Highlight Removal by Illumination-Constrained Inpainting

Ping Tan[†] Stephen Lin[†] Long Quan[‡] Heung-Yeung Shum[†]
[†]Microsoft Research, Asia*
[‡]Hong Kong University of Science and Technology

Abstract

We present a single-image highlight removal method that incorporates illumination-based constraints into image inpainting. Unlike occluded image regions filled by traditional inpainting, highlight pixels contain some useful information for guiding the inpainting process. Constraints provided by observed pixel colors, highlight color analysis and illumination color uniformity are employed in our method to improve estimation of the underlying diffuse color. The inclusion of these illumination constraints allows for better recovery of shading and textures by inpainting. Experimental results are given to demonstrate the performance of our method.

1 Introduction

Highlights in images have long been disruptive to computer vision algorithms. They appear as surface features, when in fact they are artifacts caused by lighting that change in position and appearance under different viewing conditions. This can lead to problems such as stereo mismatching, false segmentations and recognition errors. Because of the undesirable effects of highlights on image analysis, there have been several previous works that focus on highlight removal.

1.1 Related work

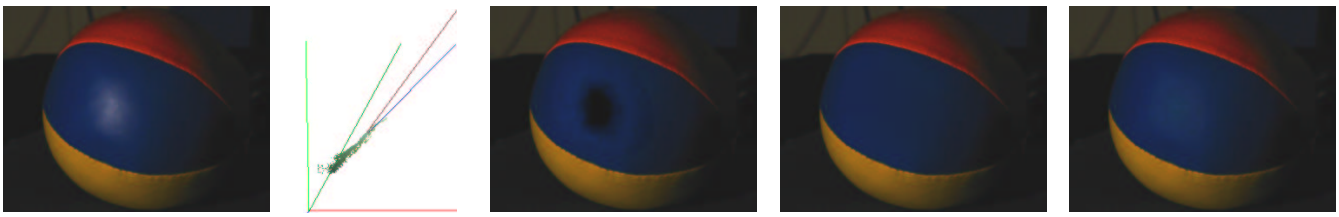
Previous methods for highlight removal have been based on color or polarization information. To make the problem more tractable, some techniques utilize data from a set of images. Wolff [16] removed highlights by taking advantage of differences in polarization between diffuse reflections and highlights. Sato and Ikeuchi [14] analyzed color data in an image sequence taken under a moving light source to compute highlight components. Nayar et al. [9] utilize both color and polarization to constrain estimates of the reflection components. While these methods have produced good results, the need for multiple images or polarization, which is sensed from three filtered images, significantly narrows their applicability.

A single-image approach for highlight removal was introduced by Klinker et al. [6]. They observed from Shafer's dichromatic reflection model [15] that in a color histogram, diffuse and highlight pixels form linear clusters in a T-shape, where the highlight cluster extends along the illuminant color direction. After fitting vectors to the highlight and diffuse clusters, they project highlight colors along the illumination color direction onto the diffuse vector to compute the diffuse colors. The highlight cluster, however, is often skewed due to surface roughness and imaging geometry [10], so an estimate of illuminant color by vector fitting can be inaccurate. Fig. 1(a-c) displays how this skew can affect highlight removal on a somewhat rough surface. Novak and Shafer [10] describe how surface roughness and imaging geometry can be derived from measurements of the histogram shape to determine this skew, but color distributions are generally too muddled to obtain reliable measurements because of image noise and multiple diffuse colors.

When RGB color is intensity-normalized and represented in a 2D chromaticity space, the skew problem disappears, and Lee [7] presents an approach that estimates illuminant chromaticity from two or more differently colored surfaces that exhibit highlights. In this method, highlight points on a uniform-colored surface form a line in chromaticity space, and the intersection of two such lines from different-colored surfaces gives the illuminant chromaticity. Obtaining in this manner an estimate precise enough for effective highlight removal, though, is generally difficult because of sensor noise. Moreover, an estimate cannot be made when highlights lie on only a single surface color.

Estimation of illuminant color is also addressed in the area of color constancy [4, 13, 5], which attempts to remove the effect of illumination color from an image. The illumination estimates by these methods, however, are much too coarse for highlight removal, and often require assumptions such as a wide distribution of surface colors or the absence of highlights.

*Correspondence email: stevelin@microsoft.com



(a) (b) (c) (d) (e)

Figure 1. A comparison of highlight removal methods. (a) Original image. (b) RGB color histogram of the inpainting region, where the dark green line is fit to the diffuse cluster, the dark red line to the highlight cluster, and the dark blue line is the actual illuminant color. (c) Result of highlight removal based on vector fitting. (d) Result of TV inpainting. (e) Result of illumination-constrained inpainting.

1.2 Our approach

An area with some relation to highlight removal is image inpainting. Inpainting is a technique for filling in an image region by propagating information from the region boundaries. This approach has demonstrated much success in applications such as restoring scratched photographs, removing objects from images, and noise reduction [1, 2, 12].

In our work, we introduce an inpainting technique that is designed for highlight removal. Previous inpainting methods address the problem of filling in an occluded region where no information about this region is known. For highlights, however, some partial information for determining the underlying diffuse reflection is often available. One source of data is the observed image color of a pixel in a highlight region, since it arises from a sum of diffuse and specular reflection components. Second, a highlight region is typically formed from a single illumination color. Third, some information on the illuminant color can be derived from chromaticity analysis. With these highlight properties, our proposed algorithm constrains the inpainting of highlight regions to produce removal results that exceed traditional inpainting and previous single-image techniques.

For highlight removal, inpainting methods benefit from illumination constraints because diffuse shading within a highlight region can be more accurately recovered. Fig. 1(d-e) exemplifies this difference for a highlight on a ball. Since traditional inpainting propagates boundary values, the interior of the highlight is assigned shading intensities that interpolate those on the highlight boundary, giving the highlight area a geometrically flat appearance. Illumination-based constraints can lead inpainting to more accurate shadings as shown in Fig. 1(e), where the diffuse shading intensities within the highlight should exceed those on the highlight border. Another advantage of these constraints is that surface textures obscured by highlights are better recovered, instead of being eliminated or distorted by traditional inpainting.

For color-based techniques, an advantage of employing inpainting is that it reasonably resolves the illumination color ambiguity in a single image. Previous methods encounter problems in obtaining an accurate illumination color, but inpainting contributes an additional smoothness constraint that yields good visible results. Furthermore, previous color-based methods require each highlight pixel to be grouped with corresponding non-highlight pixels that have the same diffuse color. Such groupings are difficult to form, especially when considering color-blended pixels that occur at texture boundaries and the similarity of texture colors when mixed with intense highlight components. Some texture colors within a highlight may not even have corresponding diffuse pixels outside the highlight. In our proposed method, such groupings are not required for highlight removal. An additional benefit of inpainting is that for saturated pixels where color measurements are incorrect, traditional inpainting can nevertheless produce fair estimates of their diffuse components.

Our algorithm takes as input a single image with user-circled highlight regions. Since textures in a single image can have an appearance identical to highlights, user interaction is needed to handle this ambiguity. For previous works on detection of highlights in multiple-image input, we refer the reader to [16, 8]. Since each highlight region is processed independently, they can each have a different illumination color, which can result from different types of light sources and interreflections.

In the remainder of the paper, we first detail the illumination constraints and describe highlight removal for the simple case of uniform-colored surfaces in Section 2. This formulation is extended in Section 3 to our general highlight removal algorithm that can also be applied to textured surfaces. Section 4 presents techniques that can facilitate the highlight removal process, followed by experimental results in Section 5 and a discussion in Section 6.

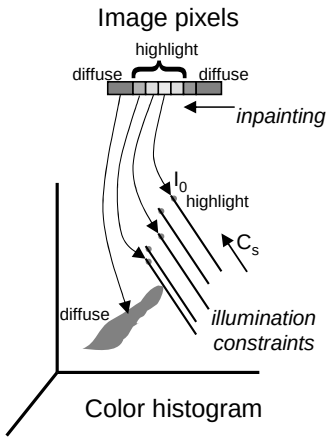


Figure 2. Inpainting subject to illumination constraints. Inpainted diffuse colors in the image space must satisfy line constraints illustrated in the color histogram.

2 Illumination-constrained inpainting

The observed RGB radiance $\boldsymbol{I}_o$ of a highlight point is formed from a sum of diffuse reflection $\boldsymbol{I}_d$ and specular reflection $\boldsymbol{I}_s$, where the diffuse color $\boldsymbol{C}_d$ is the intrinsic color of the surface, the specular color $\boldsymbol{C}_s$ is that of the illumination, and the observed color is denoted as $\boldsymbol{C}_o$. This physical relationship at a point (x, y) is represented in the dichromatic reflection model as

$$\boldsymbol{I}_o(x,y) = \boldsymbol{I}_d(x,y) + \boldsymbol{I}_s(x,y) = \alpha_d \boldsymbol{C}_d(x,y) + \alpha_s \boldsymbol{C}_s(x,y), \quad (1)$$

where α_d, α_s are coefficients that depend on imaging geometry and surface material properties. In our formulations, we normalize RGB colors by their intensity $||\boldsymbol{I}|| = R + G + B$, to give chromaticity values $r = R/||\boldsymbol{I}||$, $g = G/||\boldsymbol{I}||$, $b = B/||\boldsymbol{I}||$. When collapsing the 3D color space down to a 2D rg chromaticity space, the dichromatic model of (1) is transformed to

$$\boldsymbol{c}_o = \alpha \boldsymbol{c}_d + (1-\alpha)\boldsymbol{c}_s, \quad (2)$$

where $\alpha = \alpha_d/(\alpha_d + \alpha_s)$, and $\boldsymbol{c}_o, \boldsymbol{c}_d, \boldsymbol{c}_s$ are the rg chromaticity vectors of $\boldsymbol{C}_o, \boldsymbol{C}_d, \boldsymbol{C}_s$, respectively.

In highlight removal, the goal is to replace the observed radiance $\boldsymbol{I}_o(x, y)$ with its diffuse component $\boldsymbol{I}_d(x, y)$ for each highlight pixel (x, y). The diffuse component can be constrained by a few illumination-based quantities that can be determined from the image. From the dichromatic reflection model, it can be seen that $\boldsymbol{I}_d$ is related to the observed highlight value $\boldsymbol{I}_o$, and is also constrained by the illumination color $\boldsymbol{C}_s$. Additionally, we make the assumption that the illumination color is uniform for a given highlight, meaning that $\boldsymbol{C}_s(x, y)$ is constant over the highlight region. With (1), these highlight properties together form the following constraint on the diffuse component:

$$\boldsymbol{I}_d(x,y) = \boldsymbol{I}_o(x,y) - \alpha_s(x,y)\boldsymbol{C}_s. \quad (3)$$

This equation alone does not fully constrain the diffuse component, because $\alpha_s(x, y)$ is an unknown quantity that cannot directly be estimated from a single image. To resolve this problem, our method favors smoothness of the diffuse component by employing a total variation (TV) form of inpainting [12] that incorporates the illumination-based constraint of (3).

Let us first assume that the highlight lies on a surface that has a single diffuse color. The inpainting solution is found by minimizing the following energy function over the highlight region Ω with respect to α_s:

$$E_{IC} = \int_{(x,y)\in\Omega} \begin{array}{l} \{\ \gamma[\nabla r(x,y)]^2 + \gamma[\nabla g(x,y)]^2 \\ \quad + ||\nabla \boldsymbol{I}_d(x,y)||^2 \}\, dxdy \end{array} \quad (4)$$

$$\text{where } \boldsymbol{I}_d(x,y) = \boldsymbol{I}_o(x,y) - \alpha_s(x,y)\boldsymbol{C}_s.$$

Instead of inpainting diffuse colors in terms of R, G, B, we divide the color data into chromaticity r, g and intensity $||\boldsymbol{I}_d||$ so that chromaticity smoothness can be emphasized, using the weight γ. The illuminant color is solved by determining the value of $\boldsymbol{C}_s$ that results in the smallest E_{IC}:

$$\boldsymbol{C}_S = \arg\min_{C_s} \widehat{E}_{IC}$$

where $\widehat{E}_{IC}$ is the minimized energy for a given illuminant color $\boldsymbol{C}_s$.

Fig. 2 illustrates within a color histogram the effect of illumination-constrained inpainting. The constraint of (3) restricts the diffuse components of highlight pixels to lie on corresponding parallel lines whose orientation is given by the illumination color $\boldsymbol{C}_s$ and whose positions depend on the observed radiances $\boldsymbol{I}_o$. The smoothness favored by inpainting determines the estimated diffuse component locations on these lines. The distances that separate the parallel lines represent shading differences among the pixels. Some image noise is retained because of the line constraints, giving the inpainted region a more natural appearance with respect to the rest of the image. Estimating the illuminant color is equivalent to finding the parallel line direction that gives the smoothest change of diffuse component.

3 Highlight removal algorithm

The inpainting formulation of (4) is suitable only for image areas without texture, because of its emphasis on smoothness in the diffuse component. In this section, we extend the highlight removal algorithm to address the more

general case of surfaces that may have more than one color. For textured surfaces, illumination-constrained inpainting should not be performed across texture edges, for which a more appropriate inpainting solution should be employed.

To prevent smoothing of diffuse reflection across texture edges by illumination-constrained inpainting, we utilize the idea of edge stopping in anisotropic diffusion methods [11, 3], which has been used for image denoising without diminishing edge strength. Energy functions in anisotropic diffusion follow the form

$$E = \int_{\Omega} s(||\nabla \boldsymbol{I}||)||\nabla \boldsymbol{I}||d\Omega$$

where the stopping function s goes to zero for larger gradients. This effectively halts inpainting across edges. Our general inpainting method includes a stopping function defined as

$$s(||\nabla \boldsymbol{N}(x,y)||) = \begin{cases} 0 & \text{if } ||\nabla \boldsymbol{N}(x,y)|| > t \\ 1 & \text{if } ||\nabla \boldsymbol{N}(x,y)|| \leq t \end{cases} \quad (5)$$

where $\boldsymbol{N}(x,y)$ denotes the normal direction of the plane defined by the origin and the illumination constraint line of (x,y), which passes through $I_o(x,y)$. This plane is equivalent to the dichromatic plane spanned by $\boldsymbol{C}_d(x,y)$ and $\boldsymbol{C}_s$, as seen from (1) and (3). All points in a non-textured area should lie on a single dichromatic plane, so a difference in $\boldsymbol{N}$ among neighboring points indicates a texture edge. This quantity is used instead of image gradients, because image gradients include not only differences in texture color but also color changes that result from variations in specular component.

Since illumination-constrained inpainting is not valid across texture edges, our method inpaints specular components across edges instead, since changes in specular reflection are generally smoother than diffuse reflection over texture edges. The specular component $\boldsymbol{I}_s(x,y)$ can be expressed as $\boldsymbol{I}_o(x,y) - \boldsymbol{I}_d(x,y)$, and the energy function for specular inpainting over region Ω_S can be written as

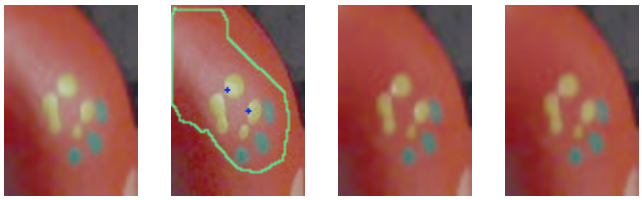
$$E_S = \int_{(x,y)\in\Omega_S} ||\nabla(\boldsymbol{I}_o(x,y) - \boldsymbol{I}_d(x,y))||dxdy. \quad (6)$$

Incorporation of the stopping function and specular inpainting into (4) yields our general inpainting equation:

$$E = \int_{(x,y)\in\Omega} \begin{array}{l} s(||\nabla \boldsymbol{N}(x,y)||)\,\{\gamma[\nabla r(x,y)]^2 + \gamma[\nabla g(x,y)]^2 \\ +[\nabla||\boldsymbol{I}_d(x,y)||]^2\} + [1 - s(||\nabla \boldsymbol{N}(x,y)||)] \\ ||\nabla(\boldsymbol{I}_o(x,y) - \boldsymbol{I}_d(x,y))||dxdy \end{array} \quad (7)$$

$$\text{where } \boldsymbol{I}_d(x,y) = \boldsymbol{I}_o(x,y) - \alpha_s(x,y)\boldsymbol{C}_s,$$

As in (4), $\boldsymbol{C}_s$ is determined as the value that gives the minimal E. The stopping function is used to switch between illumination-constrained inpainting and specular inpainting, depending on the presence of texture edges.



(a) (b) (c) (d)

Figure 3. Steps in highlight removal. (a) Original image; (b) Saturated pixels indicated by blue, and user-specified highlight area outlined in light green; (c) After our inpainting procedure; (d) Final result after TV inpainting of saturated pixels.

With this inpainting function, our algorithm proceeds as illustrated in Fig. 3. From the original image (a), the user first outlines the highlight region as shown in (b). This boundary may cross multiple texture regions. Saturated pixels, denoted by blue, have clipped colors that do not represent actual reflection colors, so they should not be processed by (7). This saturation set is determined by thresholding and then is dilated by one pixel to account for blooming effects in the CCD array. The non-saturated pixels are then inpainted by (7) as shown in (c). Since the saturated pixels contain no useful information about their diffuse component and are equivalent to being occluded, we simply color them by standard TV inpainting as displayed in (d), noting that other traditional inpainting methods may be used in its place. For this example, the yellow and green textures within the highlight do not have matching diffuse areas outside the highlight region, a scenario that is not addressed in other color-based highlight removal methods.

4 Implementation considerations

The highlight removal algorithm can be made more computationally efficient by incorporating the techniques described in this section. For estimation of the illumination color, it is expensive to calculate full inpainting solutions for multiple lighting colors, so we propose a more rapid method based on color analysis and partial inpainting. Additionally, we present a scheme that reduces inpainting computation by determining a good initial solution.

4.1 Illumination color estimation

The color distribution of the highlight region provides information that can be used to narrow the range of possible illumination colors. According to the dichromatic model of (2), highlight points on a single-colored surface form a linear cluster in the chromaticity space as illustrated in

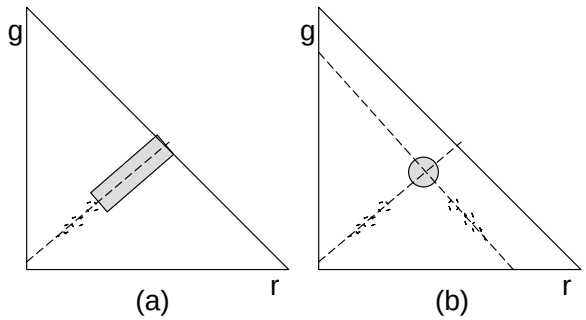


Figure 4. Possible range of illumination colors, from a highlight on: (a) a single-colored surface, (b) a two-colored surface.

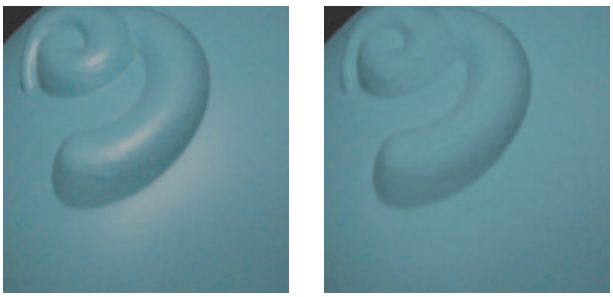


Figure 5. Highlight removal for a single-colored surface. Left: original image, Right: our result.

Fig. 4(a). This cluster lies somewhere between $\boldsymbol{c}_d$ and $\boldsymbol{c}_s$, so $\boldsymbol{c}_s$ must lie along the cluster direction beyond the chromaticity of the brightest pixel, which lies closest to $\boldsymbol{c}_s$. Because of imprecisions caused by image noise, we restrict $\boldsymbol{c}_s$ to a neighborhood around the cluster direction instead of just a single line segment. When a highlight covers a textured region, the possible range of the illuminant color is tightened, since $\boldsymbol{c}_s$ is constrained by at least two clusters as exhibited in Fig. 4(b).

The illuminant range is further narrowed to a single point by finding the value that yields the minimal inpainting energy. To reduce the amount of computation, our implementation inpaints only a subset of the highlight region that has a uniform surface color, using the basic illumination-constrained inpainting of (4). To determine such a subregion, our algorithm takes a color distribution cluster used to constrain the illumination color range, and from its corresponding image pixels the largest connected component that borders the highlight boundary is found.

4.2 Initial inpainting estimate

The efficiency of the inpainting process also depends on the number of optimization iterations needed to reach the minimal energy. This minimum can be attained more rapidly when the initial inpainting region more closely resembles the final inpainting solution. To obtain a good initial estimate of the inpainting solution, our implementation first labels each highlight pixel with its stopping function value from (5), then connected components are formed for pixels labelled 1 and for pixels labelled 0. Let us consider the highlight boundary pixels to be "processed" and the pixels within the highlight as initially "unprocessed". For components labelled 1 that are adjacent to the boundary of processed pixels, its pixels are recursively processed inwards from the boundary by computing the point on the illumination constraint line (3) with the minimum SSD to the values of its previously processed neighbors. Next, components labelled 0 and adjacent to processed pixels are then filled using the specular inpainting function in (6). These two steps are iterated until all the non-saturated highlight pixels have been processed.

This speedup technique can be used both for illumination color estimation and for the general inpainting function. After determining the initial estimate, the final inpainting solution is computed by gradient descent.

5 Experimental results

Results of our constrained inpainting method are presented for a few real images with different types of texture. The R,G,B sensor responses were calibrated for the camera, and the algorithm parameters were set to $\gamma = 100$ and $t = 0.1$ for all images.

The first image, shown in Fig. 5, is of a single-color surface containing some shape variation. The second image in Fig. 6 is of an object with simple large-scale texture. Fig. 7 shows a third image of a wooden tabletop with detailed texture. Wood presents a difficult situation for highlight removal, because its diffuse reflection does not change very smoothly. This complication leads to an illumination color estimate that is slightly inaccurate, resulting in modest artifacts in the removal result.

The performance of our method is reasonable for single images that may contain texture. Multiple-image methods start with significantly more information to constrain the diffuse reflection components, and are limited in applicability. Results for prior single-image methods have been presented only for smooth, textureless surfaces. Extending these methods to rougher and textured surfaces would require segmentation of the surface into different diffuse colors, the existence of purely diffuse counterparts for each highlight pixel, and a different approach to illumination color estimation. For our single-image algorithm that handles textures, the quality of the results is at a level sufficient

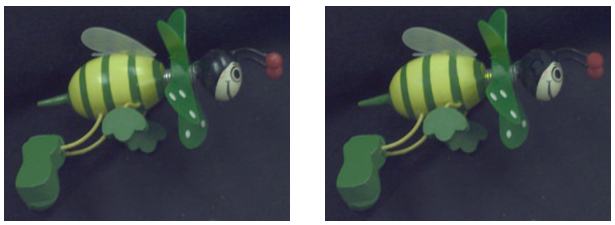


Figure 6. Highlight removal for an object with large-scale texture. Left: original image, Right: our result.

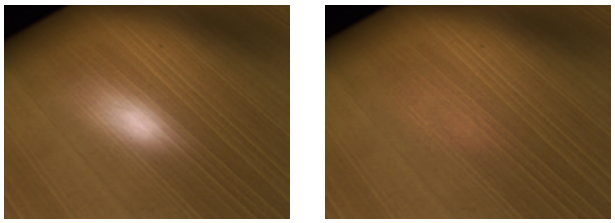


Figure 7. Highlight removal for a wood tabletop with detailed texture. Left: original image, Right: our result.

to benefit other vision algorithms as a preprocessing step.

6 Discussion

When the colors of the illumination and the diffuse component are similar, the highlight removal algorithm becomes similar to TV inpainting. In this instance, the illumination constraint lines are collinear with the diffuse cluster, so highlight points will be mapped in a way that smoothly interpolates the boundary colors. When a surface texture consists of a single color with different brightness levels, such as dark green and light green, the intensity differences that compose the texture are maintained by our method in the same manner as shading changes.

The illumination constraint is based on the assumption that the illumination color is uniform throughout the highlight. This is generally true because most highlights arise from a single light source. However, for highly specular surfaces such as metals, many highlights arise from interreflections, and it is not uncommon for multiple interreflection colors to form a contiguous highlight. The user can roughly deal with this scenario by dividing highlight regions into areas of uniform interreflection color.

By introducing illumination-based constraints into an inpainting process, our method takes fuller advantage of image information than previous single-image highlight removal techniques. With an emphasis on smoothness and adherence to physical reflectance behavior, our approach can produce reasonable results for challenging cases. Like other inpainting methods, our technique requires significant computation, even with the speedup techniques presented in Section 4, so in future work we plan to develop a faster implementation based on these ideas.

References

[1] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Computer Vision and Pattern Recog.*, pages I:355–362, 2001.

[2] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles. Image inpainting. In *Computer Graphics, SIGGRAPH 2000*, pages 417–424, July 2000.

[3] M. J. Black, G. Sapiro, and D. H. Marimont. Robust anisotropic diffusion. *IEEE Trans. Image Processing*, 7, 1998.

[4] D. H. Brainard and W. T. Freeman. Bayesian color constancy. *Journal of the Optical Society of America A*, 14, 1997.

[5] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23, 2001.

[6] 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.

[7] H.-C. Lee. Method for computing the scene-illuminant chromaticity from specular highlights. *Journal of the Optical Society of America A*, 3, 1986.

[8] S. W. Lee and R. Bajcsy. Detection of specularity using color and multiple views. *Image and Vision Computing*, 10:643–653, 1992.

[9] S. K. Nayar, X. Fang, and T. E. Boult. Removal of specularities using color and polarization. In *Proc. IEEE Computer Vision and Pattern Recog.*, pages 583–590, 1993.

[10] C. L. Novak and S. A. Shafer. Anatomy of a color histogram. In *Proc. IEEE Computer Vision and Pattern Recog.*, pages 599–605, 1992.

[11] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 1990.

[12] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[13] G. Sapiro. Color and illuminant voting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21, 1999.

[14] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *Journal of the Optical Society of America A*, 11, 1994.

[15] S. Shafer. Using color to separate reflection components. *Color Research and Applications*, 10:210–218, 1985.

[16] L. B. Wolff. Using polarization to separate reflection components. In *Proc. IEEE Computer Vision and Pattern Recog.*, pages 363–369, 1989.