Recovering Shading from Color Images

Brian V. Funt, Mark S. Drew, and Michael Brockington
School of Computing Science
Simon Fraser University
Vancouver, B.C.
Canada V5A 1S6
(604) 291-3126
funt@cs.sfu.ca
fax: (604) 291-3045

Abstract

Existing shape-from-shading algorithms assume constant reflectance across the shaded surface. Multi-colored surfaces are excluded because both shading and reflectance affect the measured image intensity. Given a standard RGB color image, we describe a method of eliminating the reflectance effects in order to calculate a shading field that depends only on the relative positions of the illuminant and surface. Of course, shading recovery is closely tied to lightness recovery and our method follows from the work of Land [10, 9], Horn [7] and Blake [1]. In the luminance image, R+G+B, shading and reflectance are confounded. Reflectance changes are located and removed from the luminance image by thresholding the gradient of its logarithm at locations of abrupt chromaticity change. Thresholding can lead to gradient fields which are not conservative (do not have zero curl everywhere and are not integrable) and therefore do not represent realizable shading fields. By applying a new curl-correction technique at the thresholded locations, the thresholding is improved and the gradient fields are forced to be conservative. The resulting Poisson equation is solved directly by the Fourier transform method. Experiments with real images are presented.

1 Introduction

Color presents a problem for shape-from-shading methods because it affects the apparent "shading" and hence the apparent shape as well. Color variation violates one of the main assumptions of existing shape-from-shading work, namely, that of constant albedo. Pentland [11] and Zheng [16] give examples of the errors that arise in violating this assumption.

We address the problem of recovering (up to an overall multiplicative constant) the intrinsic shading field underlying a color image of a multi-colored scene. In the ideal case, the recovered shading field would be a graylevel image of the scene as it *would have appeared* had all the objects in the scene been gray. We take as our definition of *shading* that it is the sum of all the processes affecting the image intensity other than changes in surface color (hue or brightness). Shading arises from changes in surface orientation and illumination intensity.

It is quite surprising how well some shape-from-shading (SFS) algorithms work when they are applied directly to graylevel images of multi-colored scenes [16]. This is encouraging since it means that shading recovery may not need to be perfect for successful shape recovery. Nonetheless, the more accurate the shading the more accurate we can expect the shape to be.

Consider the image of Fig. 1(a) which is a black and white photograph of a color image of a cereal box. The lettering is yellow on a deep blue background. Applying Pentland's [11] remarkably simple linear SFS method to the graylevel luminance version (i.e. R+G+B) of this color image generates the depth map in Fig. 1(b). Although the image violates Pentland's assumptions somewhat in that the light source was not very distant and the algorithm is known to do poorly on flat surfaces, it is clear that the yellow lettering creates serious flaws in the recovered shape. Note also that the errors are not confined to the immediate area of the lettering.

The goal of our algorithm is to create a shaded intensity image in which the effects of varying color have been removed in order to improve the performance of SFS algorithms such as Pentland's. Similar to previous work on lightness, the idea is to separate intensity changes caused by change in color from those caused by change in shape on the basis that color-based intensity changes tend to be very abrupt. Most lightness work, however, has considered only planar "mondrian" scenes and has processed the color channels separately. In lightness computations, the slowly varying intensity changes are removed from each color channel by thresholding on the derivative of the logarithm of the intensity in that channel. We instead remove intensity gradients from the logarithm of the luminance image by thresholding whenever the chromaticity changes abruptly. Both the luminance and the chromaticity combine information from all three color channels.

Many examples of lightness computation in the literature [13, 14, 1, 8, 4] use only synthetic images. A notable exception is in Horn [7] in which he discusses the problem of thresholding and the need for appropriate sensor spacing. He conducts experiments on a few very simple real images. Choosing an appropriate threshold is notoriously difficult and the current problem is no exception. By placing the emphasis on shading rather than lightness, however, fewer locations are thresholded because it is the large gradients that are set to zero, not the small ones. When a portion of a large gradient change remains after thresholding due to the threshold being too high, the curl of the remaining luminance gradient becomes non-zero. Locations of non-zero curl are easily identified and the threshold modified by a technique called "curl-correction."

In what follows, we first analyze the case of one-dimensional images before proceeding to the two-dimensional case. Then we elaborate on curl-correction and present results of tests with real images.

2 One-dimensional Case

2.1 Color Images with Shading

Let us consider as a starting point the surface described by the one-dimensional depth map shown in Fig. 2(a). If this surface has Lambertian reflectance and is illuminated by a point source from an angle of 135°, (i.e., from the upper left), the resulting intensity distribution will be as shown in Fig. 2(b). So far there is no color variation, so all the intensity variation is due to shading.

If instead, the surface has regions of different color, each described by its own color triple (R,G,B) in the absence of shading, (see Fig. 2(c)), then in a color image of the surface the RGB values will be these original color triples modulated by the shading field, as shown in Fig. 2(d). The combined effect of color edges and shading edges leads to discontinuities in the observed RGB values at image locations corresponding to both types. Fig. 2(d) has both kinds of edges—there are color discontinuities where there are no shape discontinuities, and there are also shape discontinuities without accompanying color ones.

To differentiate between the two kinds of edges, we note that if we form the analog the chromaticity [15] in RGB space, i.e.,

$$
\begin{aligned}
r &= R/(R+G+B) \\
g &= G/(R+G+B)
\end{aligned}
$$

then r and g are independent of the shading (cf. [12, 6, 2]) as can be seen in Fig. 2(e). This, of course, must be the case because rg–chromaticity is simply a way of normalizing the magnitude of the RGB triple. Both r and g are fixed throughout a region of constant color. So long as we can assume that that color edges never coincide with shape edges, rg–chromaticity will distinguish between them.

2.2 Shading recovery

In a sense, the recovery of shading from color images is the obverse of the recovery of lightness from graylevel images [7]. In the case of lightness, it is the sharp intensity changes that are taken to represent reflectance changes and are retained, while the small intensity changes are factored out on the basis that the illumination varies smoothly; whereas, in the case of shading it is the large reflectance changes that are factored out and the small ones retained. A significant difference, however, is that the reflectance changes are identified, not by sharp intensity changes, but by sharp chromaticity changes. Small chromaticity changes are assumed to be caused by changes in the spectrum of the illuminant and are retained as part of the shading.

We begin by following the usual lightness recovery strategy [7, 1], but to do so we first need to transform a color image into graylevel luminance image by forming

$$I' = R + G + B \ .$$

Under the assumption that the luminance is described well as the product of a shading component S and a color component C, the two components are separated by taking logarithms:

$$I(x) = log I'(x) \equiv log S'(x) + log C'(x)$$

Differentiating, thresholding way all components of the derivative coincident with large chromaticity changes, and integrating yields the logarithm of the shading.

Chromaticity changes (dr, dg) are determined from the derivative of the chromaticity (r, g) where the threshold function locates pixels with high $|dr|$ or $|dg|$. The threshold function is defined as $T(x) = 1$ at pixels where (dr, dg) is small and $T(x) = 0$ where it is large. T will be mostly 1; whereas, for lightness T will be mostly 0.

Applying T to the derivative of the luminance image eliminates $C = log C'$, so $S = log S'$ can be recovered by integrating the thresholded intensity derivative. In other words,

$$\begin{aligned} dS &= T(dI) \\ S &= \int dS \\ S' &= \exp S \end{aligned}$$

It is easy to see from Figs. 2(a)–(e) that this algorithm will recover the shading properly for the case of perfect step edges and the correct result is in fact obtained by the algorithm as shown in Fig. 2(f).

2.3 Integration by Fourier Expansion

A fast, direct method of integrating the thresholded derivative of the shading field in the two-dimensional case is to apply Fourier transforms. While not efficient in the one-dimensional case it is easy to understand how the method works. Firstly, if the discrete Fourier transform of dS is $F(dS)$, the effect of differentiation is given by

$$F(dS) = 2\pi i u \, F(S)$$

where the frequency variable is u. This expression no longer holds exactly, however, when the derivative is calculated by convolution with a finite-differences mask. For the case of convolution by a derivative mask, after both the mask and the image are extended with zeroes to avoid wraparound error [5], Frankot and Chellappa [3] show how to write the Fourier transform of the derivative operation in terms of u. Call this transform H. H is effectively the Fourier transform of the derivative mask, and integration of dS simply involves dividing by H and taking the inverse transform:

$$F(S) = F(dS)/H$$

This division will not be carried out at $u = 0$, of course, so that integration by this method does not recover the DC term representing the unknown constant of integration.

3 Two-dimensional Shading Recovery

To generalize the method to real, two-dimensional images two main problems need to be addressed: how to properly deal with non-step edges and, since in two-dimensions the gradient replaces the derivative, how to integrate easily the gradient image?

As in the one-dimensional case, the procedure is to determine a threshold image T from the chromaticity image, apply the threshold to the derivative of the logarithm of the luminance image I and integrate the result to obtain the shading field S. The threshold function T comes from the chromaticity itself, not its log, and for two-dimensional images is based on the gradient of the chromaticity vector field

$$\|\nabla(r,g)\| = \sqrt{\nabla r \cdot \nabla r + \nabla g \cdot \nabla g}\,.$$

For this type of problem, the boundary conditions must be treated appropriately as Blake [1] points out. He addresses lightness recovery and uses a threshold based on the value of the log-intensity gradient and applies it to the log-intensity gradient. In the present case we apply a threshold based on the chromaticity gradient to the log-luminance gradient; nevertheless, Blake's results can still be applied.

Blake proves that given a threshold image T, the (log of) lightness L can be recovered from the (log of) intensity I by inverting the Laplacian of I provided correct boundary conditions are used and provided the *thresholded intensity gradient forms an irrotational field.*

Specifically, he shows that

$$\nabla L = T\nabla I \implies \begin{cases} \nabla^2 L = \nabla \cdot T\,\nabla I \\ \vec{n} \cdot \nabla L = \vec{n} \cdot T\nabla I \\ \quad \text{on boundary} \end{cases}$$

However, the converse holds only if the field $T\nabla I$ has zero curl:

$$\left.\begin{array}{r}\nabla^2 L \;=\; \nabla\cdot T\,\nabla I\\ \vec{n}\cdot\nabla L \;=\; \vec{n}\cdot T\nabla I\\ \text{on boundary}\\ \\ \nabla\times(T\nabla I)=0\end{array}\right\}\Longrightarrow\;\nabla L \;=\; T\nabla I$$

Blake argues that in theory $T\nabla I$ will have zero curl and thus forms a conservative field. Furthermore, he points out that if this condition is violated in practice, the best solution is robust in the sense that it minimizes a least-squares energy integral for L. The demand that the curl be approximately zero is important because it amounts to the condition that the recovered lightness (or shading field in our case) be integrable from its gradient.

The fact that we use a threshold T that is not derived from I itself, but is instead derived from (r, g), does not make any difference in the proof of Blake's theorem. In fact, any T will do—the formal statement of the theorem follows through no matter how T is chosen. The crucial point is that while the theorem holds for continuous images and step edges, in practice the curl may not be zero because of edges that are not perfect steps. With a non-step edge, thresholding may zero out only half the effective edge, say, in which case $T\nabla I$ will not be conservative.

Another situation that can affect integrability is when some chromaticity edges are slightly stronger than others so that some of the weaker edges are missed by the threshold. For example, consider the case of a square of a different color in the middle of an otherwise uniform image. If for some reason the horizontal edges are slightly stronger than the vertical ones, so that only the horizontal ones are thresholded away, then the curl—necessarily zero everywhere in the input gradient image—will become non-zero at the corners of the square. The non-zero curl indicates that the resulting integrated image will not make sense and we cannot hope to recover the correct, flat shading image from this thresholded gradient.

What should be done to enforce integrability? Blake further differentiates the thresholded gradient calculating its divergence, which results in the Laplacian of the lightness field. In essence, Blake's method enforces integrability by mapping the two components of the gradient image back to a single image L. Differentiating L will clearly result in an irrotational gradient field.

In the context of shape-from-shading, Frankot and Chellappa [3] enforce integrability of the gradient image (p, q) by projecting in the Fourier domain onto an integrable set. This turns out to be equivalent to taking another derivative of (p, q) and assuming the resulting sum equals the Laplacian of z [13]. For the lightness problem, then, integrating by forming the Laplacian and inverting is a method of enforcing integrability of $T\nabla I$. The most efficient method for inverting the Laplacian is integration in the Fourier domain, as set forward in [13].

While these methods of projecting $T\nabla I$ onto an integrable vector field generate the optimal result in the sense that it is closest to the non-integrable original, in the case of the thresholded shading gradient "closest" is not necessarily best. For example, consider the luminance edge associate with the color change shown in Fig. 3(a) and its gradient image Fig. 3(b) (using one dimension for illustrative purposes). Since the chromaticity edge is not a perfect step, we can expect thresholding to eliminate only part of the edge as shown in Fig. 3(c). The projection method of integration by forming a Laplacian and inverting uses the integrable gradient that is best in the sense that it is closest to Fig. 3(c). Fig. 3(d) shows the result after integration. The problem is that while the gradient of Fig.3 (d) may be curl-free, a lot of the edge that should have been screened out remains. We would prefer a method that enforces integrability while also removing more of the unwanted edge.

3.1 Curl-Correction

If the thresholding step had succeeded in zeroing the entire gradient at the edge, then the resulting image would have had zero curl. To accomplish the dual goals of creating an integrable field and of eliminating the edge, we propose thresholding out the gradient wherever the curl is non-zero. This must be done iteratively, since further thresholding may itself generate more pixels with non-zero curl. Iteration continues until the maximum curl has become acceptably small. Since the portion of the edge that was missed by the initial thresholding created the curl problem, the thresholded region will expand until the whole edge has been removed.

An alternative curl-correction scheme is to distribute the contributions of the x and y partial derivatives of the gradient that make the curl non-zero among the pixel and its neighboring pixels so that the result has zero curl. As an example of this type of scheme one can determine which part of the curl, the x derivative of the y-gradient or the y derivative of the x-gradient, is larger in absolute value. Then the larger part is made equal to the other by adjusting the larger gradient value contributing to the curl. Tests with this method did not show that it worked any better than the simpler scheme of simply zeroing the gradient. Although it might work better in some other context, all results reported in this paper simply zero the gradient.

3.2 Boundary Conditions

Blake imposes Neumann boundary conditions in which the derivative at the boundary is specified. The process of integration by Fourier expansion is simplified slightly if, instead, Dirichelet boundary conditions are used in which the values at the border are fixed. Surrounding the image with a black border accomplishes this.

In the case of lightness, the Dirichelet conditions will not work because the intensity variation removed via thresholding does not balance out from one edge of the image to another. For shading recovery, however, as long as the color changes are contained within the image, what is thresholded does balance across the image. Color changes are generally completely contained within the image as, for example, with colored letters on a colored background. For convenience, our current implementation uses Dirichelet conditions, but could straightforwardly be changed to Neumann conditions if necessary.

3.3 Algorithm

To summarize the above discussion, the shading-recovery algorithm is as follows:

1. Find color edges.

   (a) Form chromaticity images $r = R/(R+G+B)$, $g = G/(R+G+B)$

   (b) Smooth chromaticity images by convolution with a Gaussian.

   (c) Form gradient images ∇r, ∇g.

   (d) Form threshold image: $T = 0$ if $\|\nabla(r,g)\|$ is larger than a percentage of the maximum in the initial image. Else T=1.

2. Make log-of-luminance image, $I = \log(R+G+B)$.

3. Make thresholded x- and y-gradient images of I. Denote these images by P and Q since they are analogous to the gradient (p,q) in the SFS problem.

   (a) Surround image I with a border of zeroes.

   (b) Form $P = \partial I/\partial x$, $Q = \partial I/\partial y$.

   (c) Apply T to both P and Q yielding thresholded gradient images $T(P)$ and $T(Q)$.

4. Curl-correction

   (a) Integrability requires $\partial/\partial_y(T(P)) - \partial/\partial_x(T(Q)) = 0$. For the input log-intensity image I, this is precisely true up to numerical accuracy.

   (b) If the maximum curl is not sufficiently small, then set to zero all locations of non-zero curl. The locations to be zeroed also include the immediately surrounding neighborhood. For example, when a 3×1 mask $(-0.5, 0, 0.5)$ is used for the partial derivatives, then the horizontal and vertical neighbors of the 3×3 surrounding square should also be made zero.

   (c) Repeat curl-correction until the maximum curl value decreases sufficiently—for example, until it reaches 50% of the original maximum curl.

5. Combine the thresholded gradient components by taking another derivative. This gives the Laplacian of S: $\nabla^2 S = \partial/\partial_x(T(P)) + \partial/\partial_y(T(Q))$.

6. Solve this Poisson equation by the method of Fourier expansion [13]. Exponentiate to find the actual shading image S'.

3.4 Test image

Fig. 4 illustrates several of the above steps on the synthetic image shown in part (a). The actual image is in color but since the colors are not crucial to the argument, it is reproduced only in black and white. It consists of a colored square (128 by 128 pixels) on a colored background. The color of the square and the background are each uniform, but there is a luminance gradient in all three bands increasing from left to right. To generate non-step edges, the image has been smoothed by convolution with a Gaussian. Part (b) shows a graph of the intensity along the middle row. The red and green chromaticity images (scaled for display) are shown in parts (c) and (d). The chromaticities are flat within the square and the surround. The thresholded log-luminance image, before curl-correction, is shown in part (e). Curl-correction was repeated until the maximum value of the curl anywhere in the image decreased to 50% of the maximum value of what it was originally. The number of iterations was 13, which is high compared to non-synthetic images (see below). The final, curl-corrected threshold image is shown in part (f). Part (g) shows the (non-log) recovered shading image. The intensity gradient across this final shading image is plotted in part (h).

4 Results on Real Images

The luminance image derived from the cereal box, color image of Fig. 1 is shown in Fig. 5 (a). The corresponding chromaticity images r, g (scaled) are Figs. 5 (b,c). Applying the gradient operator to these chromaticity images and thresholding at 40% of the gradient yields the initial threshold image in Fig. 5 (d) Reducing the maximum curl to 60% of its original maximum via curl correction generates the extended threshold image, Fig. 5 (e). The number of curl-correction iterations required was 5. The recovered shading image is Fig. 5 (f) with the difference between figures (a) and (f) shown in Fig. 5 (g).

In order to compare the algorithm's performance with "ground truth," we also considered the image of Fig. 6 (a), which was created by Lambertian shading of a laser range-finder depth map of a plaster bust of Mozart. Fig. 6 (b) overlays this shading field with color by multiplication with the colors measured in the cereal box image. Thus, both the shading and the color edges come from natural objects, but in a controlled fashion, so the result is a synthetic image constructed from real shapes and colors including noise. The image shown is actually the luminance image derived from the color image. To take into account the color and not shape of the box, the colors were extracted from the chromaticity images Fig. 5 (a,b), with the b image formed as $1 - r - g$, rather than using the RGB directly. So that the intensity image would not simply equal the original shading image, the r, g, b components were multiplied by unequal amounts.

The *chromaticity* images of the input color image are thus precisely those of Fig. 5 (a,b) (because Fig. 6 (a) contains no pixels that are exactly zero). The initial chromaticity-derived threshold function, with a threshold level of 30%, is shown in Fig. 6 (c). Requiring the maximum curl value to be reduced to 60% of its original value took a single iteration—lowering the initial threshold further, from 40% in Fig. 5 to 30% substantially speeds up the curl-correction step.

Curl-correction extends the threshold function as in Fig. 6 (d). Applying the algorithm to the luminance image Fig. 6 (b), results in Fig. 6 (e) Comparing to Fig. 6 (a), the shading is recovered well in that the difference between figures (a) and (e) is negligible.

5 Assumptions and Limitations

Stated explicitly the assumptions and limitations of the algorithm are:

- Color edges must not coincide with shading edges.
- All color edges must involve a change of hue/chromaticity, not just brightness (e.g. not orange to dark orange).
- Surfaces are Lambertian reflectors. Strong specularities will be mistaken for reflectance changes, while weak specular components will be attributed to shading.
- The spectral power distribution of the illumination should be constant, but of course its intensity can vary. Gradual changes will be attributed to the shading to the extent that they affect the luminance image. Abrupt changes in intensity are allowed and will be correctly attributed to shading because they will not cause an abrupt chromaticity change. This is unlike retinex algorithms, which will be fooled by sharp intensity changes because they treat each color channel separately.
- The shading is recovered up to an overall multiplicative scaling constant.

6 Conclusions

Color creates problems for shape-from-shading algorithms which assume that surfaces are of constant albedo. We have implemented and tested on real images an algorithm that recovers shading fields from color images which are equivalent to what they would have been had the surfaces been all one color. It uses chromaticity to separate the surface reflectance from surface shading and involves thresholding the gradient of the logarithm of the image luminance. The resulting Poisson equation is inverted by the direct, Fourier transform method.

7 Acknowledgements

M.S. Drew is indebted to the Center for Systems Science at Simon Fraser University for partial support; B.V. Funt thanks both the CSS and the Natural Sciences and Engineering Research

Council of Canada for their support. The laser range data for the bust of Mozart is due to Fridtjof Stein of the USC Institute for Robotics and Intelligent Systems.

8 Figure Captions

- Figure 1.
  (a) Black and white photograph of a color image of a cereal box—blue background with yellow lettering.
  (b) Recovered depth image using shape from shading algorithm of [11].

- Figure 2. One-dimensional case:
  (a) Initial depth map.
  (b) Shading field for (gray) Lambertian surface illuminated by a point source from upper left.
  (c) Same surface with colored stripes (red—solid, green—dotted, blue—dashed).
  (d) Colors in image (c) multiplied by shading field of image (b).
  (e) Chromaticities formed from observed camera values.
  (f) Shading field recovered by algorithm.

- Figure 3. Thresholding:
  (a) Smooth step.
  (b) Derivative.
  (c) Thresholded derivative.
  (d) Integration of (c)—in 2 dimensions would be integration of gradient images under integrable projection.

- Figure 4.
  (a) Black and white photograph of synthetic color image: flat square and surround with illumination gradient, smoothed with Gaussian.
  (b) Intensity across the image.
  (c) Red chromaticity image.
  (d) (2nd row) Green chromaticity image.
  (e) Initial threshold.
  (f) Threshold after curl-correction.
  (g) (3rd row) Output shading image.
  (h) Intensity across output image.

- Figure 5. Cereal box:
  (a) Intensity image.
  (b) Red Chromaticity image.
  (c) Green Chromaticity image.
  (d) (2nd row) Initial threshold.
  (e) Threshold after curl-correction.

(f) Output shading image.
(g) (3rd row) Difference between (a) and (f).

- Figure 6. Mozart bust overlaid with color:
  (a) Intensity image.
  (b) Red chromaticity image.
  (c) Green chromaticity image.
  (d) (2nd row) Initial threshold.
  (e) Threshold after region-growing.
  (f) Output shading image.
  (g) (3rd row) Difference between (a) and (f).

References

[1] A. Blake. Boundary conditions for lightness computation in Mondrian world. *Computer Vision, Graphics, and Image Processing*, 32:314–327, 1985.

[2] P. T. Eliason, L. A. Soderblom, and P. S. Chavez Jr. Extraction of topographic and spectral albedo information from multispectral images. *Photogrammetric Engineering and Remote Sensing*, 48:1571–1579, 1981.

[3] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:439–451, 1988.

[4] B.V. Funt and M.S. Drew. Color constancy computation in near–Mondrian scenes using a finite dimensional linear model. In *Computer Vision and Pattern Recognition Proceedings*, pages 544–549. IEEE Computer Society, June 1988.

[5] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison/Wesley, 2nd edition, 1987.

[6] G. Healey. Using color for geometry-insensitive segmentation. *J. Opt. Soc. Am. A*, 6:920–937, 1989.

[7] B. K. P. Horn. Determining lightness from an image. *Computer Vision, Graphics, and Image Processing*, 3:277–299, 1974.

[8] A. Hurlbert. Formal connections between lightness algorithms. *J. Opt. Soc. Am. A*, 3:1684–1692, 1986.

[9] E.H. Land. Recent advances in retinex theory. *Vision Res.*, 26:7–21, 1986.

[10] E.H. Land and J.J. McCann. Lightness and retinex theory. *J. Opt. Soc. Amer.*, 61:1–11, 1971.

[11] A. P. Pentland. Linear shape from shading. *Int. J. Comput. Vision*, 4:153–162, 1990.

[12] T. Poggio. Mit progress in understanding images. In *DARPA Image Understanding Workshop*, pages 56–74, 1989.

[13] T. Simchony, R. Chellappa, and M. Shao. Direct analytical methods for solving Poisson equations in computer vision problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:435–445, 1990.

[14] D. Terzopoulos. Uimage analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:129–139, 1986.

[15] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. Wiley, New York, 2nd edition, 1982.

[16] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:680–702, 1991.