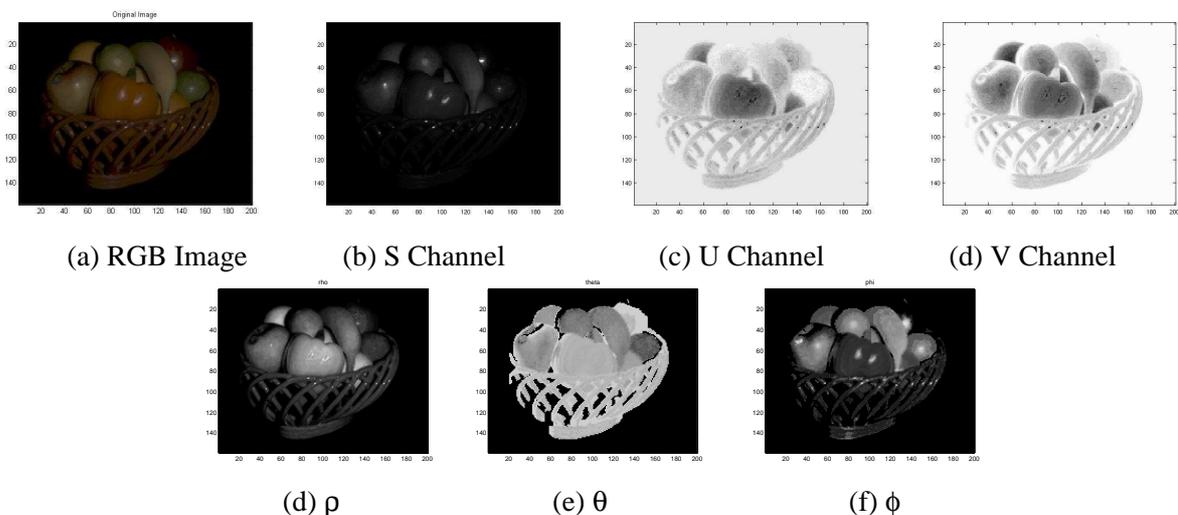


Figure 4: (a)-(c) Original RGB image along with its S, U, and V channel counterparts. After reparameterization, (d)-(e) represent the purely diffuse ρ , and θ , while ϕ contains the specular information.

In the following series of images in Figure 5, we observe the level curves of the successively eroded ϕ images. The first row displays the entire family of images, behaving simultaneously. The second row displays one level curve's behavior at intensity 0.9. We can observe it shrinking and eventually vanishing about the specularities, as expected.

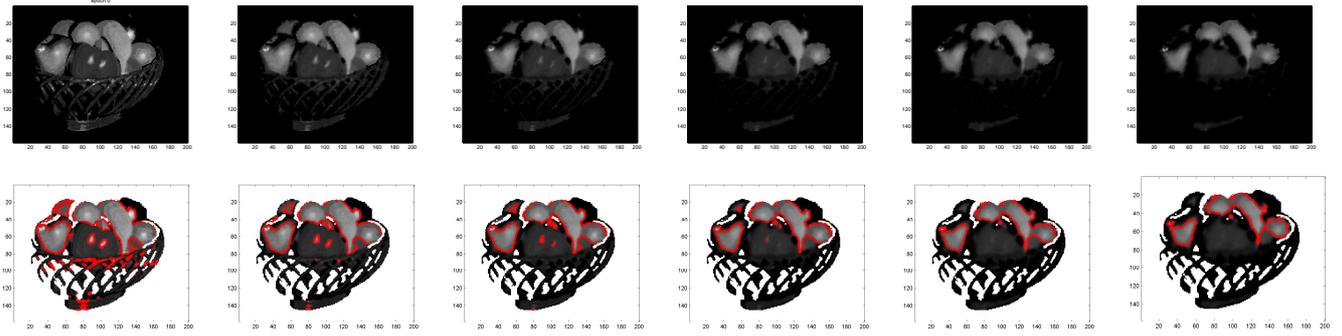
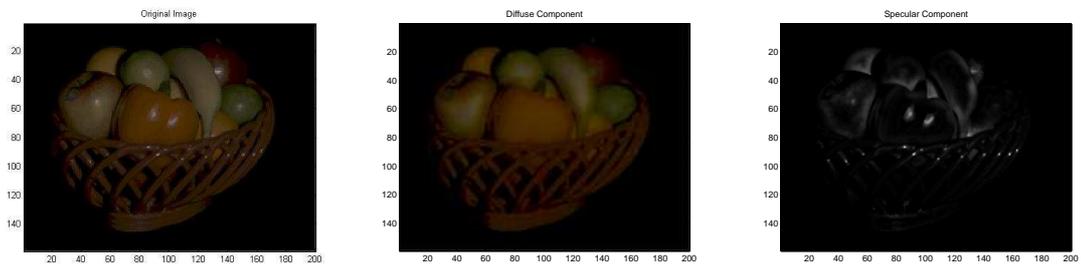


Figure 5: Level curves of the successively eroded ϕ images at epochs 0, 20, 40, 60, 80, 100, $\Delta t = 0.1$.

The resulting complete separation after recovering the RGB image - the diffuse and specular components are successfully separated:

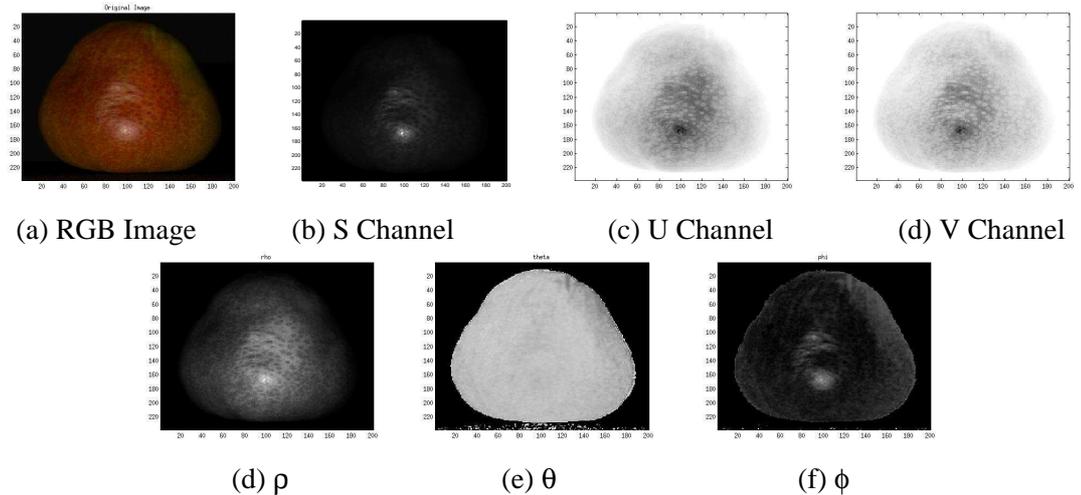


(a) Original Image [5] (b) Diffuse Component (c) Specular Component

Figure 6: Resulting Separation - Isotropic Erosion

4.2.2 Anisotropic Erosion

We apply anisotropic erosion to the image of a textured pear. If we instead eroded isotropically, the texture would become blurred or corrupted.



(a) RGB Image (b) S Channel (c) U Channel (d) V Channel

(d) ρ (e) θ (f) ϕ

Figure 7: Parameterization of Pear RGB.

In Figure 8, we observe the level curves of the successively eroded ϕ images of the textured pear. Again, the first row displays the entire family of images behaving simultaneously while the second row displays one level curve's behavior at intensity 0.8. We observe shrinking about the specularities.

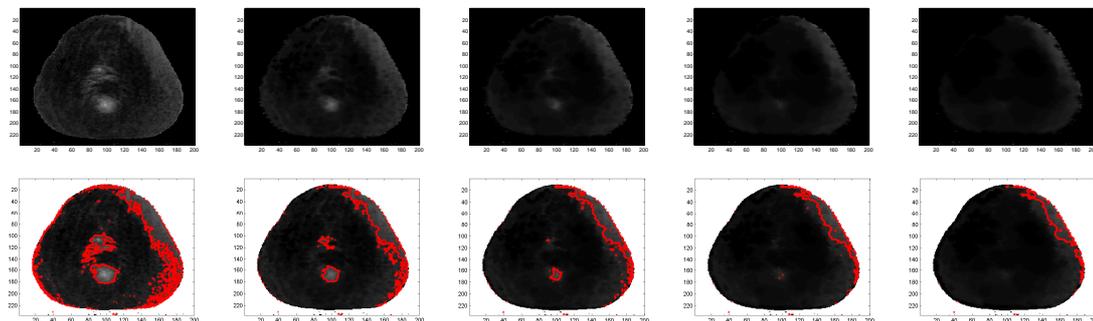
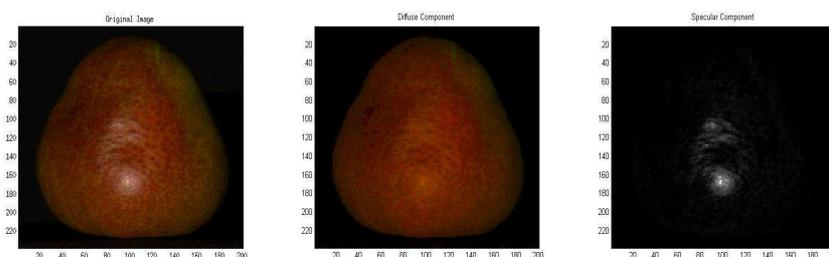


Figure 8: Level curves of the successively eroded ϕ images at epochs 0, 50, 75, 100, 150, 200, $\Delta t = 0.1$.

Despite the presence of noise due to JPEG artifacts, the algorithm manages to reduce the specularity. In the following separation, notice that the pear's diffuse texture is preserved throughout, while the specularity is picked up:



(a) Original Image [5] (b) Diffuse Component (c) Specular Component

Figure 9: Resulting Separation - Anisotropic Erosion

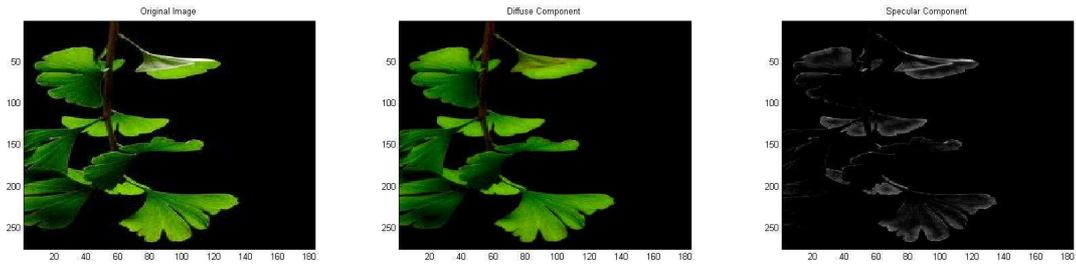
Additional Results

Anisotropic erosion (which handles both textured and untextured surfaces) with illuminant white and gamma 2.2 was applied to the following JPEG images [8][10].



(a) Original Image [8] (b) Diffuse Component (c) Specular Component

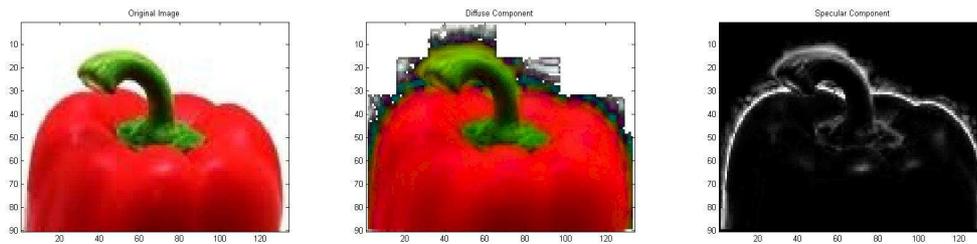
Figure 10: Recovery of diffuse and specular components in image of flowers.



(a) Original Image [10] (b) Diffuse Component (c) Specular Component
Figure 11: Recovery of diffuse and specular components in image of leaves.

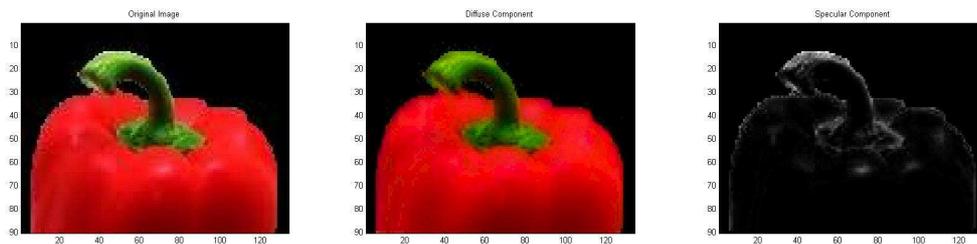
4.2.3 Encountered Algorithm Limitations

At times, the algorithm had difficulty in separating the two reflectance components. This may be due to the fact that the specular and diffuse components are not distinct, especially when the illuminant is assumed to be white and a white background is present in the image. Therefore Shafer's dichromatic model of reflectance does not hold. We notice that the authors have chosen to run their algorithm on images with black backgrounds (See Figures 3-6 in [5]), and this may be due exactly to this issue. In the image below, we suspect the algorithm will attempt to erode some of the white background around the pepper.



(a) Original Image [7] (b) Diffuse Component (c) Specular Component
Figure 12: Resulting Separation

We decided to replace the white background with a black background to test our hypothesis. The algorithm succeeds in separating the diffuse and specular components with a black background as opposed to a white one.



(a) Original Image (b) Diffuse Component (c) Specular Component
Figure 13: Resulting Separation

Videos for the above results as well as additional results can be viewed at <http://www.cs.mcgill.ca/~mscacc/Comp766/videos.html>

5 Discussion and Conclusion

Mallick et al. have presented a specular removal technique requiring no manual intervention, no segmentation of the image, or polarization filters to reduce reflections. Their algorithm can be successfully applied to a variety of images, including textured and untextured surfaces and has led to some practical results. In [5], the approach is shown to have application in dichromatic editing, where different object surface appearances could be simulated by editing and recombining the two reflection components [5].

We have successfully implemented the approach and demonstrated its effectiveness on a variety of fruit and plant images, both textured and untextured. The successful recovery of diffuse and specular components for both isotropic and anisotropic erosion was observed. In the process, we have also encountered a few limitations of the approach. Because the approach relies purely on local color and shading information, it is limited to dichromatic surfaces where the diffuse and specular components are distinct [5]. Also, it requires that the illuminant color is known or approximated beforehand. This may potentially result in suboptimal separations. For instance, if the illuminant color is unknown (and assumed to be white), the approach may suffer when applied to images with specularities and a white background. There may be difficulty distinguishing the white of the specularity from that of the surrounding diffuse background. In this case, it may be necessary to segment the image in order to restrict erosion. Another possibility is to take into consideration additional cues such as polarization [4] or local shape [5], instead of only color to further guide the erosion correctly.

To further improve the approach, it may be helpful to approximate the illuminant when it is unknown, instead of assuming it is white. For instance, by implementing a method proposed by Finlayson et al. [2] which relies on the fact that the “chromacity of most illuminants lie along a known curve in chromacity space” [2] [4]. With this, the illuminant color is recovered using the image of a single homogeneous dichromatic surface.

The approach currently recovers a monochromatic specular component. We inquire as to whether it would work with a scene that is lit using multiple light sources of possibly different unknown colors. It would be interesting to test this in a laboratory with a real scene. The authors demonstrate a natural extension of their approach to videos. Again, it appears that the illuminant is known a priori and is assumed to be constant throughout. It would be interesting to observe what would happen if the scene were subsequently lit with different light sources of different colors in a video.

The paper [5] offers little discussion regarding numerical techniques used. There is a brief mention about the use of morphological derivatives, for which we have used an upwind scheme to successfully handle the issue of shocks. There is no discussion about the ideal settings for erosion threshold (for the stopping function) versus temporal step size, and how it affects the accuracy of results. The authors mention that erosion should be attenuated when $\|\rho\|$ is large, but do not clarify the notion of “large”. Through trial and error, we found the algorithm effectively eroded specularities with small step size ($\Delta t \approx 0.1 - 0.5$) and larger erosion threshold ($\tau \approx (\max \rho)/2$). Because the color-space transformation and erosion is applied to each pixel, we found the approach computationally expensive for large images. This may not render it suitable for real-time application. The approach is however practical in applications where time is not an issue.

It is interesting how any erosion process can be treated as a curve evolution process, and we reflect on the equivalence between the two. We also reflect on the similarities between erosion and anisotropic diffusion. Anisotropic diffusion can be considered as a specular removal filtering process. It homogenizes the

texture inside edges [6], so the colors are diffused and shiny areas are removed. Unlike anisotropic erosion, it does not require any a priori information about the illuminant. However, it may not be suitable for textured surfaces, as it may blur the texture inside regions which is otherwise preserved by anisotropic erosion.

In all, Mallick et al. have presented a clever technique for specular removal in images requiring no manual intervention. Through our experimental results, the approach proved to be robust under the presence of noise. We did not know the true illuminant color and had access only to the JPEG versions of the images. Nonetheless, our results were comparable to those from the paper. The additional results also demonstrated successful separation of the two reflectance components. In the future, it would be interesting to try the approach on a real scene by acquiring an image using a high-quality camera, and measuring the illuminant using a spectrometer.

Acknowledgements

I would like to thank Professor Kaleem Siddiqi for his generous guidance and insightful teachings throughout the development of this project. I would also like to thank Professor Michael Langer for his supportive feedback and discussions on photography.

References

- [1] R. Brockett and P. Maragos. Evolution equations for continuous scale morphology. *IEEE trans. on Sig. Proc.*, 1994.
- [2] G. Finlayson and G. Schaefer. Constrained dichromatic colour constancy. *In Proc. European Conf. on Computer Vision*, 2001.
- [3] R. W. Fleming, A. Torralba, and E. H. Adelson. Specular reflection and the perception of shape. *Journal of Vision*, 2004.
- [4] S. P. Mallick, T. Zickler, D. J. Kriegman, and P. N. Belhumeur. Beyond lambert: Reconstructing specular surfaces using color. *In Proc. Int. Conf. on Computer Vision*, 2001.
- [5] S. P. Mallick, T. Zickler, D. J. Kriegman, and P. N. Belhumeur. Specularity removal in images and videos: A pde approach. *Proc. European Conference on Computer Vision*, 2006.
- [6] S. Le Moan, A. Mansouri, T. Sliwa, M. P. Patricio, Y. Voisin, and J. Y. Hardeberg. Convex objects recognition and classification using spectral and morphological descriptors. *CGIV 2010 – Fourth European Conference on Colour in Graphics, Imaging, and MCS/10 Vision 12th International Symposium on Multispectral Colour Science*, 2010.
- [7] Pepper JPEG photograph. 123rf Stock Photography, 2011.
- [8] Layout Sparks Photographs. Army brat black background flowers JPEG photograph, 2011.
- [9] C. A. Poynton. *Digital Video and HDTV: Algorithms and Interfaces*. Morgan Kaufmann, 2003.
- [10] N. Prachai. Ginkgo biloba leaves JPEG photograph. <http://www.student.chula.ac.th>, 2011.
- [11] J.A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
- [12] K. Siddiqi. Shape analysis in computer vision, lecture notes. McGill University, 2011.