# Color Space Analysis of Mutual Illumination

Brian V. Funt and Mark S. Drew

*Abstract*— **Mutual illumination occurs when light reflected from one surface impinges on a second one. The resulting additional illumination incident on the second surface affects both the color and intensity of the light reflected from it. As a consequence, the image of a surface in the presence of mutual illumination differs from what it otherwise would have been in the absence of mutual illumination. Unaccounted for mutual illumination can easily confuse methods that rely on intensity or color such as shape-from-shading or color-based object recognition. In this correspondence, we introduce an algorithm that removes mutual illumination effects from images. The domain is that of previously-segmented images of convex surfaces of uniform color and diffuse reflectance where for each surface the interreflection occurs mainly from one other surface and can be accurately accounted for within a one-bounce model. The algorithm is based on a singular value decomposition of the colors coming from each surface. Geometrical information about where on the surface the colors emanate from is not required. The RGB triples from a single convex surface experiencing interreflection fall in a plane; intersecting the planes generated from two interreflecting surfaces results in a unique interreflection color. Each pixel can then be factored into its interreflection and no-interreflection components so that a complete no-interreflection image is produced.**

*Index Terms*—**Mutual illumination, interreflection, color vision, color histogram, computer vision, shape from shading.**

## I. Introduction

The interreflection of light in color images can substantially alter colors and intensity levels of pixels corresponding to points receiving light, not just from a light source, but by reflection from other surfaces in the scene as well. This interreflected light is often called mutual illumination and its effects are well known [13], [6], [7], [20], [9].

The mutual illumination effect can be slight when there is a lot of illumination directly from a light source [3], [9], [4]; however, in enclosed spaces such as a room, the infinite exchange of radiation between the walls provides a large fraction of the light in the room [14]. In computer graphics, it has been necessary to model these interreflections using the radiosity method [11] in order to obtain realistic images.

For computer vision, the one-bounce model of mutual illumination proposed in [3], [9], [4] provides a good approximation in many situations. The one-bounce model simply makes use of the fact that the intensity of the interreflection diminishes substantially with each bounce.

Even when mutual illumination is small as a percentage of the total illumination, it may profoundly alter the results of computer vision algorithms [6], [7]. In part, this is because mutual illumination changes image intensities in a consistent, nonrandom way so that errors it introduces into, say, a shape-from-intensity solution accumulate rather than cancel out. For example, imagine the region near a concave fold in a surface of a uniformly colored object. Intensities of points near the fold are all increased (but by differing amounts) due to interreflection and this increase will lead most shape-from-intensity algorithms into ascribing an incorrect set of direction cosines to the surface there. Nayar [20] observes that the concavity will always appear as being shallower than it actually is.

Nayar overcomes this problem by using an iterative scheme in which first an estimate of the shape is calculated from the intensity data; then with this shape estimate as input, the radiosity method is applied to estimate a corrected, no-interreflection image intensity distribution. These steps are iterated until convergence. The method as it stands is flawed, however, because it does not take into account the color of the interreflected light.

Near a concave edge formed by two surfaces of different color, interreflection causes a significant component of the light reflected from either side to have a color which differs from that of either of the original surface colors. The spectrum of the interreflected light is formed as the product of the illumination spectrum and the spectra of the two surface reflectance functions. In fact, the surface reflectance functions multiply together as many times as the light is interreflected, with the total mutual illumination contribution equaling the sum of an infinite number of interreflections. The resulting spectrum is then filtered by the sensor response functions to obtain an RGB triple representing the color.

Instead of calculating the energy leaving each surface facet in terms of the complete spectrum, or in three bands, Nayar *et al.* [20] assume that the reflectance can be summed up by a single number, the albedo. The difficulty with this assumption can be seen by considering a concave edge that is red on one side and blue on the other. In the extreme case, the spectral reflectance function of the red will be zero at the blue end of the spectrum and vice-versa. The product of the red and blue spectra will therefore be zero everywhere. As a result, while mutual illumination is to be expected based on the edge geometry, none in fact occurs; and the radiosity step of Nayar's algorithm, which describes surface reflectance by a single albedo value, will over-estimate the interreflection and lead to an incorrect calculation of the surface shape.[1] The problem persists even for less extreme cases, such as an edge which is red on both sides, because the product of the red spectrum with itself does not equal itself.[2]

Color and mutual illumination are inextricably interrelated. Unless we are to restrict ourselves to surfaces that are varying shades of gray, color will be fundamental to the analysis of mutual illumination. Conversely, mutual illumination can be important to the analysis of color. In previous work, [3], [9], [4] we analyzed mutual illumination to obtain color constancy (i.e., illumination-independent color descriptors). In fact, it is possible to reconstruct good approximations to the *entire* surface spectral reflectance functions for two surfaces participating in mutual illumination.

This is *strong* color constancy in that the entire spectrum is established [12]. Weaker versions of color constancy have also been advocated. For example, Forsyth [5] has suggested that RGB values obtained under a canonical illuminant may be viewed as color-constant descriptors.

In the present correspondence, we describe an algorithm that produces an image which is free of mutual illumination effects. Since these effects include induced color shifts, their elimination is also related to the general problem of determining stable color descriptors,

Manuscript received April 17, 1991; revised June 1, 1992. M. S. Drew was supported in part by the Centre for Systems Science, Simon Fraser University. B. V. Funt was supported by the CSS and the Natural Sciences and Engineering Research Council of Canada. Recommended for acceptance by Associate Editor S. Shafer.

The authors are with the School of Computing Science, Simon Fraser University, Vancouver, BC, Canada V5A 1S6.

IEEE Log Number 9211838.

[1] In response to this criticism raised in [8], Nayar [19] has recently extended his original algorithm to narrowband RGB sensors.

[2] This last observation is due to Michael Swain, personal communication, Dec. 1990.

but in a weaker sense than either of the above definitions of color constancy. Our algorithm finds the color of each object independent of mutual illumination, where "color" means the RGB values produced under the light source in use, not a canonical source.

The domain is restricted to interreflections between two matte, convex surfaces, each of uniform color, with illumination that can vary spatially in its intensity but not in its spectral composition. The algorithm generates two images. One is a no-interreflection image in which the colors and shading on each surface are as they would have been if the alternate surface had not been in the scene. The second is an image of the mutual illumination distribution, the magnitude of which represents a type of "configuration factor" between the objects [23]. Here, we use the term "interreflection factor" to describe this field because it includes the effect of multiple interreflections within a one-bounce model; whereas, a true configuration factor represents only a single bounce. This interreflection factor encapsulates a good deal of information about the scene geometry including orientation, occlusion, and shape. Potentially, it could be used to extract some of these features.

A one-bounce model of mutual illumination predicts that the RGB triples measured from each of an interreflecting pair of surfaces should consist of linear combinations of a *no-bounce* color and a *one-bounce* color. A no-bounce color means the only-reflected-once (i.e. no interreflections involved) color that results from the reflection of the source illuminant off one of the surfaces. A one-bounce color results from the reflection, by the second surface, of the first surface's no-bounce color. Although in general there are two no-bounce colors, one for each surface, there is only one one-bounce color because of symmetry.

To the extent that the one-bounce model holds and the domain restrictions apply (i.e., no self-reflection or specularities), the linear combination of two components means that there will be a plane in RGB color space containing all the RGB triples of a surface. A reliable method for determining the plane is to carry out a singular value decomposition (SVD) of the RGB values [10]. A good plane for the SVD to find will only exist, however, if there is mutual illumination present and the hue of the one-bounce color differs sufficiently from the no-bounce color. If either of these two conditions is violated, the RGB values will fall in a one- rather than two-dimensional subspace as discussed further in Section II.

In this correpsondence, we assume that the the image has been segmented, but this restriction could be weakened by using a Hough transform to identify candidate planes in Hough space. Other approaches are discussed in [1], [22].

We would like to find the RGB vectors of the one-bounce and no-bounce colors. The two principal SVD eigenvectors define the plane, but unfortunately they do not represent these colors. Nonetheless, in principle once a surface's color-space plane is known, bands within which these colors must lie can be determined by the method originated by Lawton and Sylvestre [17] and further elucidated by Kawata *et al.* [15]. This method treats each surface independently.

We can, however, make use of the fact that the two color planes are related via their common one-bounce color. For example, the spectrum of light from a red surface reflected secondarily by a blue surface is the same as that of light from a blue surface reflected secondarily by a red one; hence, we can expect both the color planes of the two surfaces to contain the *same* one-bounce, red-blue, interreflection color. Intersecting these planes will yield the one-bounce color.

The idea of intersecting color-space planes has been used before, but in the analysis of specularities. In the analysis of interreflections, we will follow the SVD analysis used by Tominaga and Wandell [25] in their treatment of surface and body reflectance from multiple surfaces under a simple model of specularity. They make use of the fact that it is possible to derive the spectrum of the illuminant, within the dichromatic reflection model [21], [16], by crossing the planes of color signals (entire spectra) from each of several surfaces [26], [27] (and see also [24]). Then the SVD analysis for each surface can be combined, via the Lawton and Sylvestre method, to determine a band of spectra within which the body reflection function must lie.

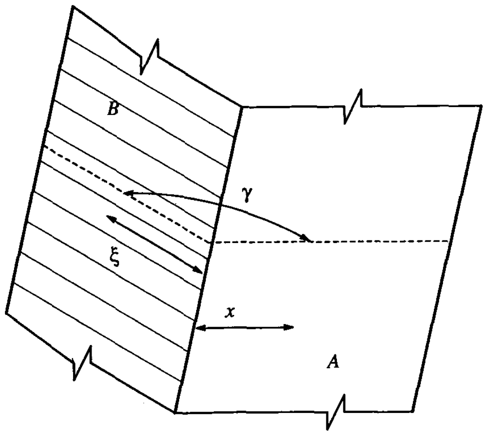


Fig. 1. Two semi-infinite planes that join with opening angle $\gamma$. In Section III, we use $\gamma = 45°$, creating an edge that appears much more closed.

Section II sets out the color space analysis of mutual illumination: how the color planes are found and how the one-bounce color is determined. The no-bounce color lies within a band defined by the one-bounce color along with the constraint that all colors must have entirely nonnegative components. Given the no-bounce color of a surface, the shading and the geometric interreflection factor are easily calculated at each pixel.

We make use of 3-component vectors, not entire spectra, so that *ab initio* it is not clear how well the SVD method will work. Therefore it is useful to start with a simple, simulated image. Since the physics of a planar edge under diffuse illumination is perfectly understood [23], in Section III, we examine the simulated image of a two-color, semi-infinite planar edge (cf. [3]). We find that the Lawton and Sylvestre method runs into problems because the vectors have so few components (i.e., 3) and present a modified method for this case.

Section IV details the result of applying the algorithm to a real image as well as a method for automatically evaluating the confidence with which the method has identified the correct surface colors, shading fields, one-bounce color, and interreflection factor fields.

## II. Mutual Illumination and Color Space Analysis

### A. One-Bounce Model of Mutual Illumination

Consider a semi-infinite edge formed by two differently colored materials, as shown in Fig. 1. Mutual illumination can occur between nontouching surfaces as well, as shall be shown in Section IV, but the edge in Fig. 1 will be studied in Section III and might be considered fundamental to the problem.

Suppose the edge is illuminated with spectral power distribution $E(\lambda)$. Denote the surface spectral reflectance function for surface $A$ by $S^A(\lambda)$ and that for surface $B$ by $S^B(\lambda)$. Let us assume that all surfaces are Lambertian, or at least that we exclude specularities, and call the RGB values resulting from the direct illumination impinging on both surfaces, plus the results of all interreflection, $\rho_k, k \cdots 3$.

Naturally, the values of $\rho_k$ vary over the surfaces $A$ and $B$ and especially while moving away from the edge.

To exactly model the $\rho_k$ requires evaluating an integral equation representing the infinite-bounce interreflection at each wavelength and then applying the color filters. The physics of the situation is given in [23]. To simplify the analysis, we adopt a one-bounce approximation of the radiant exchange between the two surfaces [3], [9], [4].

The one-bounce model describes the color signal $C^A(x, \lambda)$ from a pixel at location $x$ on surface $A$ as

$$\begin{aligned} C^A(x, \lambda) &\simeq \alpha^A(x) E(\lambda) S^A(\lambda) \\ &+ \beta^{BA}(x) S^A(\lambda) E(\lambda) S^B(\lambda). \end{aligned} \quad (1)$$

A similar equation can be written for surface $B$.

Equation (1) states that $C$ is comprised of the light reflected from surface $A$ with no interreflection—the no-bounce color signal—plus the light reflected from surface $B$ and then reflected again from surface $A$. Of course, this once-reflected light will reflect again off $B$, but we truncate an infinite series of bounces and call the second term the one-bounce contribution.

Due to changes in surface orientation and position, the intensity of the direct illumination $E$ will generally vary spatially, although its spectrum is assumed to remain constant. The factor $\alpha^A(x)$ represents this intensity variation or *shading* at point $x$.

For reflection from surface $B$ to surface $A$, the proportionality factor $\beta^{BA}(x)$ represents the relative contribution of interreflection to the color signal. It encapsulates into a single number all the possible circumstances that can affect the magnitude of the one-bounce contribution, including the local surface orientation at $x$, the overall shape of both surfaces, the shading on both surfaces, and the possibility that some points on surface $B$ are occluded from the vantage point of pixel $x$. For image synthesis, calculating $\beta^{BA}(x)$ poses a computationally intensive problem (cf. [11]); for image analysis, on the other hand, *deriving* the proportionality factors $\alpha(x)$ and $\beta(x)$ given an RGB image can be done quite efficiently.

To derive the colors and proportionality factors, we must consider the nature of the color space. On sensors of sensitivity $R_k(\lambda)$, a color signal $C$ creates sensor response $\vec{\rho}$ according to

$$\rho_k(x) = \int C(x, \lambda) R_k(\lambda) d\lambda. \quad (2)$$

Application of (2) to the $C(\lambda)$ in equation (1) leads to vectors $\vec{\rho}$ on surfaces $A$ and $B$ given by

$$\begin{aligned} \rho_k^A(x) &\simeq \alpha^A(x) \rho_k^{A\_nobounce} + \beta^{BA}(x) \rho_k^{onebounce} \\ \rho_k^B(x) &\simeq \alpha^B(x) \rho_k^{B\_nobounce} + \beta^{AB}(x) \rho_k^{onebounce}, \end{aligned} \quad (3)$$

where

$$\rho_k^{A\_nobounce} = \int E(\lambda) S^A(\lambda) R_k(\lambda) d\lambda$$

$$\rho_k^{B\_nobounce} = \int E(\lambda) S^B(\lambda) R_k(\lambda) d\lambda$$

and

$$\rho_k^{onebounce} = \int E(\lambda) S^A(\lambda) S^B(\lambda) R_k(\lambda) d\lambda.$$

All $\vec{\rho}$ from surface $A$ are formed by a linear combination of $\vec{\rho}^{A\_nobounce}$ and $\vec{\rho}^{\,onebounce}$. Similarly, on surface $B$ the $\vec{\rho}$ are made up of $\vec{\rho}^{B\_nobounce}$ and the *same* $\vec{\rho}^{\,onebounce}$. To the extent that (1) correctly approximates the color signal from surface $A$, all the $\vec{\rho}$ arising from surface $A$ will lie in a plane in $\vec{\rho}$-space. In Brill's terminology [2], the $\vec{\rho}$ form a rank-2 image field.

## B. SVD Color Space Analysis

Since the two color-space planes formed by all the $\rho_k^A$ and all the $\rho_k^B$ contain the common color $\vec{\rho}^{\,onebounce}$ it lies in their intersection and so is easy to compute. Singular value decomposition (SVD) analysis provides a reliable method of finding the planes. The SVD operates on the matrix $\boldsymbol{R}$ of all the data from surface $A$. $\boldsymbol{R}$ is $N \times 3$, where $N$ is the number of pixels. The SVD analysis generates additional matrices that decompose $\boldsymbol{R}$ as

$$\boldsymbol{R} = \boldsymbol{U \Lambda V}$$

where $\boldsymbol{U}$ is an $N \times 3$ matrix, $\boldsymbol{\Lambda}$ is a $3 \times 3$ diagonal matrix of eigenvalues, and $\boldsymbol{V}$ is a $3 \times 3$ matrix of eigenvectors in three-dimensional RGB color space. If the third eigenvalue in $\boldsymbol{A}$ is much smaller than the first two, then most of the pixels lie in the plane defined by the principal component vectors $\vec{v}_1, \vec{v}_2$ formed from the columns of $\boldsymbol{V}$.

## C. One-Bounce Color from Crossing Planes

By analogy to the similar situation that arises with specularities [26], [27], [24], we proceed to cross the color space planes generated from the two surfaces in order to find the one-bounce color $\vec{\rho}^{\,onebounce}$. In [24], the planes were embedded in high-dimensional spaces of vectors representing entire spectra, so the SVD method was used in calculating their intersection. In the present situation, however, the planes are simply embedded in three-space, so that the crossed-planes color $\vec{\rho}^{\,onebounce}$ is unique and can be found by algebra.

If the first two principal component vectors on surfaces $A$ and $B$ are $\vec{v}_1^A, \vec{v}_2^A, \vec{v}_1^B, \vec{v}_2^B$ then solving the following vector equation in the three unknowns $u_2, w_1, w_2$ yields the intersection vector:

$$\vec{v}_1^A + u_2 \vec{v}_2^A - w_1 \vec{v}_1^B - w_2 \vec{v}_2^B = \vec{0}. \quad (4)$$

Therefore,

$$\vec{\rho}^{\,onebounce} = \vec{v}_1^A + u_2 \vec{v}_2^A. \quad (5)$$

For surface $A$, the one-bounce model predicts that any $\vec{\rho}$ is composed of the above color and the no-bounce color for surface $A$, $\vec{\rho}^{A\_nobounce}$. As will be addressed in the next section, the problem remains of how to find the no-bounce color of each side.

## D. Quarter-Circle Analysis

Within the one-bounce model of mutual reflection, each $\vec{\rho}$ is composed of a linear combination of $\vec{\rho}^{\,onebounce}$ and $\vec{\rho}^{\,nobounce}$. From a measured set of $\vec{\rho}$'s and the fact that $\vec{\rho}^{\,onebounce}$ has already been calculated, we wish to derive $\vec{\rho}^{\,nobounce}$. For this purpose, the most convenient phrasing of the Lawton and Sylvestre [17] analysis is that given by Tominaga and Wandell [25], which they called 'quarter-circle' analysis. Following this method, we first project all the measured $\vec{\rho}$ on surface $A$ onto the $\vec{v}_1^A, \vec{v}_2^A$ plane and then rotate each $\vec{\rho}$ into a new coordinate system with axes $\vec{a_1} \equiv \vec{\rho}^{\,onebounce}$ and $\vec{a_2}$ perpendicular to $\vec{a_1}$ in that plane.

Described in this new coordinate system and normalized to unit length, each measured $\vec{\rho}$ vector must lie within the first quadrant of a unit circle (see Fig. 2). Each $\vec{\rho}$, therefore, has positive components $a_1, a_2$ along the $\vec{a_1}, \vec{a_2}$ directions. The task is to establish limits on the direction of $\vec{\rho}^{\,nobounce}$.

The first constraint on $\vec{\rho}^{\,nobounce}$ stems from the fact that of the measured $\vec{\rho}$'s, the one closest to the $\vec{a_2}$ axis represents the point on the surface where the mutual illumination is least. The $\vec{\rho}$ nearest to $\vec{a_2}$ has the largest ratio of its components, $a_2/a_1$ (histogramming

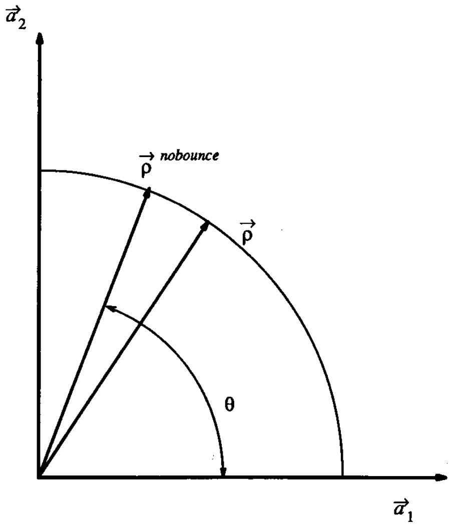


Fig. 2. Projecting normalized RGB vectors onto the cross-planes direction and a perpendicular to it in the object's color plane yields vectors all on a quarter-circle.

the ratios avoids picking an outlier). This $\vec{\rho}$ gives a lower bound on $a_2/a_1$ for $\vec{\rho}^{\,noubounce}$:

$$a_2/a_1 \geq \max_{data}(a_2/a_1).$$

Although our experiments in Section 3 below indicate that the lower bound is the more useful, a theoretical upper bound follows from the condition that $\vec{\rho}^{\,nobounce}$ cannot have negative components. Since $\vec{a_1}$ has positive components and $\vec{a_2}$ is perpendicular to it, it follows that $\vec{a_2}$ has some negative components in RGB color space. As we move away from $\vec{a_2}$ toward $\vec{a_1}$, we reach a point where every component of the $\vec{\rho}$ vector is nonnegative. That point is the upper bound for $a_2/a_1$.

The limiting vector contains just enough of $\vec{a_1}$ added to $\vec{a_2}$ that the component corresponding to the most negative component of $\vec{a_2}$ becomes precisely *zero*. The condition is

$$a_1\vec{a_1} + a_2\vec{a_2} \geq \vec{0}$$

so that

$$a_2/a_1 \leq [\max_{k=1\cdots 3}(-a_{2k}/a_{1k})]^{-1}$$

Once the band of possible $\vec{\rho}^{\,nobounce}$ vectors has been established, the original image $\vec{\rho}$ data is easily decomposed into its components in the $\vec{\rho}^{\,onebounce}$ and $\vec{\rho}^{\,nobounce}$ directions. Of course, the previous analysis has established only that $\vec{\rho}^{\,nobounce}$ lies within a band, so the components are only restricted to be within a band as well.

Given a particular choice of $\vec{\rho}^{\,nobounce}$ from the band of possible $\vec{\rho}^{\,nobounce}$ vectors that forms an angle $\theta$ with $\vec{\rho}^{\,onebounce}$, the components of any $\vec{\rho}$ in these one-bounce and no-bounce directions are given by

$$\vec{\rho} = b_c\vec{\rho}^{\,nobounce} + b_1\vec{\rho}^{\,onebounce}, \qquad (6)$$

with

$$b_c = a_2/\sin\theta, b_1 = a_1 - a2\tan\theta,$$
$$a_1 = \vec{\rho}\cdot\vec{\rho}^{\,onebounce},$$
$$a_2 = \vec{\rho}\cdot\vec{a}_2, \vec{a}_2 \perp \left\{\vec{a}_1 \equiv \vec{\rho}^{\,onebounce}\right\}.$$

The coefficient $b_c$ is proportional to the shading, while $b_1$ is the relative magnitude of the light from the other surface.

### E. Special Cases

There is a special case caused by white surfaces in which the SVD analysis breaks down. For example, suppose a red surface interreflects with a white or gray surface (i.e., $S(\lambda)$ = constant). In that case, the no-bounce color on the red side will equal the one-bounce, interreflection color; and as a result, the second and third eigenvectors will arise from a combination of the second and higher bounces, not the first bounce.

Since the red will be unchanged by reflection off the white surface, it will not be possible to distinguish the no-bounce red from its one-bounce counterpart and the SVD analysis will discover a one-dimensional, not two-dimensional, subspace of color space. As a result, on the red side the mutual illumination component cannot be removed by the algorithm. Being able to determine the effective dimensionality of the subspace is a major strength of the SVD method and parallels its use in determining effective rank of matrices in pseudo-inverse methods. In the present case, it means that the algorithm can report the circumstances in which it will fail.

On the white side, however, the subspace is two-dimensional. Running the algorithm as usual will still produce correct results on the white side. The first eigenvector on the red side will lie essentially along the red no-bounce direction and any plane formed on the red side will therefore necessarily include that red. The white side's white–red plane will intersect the red side's $\vec{v}_1 - \vec{v}_2$ plane in the red no-bounce/one-bounce color. As a result the one-bounce color is correctly determined and the mutual illumination component will be removed successfully from the white side.

Exactly the same reduction in subspace dimensionality occurs whenever the one-bounce color happens to produce the same hue as one of the no-bounce colors. For example, if one object is a saturated red and the other object (a yellow, say) has a reflectance spectrum that is constant over the the red region of the spectrum then the one-bounce color will have approximately the same hue as the red. In these cases, the SVD analysis of dimensionality is again a useful indicator of problems.

The other obvious special case in which the SVD analysis does not apply arises when there is no interreflection. This can occur for two reasons: the edge may be convex not concave; or it may be that the product of the two reflectance spectra is zero. An example of the latter is a saturated red interreflecting with a saturated blue. Since the blue will reflect almost none of the red, the one-bounce "color" is black.

Instead of analysing interreflection in color space it could analysed in chromaticity space. For RGB values inhabiting a plane, the corresponding chromaticity coordinates form a line. The intersection of two such lines will give the chromaticity coordinates of the one-bounce color. This is similar to the chromaticity space analysis of body and specular reflection in [18]. Whether or not the analysis is done with the two-dimensional chromaticity data or the original three-dimensional RGB data makes little difference. In either case, it is useful to find the lines or planes via the SVD so that the effective dimensionality of the data can be determined from the eigenvalues.

In the next section, we apply our method to a simulated image to test the accuracy of the one-bounce model and determine the width of the band of possible no-bounce colors.

## III. Simulated Image Test

### A. Determining the One-Bounce and No-Bounce Colors

We constructed a multispectral image of a theoretical edge composed of two semi-infinite planar surfaces illuminated by diffuse light (see Fig. 1) using the infinite-bounce method described in [3]. We used the reflectances of pink and green Pantone papers and the spectrum of standard illuminant A [28]. From the multispectral image, $\vec{\rho}$ vectors are calculated using the spectral transmission curves of the common color-separation filters, Kodak #25 (red), #58 (green) and #47B (blue).

To challenge the algorithm, we chose a fairly small opening angle $\gamma = 45°$ and made side $B$ twice as long as side $A$. This means that even at the pixel located at the maximum width of side $A$, with $x = x_{max}$, we expect a comparatively high mutual illumination contribution. At the maximum width pixel on side $B$, though, the mutual illumination effect will be negligible. The simulation includes no noise or spatial variation of the illumination.

The first question is how well does the one-bounce model fit this data? An SVD analysis on each side of the edge separately finds eigenvalues for side $A$ of $\Lambda = (22.6, 0.141, 0.00296)$ and for side $B$ of $\Lambda = (22.6, 0.454, 0.030)$. Since the first two eigenvalues outweigh the third in each case, the vectors $\vec{\rho}$ on each side are well modeled by a plane in RGB space. The $\vec{\rho}^{\,onebounce}$ direction is found by crossing these planes, and the normalized result is $\vec{\rho}^{\,onebounce} = (0.613, 0.625, 0.484)$.

Does this color match the actual one-bounce color of the simulated edge? The actual color is $(0.621, 0.63, 0.417)$ and in RGB space the angle between the derived $\vec{\rho}^{\,onebounce}$ and the correct one is 4.4°. As an alternate error measure, we calculate the rms error between the actual and derived one-bounce colors by forming

$$\mathcal{E} \equiv \left[ \left(\vec{\rho}^{\,actual} - \vec{\rho}^{\,approx}\right)^2 / \left(\vec{\rho}^{\,actual}\right)^2 \right]^{1/2} \times 100\%. \quad (7)$$

The result is 7.7%, which seems reasonable.

Of course, we are more interested in whether or not the two no-bounce colors are correct and what choice of $\vec{\rho}^{\,nobounce}$ best represents the band of possible $\vec{\rho}^{\,nobounce}$ vectors. The ratios $a_2/a_1$ for side $A$ range from 0.497 to 0.525. Therefore, the lower bound on this ratio for the no-bounce color is 0.525. The nonnegativity constraint imposes an upper bound of 0.881 on the same ratio. Using the lower bound to calculate the corresponding no-bounce color for side $A$ results in $\vec{\rho}^{\,nobounce} = (0.900, 0.400, 0.173)$. This compares favorably to the actual $\vec{\rho}^{\,nobounce} = (0.914, 0.376, 0.151)$, with an angular difference of 2.05° or $\mathcal{E} = 3.57\%$.

Using the upper bound to calculate the no-bounce color results in $\vec{\rho}^{\,nobounce} = (0.968, 0.251, 0.000)$, which is off by 13.7°. Clearly, this error is high. Even the average of the two estimates leads to an error of 4.82°. On side $B$ the situation is even more lopsided: the lower-bound, no-bounce color has a small error—0.435°—whereas the upper-bound vector has a large error—27.9°. For side $B$ using the lower-bound we calculate $\vec{\rho}^{\,nobounce} = (0.430, 0.785, 0.446)$ compared with the correct values $(0.423, 0.789, 0.446)$.

Although the nonnegativity constraint is theoretically well founded, it actually is producing a very poor limit on the band of no-bounce vectors because it requires that one out of the three RGB components be zero. In Tominaga *et al.* [25], the same theoretical limit works better because there are many more samples taken across the spectrum. In that case, only one small section of the spectrum is restricted to zero in the limiting vector.

Based on the observation that in all the cases we tried the correct vector lies much closer to the limit imposed by the lower bound constraint than that imposed by the upper bound constraint, we abandon the nonnegativity part of the quarter-circle analysis as ineffective and shift to estimating $a_2/a_1$ entirely on the basis of the data point found to be farthest from the crossed-planes color. In essence, this is equivalent to assuming that each surface has at least one point where mutual illumination is negligible.

On a CRT display, the recovered no-bounce colors are indistinguishable by eye from the correct colors. The one-bounce color is an intermediate color between the pink and green of the Pantone papers.

### *B. Shading and Interreflection Factors*

Once $\vec{\rho}^{\,onebounce}$ and $\vec{\rho}^{\,nobounce}$ are known, equation (6) will generate coefficients $b_c(x)$ and $b_1(x)$ for every image location $x$. Using $\vec{\rho}^{\,nobounce}$ for side $A$, the image $b_c(x)\vec{\rho}^{\,nobounce}$ predicts what side $A$ would have looked liked had side $B$ not been in the scene. The image $b_1(x)\vec{\rho}^{\,onebounce}$ represents the mutual illumination incident at each point. Since all the parameters of the simulated edge are known, we can assess how well the algorithm performs by comparing these images to corresponding theoretical ones.

Since the simulated edge was illuminated by diffuse light, we expect that in the absence of mutual illumination there will be no spatial variation in the shading anywhere on either surface. Thus $b_c(x)$ should be constant everywhere and equal to 1 if the normalization is chosen correctly. Fig. 3(a) shows that this expectation is borne out, with the normalization accomplished by dividing by the exact $\|\vec{\rho}^{\,nobounce}\|$. Normalization is necessary because the derived $\|\vec{\rho}^{\,nobounce}\|$ is always a unit vector; its true magnitude is not found.

The mutual illumination image $b_1(x)\vec{\rho}^{\,onebounce}$ should approximately equal the mutual illumination effects of a single bounce since the idea was to capture infinite-bounce effects within a one-bounce model and second and higher bounces are known to be relatively small. If the one-bounce assumption holds, the interreflection factor $b_1(x)$ should equal—up to a single, overall normalization constant—the configuration factor (i.e., the proportion of the light leaving the other surface that arrives at $x$ [3, 9]).

Fig. 3(b) compares $b_1$ to the theoretical configuration factor for the simulated scene's geometry. The results are normalized relative to the actual one-bounce color by dividing by the known $\|\vec{\rho}^{\,onebounce}\|$ Results are good for side $B$, but not for side $A$, although $b_1$ still gives the correct shape for the configuration factor there.

The difference between side $A$ and side $B$ is that for side $B$ the mutual illumination effect becomes negligible far from the edge, whereas this is not the case for side $A$. Since the lower bound estimate of the no-bounce color identifies the pixel with the least amount of one-bounce color as *identical* to the no-bounce color, the results are most satisfactory when there is indeed a spot with very little mutual illumination. Had we used the actual vectors $\vec{\rho}^{\,nobounce}$ in calculating $b_1$, results for both sides would have been excellent.

If there is one point where there is very little incident mutual illumination, then there probably will be a region of neighboring points having little or no mutual illumination, because mutual illumination effects tend to vary slowly spatially. This observation points to a verification procedure that can be used to establish whether or not the algorithm has been applied to an image on which it can succeed. Namely, the algorithm's results are reliable when a significant number of pixels in the original image have the same color as the derived no-bounce color.

In the next section, we apply the algorithm to a real laboratory image.

## IV. Real Image Test

### *A. Experimental Setup*

We used colored Pantone papers to make the convex surfaces. To the eye, these papers appear relatively free of specular highlights.

Fig. 5(a) shows an approximately planar strip of pale blue paper (Pantone #290) making an angle of ~ 45° with another strip (#360) colored grass green. The blue strip was nearly vertical and the green

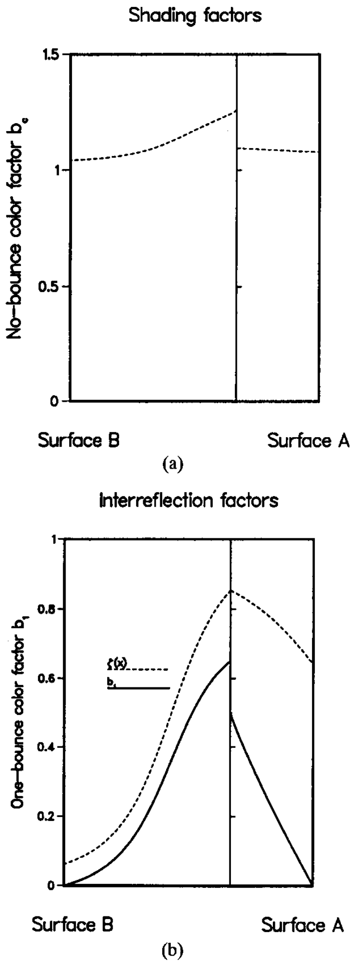


Fig. 3. (a) Shading factors $b_c$ recovered by the algorithm are shown for the planar edge of Fig. 1, with the width of surface $B$ equal to twice the width ofsurface $A$ and $\gamma = 45°$. (b) Interreflection factors $b_1$ recovered by the algorithm for the planar edge (——). For comparison is the theoretical factor $\zeta$ for a single bounce (— —).

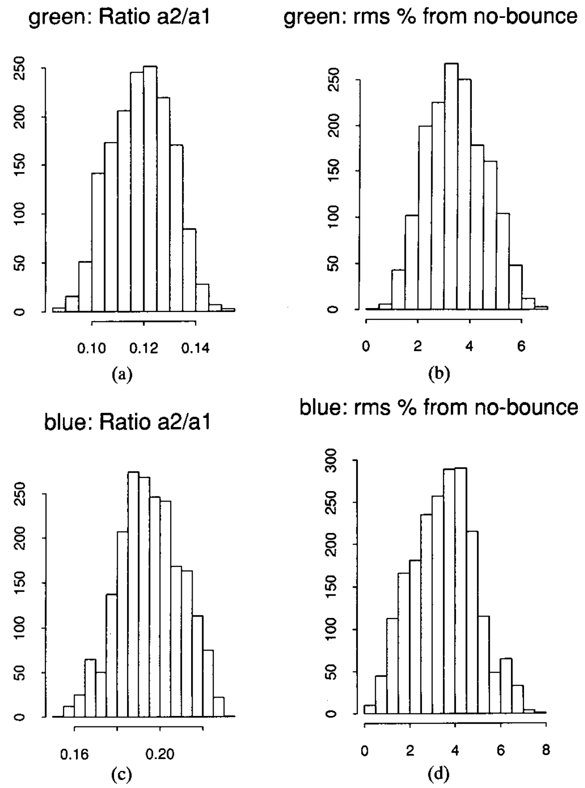


Fig. 4(a). Histogram of ratio $a_2/a_1$ for the green strip's color space plane. (b) Histogram of rms differences between measured green strip pixels and the calculated one-bounce color. (c) Histogram of ratio $a_2/a_1$ for the blue strip's color space plane. (d) Histogram of rms differences between measured blue strip measurements and the calculated one-bounce color.

strip was inclined, with the camera placed directly above the scene. In the image, the blue strip is at the top and the green strip is at the bottom. We placed black cardboard behind the scene.

Mutual illumination induces a color shift that is surprisingly hard to discern by eye, although it is clearly evident in the numerical image data. Since the color image is not that informative, Fig. 5 presents a black and white photograph of the color image.

A tungsten light at a distance of ~ 1 m and at an angle of ~ 45° to the horizontal illuminated the scene. Color images were taken using a CCD Camera with infrared cutoff and color separation filters (Kodak Wratten filters #25, #58 and #47B). The camera responds linearly with intensity, but with unequal sensitivity across the spectrum. For the method described in this paper, sensitivity calibration is not required.

### B. Results of Color Space Analysis

We first segmented the image into three regions: background, blue strip, green strip. SVD analysis of the green strip produces eigenvalues $\Lambda = (3835.2, 43.0, 16.2)$. For the blue strip, $\Lambda = (5482.9, 72.7, 19.9)$. Clearly, the data does not fit a plane as well for this real image as it did for the simulated image. Factors that may be contributing to the third dimension are noise, stray light, and small amounts of specularity.

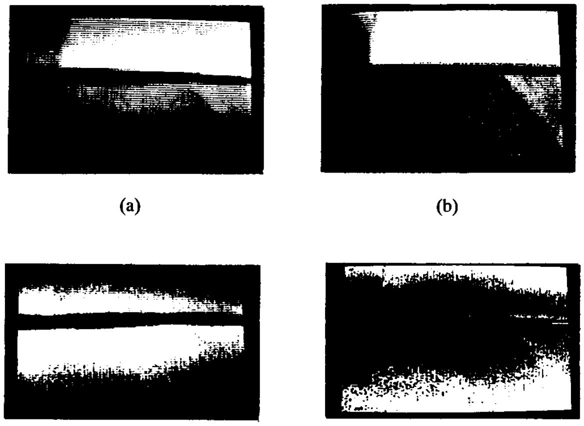
The intersection of the two planes defines the green-to-blue (or equivalently blue-to-green), one-bounce color, $\vec{\rho}^{\,onebounce} = (0.573, 0.653, 0.496)$. Since the green Pantone paper is more saturated than the blue, this color appears a bit more green than blue.

Fig. 4(a) histograms the ratio $a_2/a_1$ for pixels from the green strip. The maximum value for the ratio is 0.154 and it is not an outlier. This ratio defines no-bounce green of $\vec{\rho}^{\,nobounce} = (0.673, 0.633, 0.383)$.

For comparison, the actual green measured under the same illumination but at a location where there was no mutual illumination from the blue strip was $\vec{\rho}^{\,nobounce} = (0.662, 0.636, 0.397)$. This measured no-interreflection color differs by 1.05° from that produced by the algorithm.

To assess the algorithm's results without knowing the actual reflectances of the two colored strips, we can reconstruct an image based on the one-bounce analysis and compare it to the actual image, where the reconstructed image is obtained from $b_1(x)$ and $b_c(x)$ via equation (6). For the green strip, the errors $\mathcal{E}$ between actual data and reconstructed one-bounce model values have median value of only 0.27%. For the blue strip, this value is 0.25%. Clearly, the one-bounce model performs well.

To further assess the results, we apply the verification procedure described above. In particular, the test finds that many pixels in the



(a) (b)

(c) (d)

Fig. 5(a). Image of a blue strip and a green strip. Since the color shift due to mutual illumination is not easily discernible in any case, the image is only shown in black and white. (b) Black and white version of nobounce image representing the the original image with interreflections removed. The shading is as it would have been in the absence of mutual illumination. (c) Image of onebounce color (i.e., the interreflection factors). (d) Image of nobounce colors, shown in black and white, masked with 1 or 0 depending on whether the true color is close to the calculated nobounce color.

original image are close in color to the derived $\vec{\rho}^{\,nobounce}$, which indicates that there is a high likelihood that some location in the image had an insignificant amount of mutual illumination impinging on it and therefore that $\vec{\rho}^{\,nobounce}$ is correct. Fig. 4(b) shows a histogram of differences $\mathcal{E}$ between the derived no-bounce color on the green side and the RGB values of the original image. Many pixels fall close to the calculated green. The average difference from the no-bounce color is 3.4% and the percentage of pixels within 3% of the no-bounce color is 39.1%.

For the blue strip, Fig. 4(c) shows the distribution of $a_2/a_1$. The maximum value of the ratio is 0.232. Composing a no-bounce color from this ratio we find $\vec{\rho}^{\,onebounce} = (0.735, 0.576, 0.357)$. Fig. 4(d) shows the distribution of $\mathcal{E}$ from this color for the blue strip. The average difference from the no-bounce color is 3.5% and the percentage of pixels within 3% of the no-bounce color is 35.4%. The actual blue measured at a location without mutual illumination, but under the same direct illumination from the light bulb, was $\vec{\rho}^{\,nobounce} = (0.724, 0.5834, 0.367)$; the vector produced by the algorithm was 1.0° away from this color.

### C. Output Images

The algorithm decomposes the original image into two component images. Fig. 5(b) shows the original scene with mutual illumination removed (i.e., $b_c(x)\vec{\rho}^{\,nobounce}$). Fig. 5(c) shows the image of the mutual illumination component (i.e., $b_1(x)\vec{\rho}^{\,onebounce}$). Since this effect is small, the image has been multiplied by the factor 3 for display. As should be expected, the image of mutual illumination is brightest near the edge, where the interreflection effect is strongest.

Fig. 5(d) shows which parts of the original image are similar in color to the calculated no-bounce color by masking the no-bounce color with a 1 when the percentage difference $\mathcal{E}$ between the measured pixel values and the no-bounce color is below a threshold of 3%. In the resulting image, there are connected regions close to the calculated color, so we can have confidence in the algorithm's results. These regions are away from the edge, where the effect of mutual illumination diminishes.

## V. Conclusion

Under a one-bounce model of interreflection, the RGB vectors for a region of uniform color lie in a plane. Intersection of two such planes from two interreflecting regions yields the color of the interreflected light. This one-bounce color provides one of the two axes on which the original data can be projected. By projecting the data onto this axis, an image of the mutual illumination field is produced. Quarter-circle analysis constrains the second axis—the no-bounce color—to a band, but in fact, only one limit of the band turns out to be useful. Projection of the original image onto the axis defined by this limit results in a no-interreflection image (i.e., an image of each surface *as if* the other surface had not been present).

The method generalizes to interreflections caused by self-reflection. For example, an image of the concave surface inside a cup will have a second illumination component due to the mutual illumination from the cup to itself, and the color of the interreflected light will not in general be the same as the cup itself.[3] In situations generating three components (e.g., a main reflection component, self-reflection and interreflection; or main reflection, interreflection and specularity) the full Lawton–Sylvestre–Kawata method—which is not restricted to finding planar subspaces—in principle should apply, but we have not tested it.

The color space method is quick (well under a second), simple, and includes a reliability check. Results from tests on real and synthesized images show that it works well. Any shape-from-shading method can be applied directly to the no-interreflection image. The mutual illumination image approximates an image of configuration factors encoding other potentially useful geometrical information about the scene.

## References

[1] R. Bajcsy, S. W. Lee, and A. Leonardis, "Color image segmentation and with detection of highlights and local illumination induced by interreflections," *IEEE Int. Conf. Pattern Recognit.*, Atlantic City, NJ, vol. 1, 1990, pp. 785–790.

[2] M. H. Brill, "Image segmentation by object color: A unifying framework and connection to color constancy," *J. Opt. Soc. Am. A*, vol. 7, pp. 2041–2047, 1990.

[3] M. S. Drew and B. V. Funt, "Calculating surface reflectance using a single-bounce model of mutual reflection," in *Proc. IEEE Int. Conf. Computer Vision*, Osaka, Japan, Dec. 4-7, 1990, pp. 393–399.

[4] ——, "Variational approach to mutual illumination in color images," *J. Opt. Soc. Am. A*, vol. 32, pp. 1255–1265, 1992.

[5] D. Forsyth, "A novel algorithm for color constancy," *Int. J. Comput. Vision*, vol. 5, pp. 5–36, 1990.

[6] D. Forsyth and A. Zisserman, "Mutual illumination." in *Proc IEEE Comput. Vision and Pattern Recognit.*, 1989, pp. 466–473.

[7] ——, "Shape from shading in the light of mutual illumination," *Image and Vision Computing*, vol. 8, pp. 42–49, 1990.

[8] B. V. Funt and M. S. Drew, "Color space analysis of mutual illumination," Tech. Rep. CSS/LCCR TR 91-03, Simon Fraser Univ., School of Computing Sci., 1991.

[9] B. V. Funt, M. S. Drew, and J. Ho, "Color constancy from mutual reflection," *Int. J. Comput. Vision*, vol. 6, pp. 5–24, 1991.

[10] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: John Hopkins Univ. Press, 1983.

[11] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," *Comput. Graphics*, vol. 1, pp. 213–222, 1984.

[12] J. Ho, B. V. Funt, and M. S. Drew, "Separating a color signal into illumination and surface reflectance components: Theory and applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 966–977, Dec. 1990.

[13] B. K. P. Horn, "Understanding image intensities," *Artificial Intell.*, vol. 8, pp. 201–231, 1977.

[3] For example, for the edge of Section III-A, the pink color is (0.914, 0.376, 0.151), and the pink-reflected-by-pink color is $(0.961, 0.259, 0.101)$; the angular difference is 7.8° or $\mathcal{E} = 6\%$.

[14] J. E. Kaufman, *IES Lighting Handbook*, 4th ed. New York: Illuminating Eng. Soc., 1966.
[15] S. Kawata, K. Sasaki, and S. Minami, "Component analysis of spatial and spectral patterns in multispectral images. I. Basis," *J. Opt. Soc. Am. A*, vol. 4, pp. 2101–2106, 1987.
[16] G. J. Klinker, S. A. Shafer, and T. Kanade. "The measurement of highlights in color images," *Int. J. Comput. Vision*, vol. 2, pp. 7–32, 1988.
[17] W. H. Lawton and E. A. Sylvestre, "Self modeling curve resolution," *Technometrics*, vol. 13, pp. 617–633, 1971.
[18] H.-C. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," *J. Opt. Soc. Am. A*, vol. 3, pp. 1694–1699, 1986.
[19] S. K. Nayar and Y. Gong, "Colored interreflections and shape recovery," in *Image Understanding Workshop*, in DARPA, San Diego, CA 1992.
[20] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from interreflections," in *Proc. IEEE Int. Conf. Comput. Vision*, Osaka, Japan, Dec.4-7, 1990, pp. 2–11.
[21] S. A. Shafer, "Using color to separate reflection components," *Color Res. Appl.*, vol. 10, pp. 210–218, 1985.
[22] S. A. Shafer, T. Kanade, G. J. Klinker, and C. L. Novak, "Physics-based models for early vision by machine," in *Perceiving, Measuring, and Using Color*. New York: SPIE, 1990, vol. 1250, pp. 222–235.
[23] R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*. New York: Hemisphere, 1981.
[24] S. Tominaga and B. A. Wandell, "Standard surface-reflectance model and illuminanat estimation," *J. Opt. Soc. Am. A*, vol. 6, pp. 576–584, 1989.
[25] ——, "Component estimation of surface spectral reflectance," *J. Opt. Soc. Am. A*, vol. 7, pp. 312–317, 1990.
[26] F. Tong and B. V. Funt, "Specularity removal for shape from shading," in *Proc. Vision Interface 1988*, Edmonton, AB, Canada, 1988, pp. 98–103.
[27] ——, "Removing specularities from color images for shape from shading," in *Computer Vision and Shape Recognition*, A. Krżyzak, Ed. New York: World Scientific, 1989.
[28] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulas*, 2nd ed. New York: Wiley, 1982.

# Vector Quantization Technique for Nonparametric Classifier Design

Qiaobing Xie, Charles A. Laszlo, and Rabab K. Ward

***Abstract*—An effective data reduction technique based on vector quantization is introduced for nonparametric classifier design. Two new nonparametric classifiers are developed, and their performance is evaluated using various examples. The new methods maintain a classification accuracy that is competitive with that of classical methods but, at the same time, yields very high data reduction rates.**

***Index Terms*— Condensing algorithms, data reduction, *k*-nearest-neighbor (*k*NN) classifier, nonparametric classification, Parzen kernel classifier, vector quantization.**

## I. Introduction

Nonparametric classification has been of great importance in statistical pattern recognition [1], [2]. When dealing with problems of nonnormal distributions, nonparametric classifiers frequently show much higher classification accuracy than that achieved by the popularly used parametric classification approaches, such as the linear classifier and the quadratic classifier. In addition, it has been found that the nonparametric algorithms sometimes outperform their parametric counterparts even when the data are from normal populations [3], [4]. In practice, however, the application of the nonparametric classifiers often suffers from various difficulties, especially as the size of the problem increases.

The common disadvantages of classical nonparametric approaches (kernel estimator, *k*NN classifier, etc.) are their computational complexity and the requirement for a very large amount of computer storage to retain the design sets. Unfortunately, large design sets are always desirable because nonparametric methods require adequate statistical information. Thus, on-line application of nonparametric classifiers is rare since these are usually too complex and slow in operation. Their uses are often limited to situations where the computation time is not a crucial factor, such as in the estimation of the Bayes error and data structure analysis [1].

A solution to the above problems is to reduce the size of the design set while insisting that the classifiers built on the reduced design set should perform as well, or nearly as well, as the classifiers built on the original design set. This idea has been explored for various purposes over a period of time and resulted in the development of many algorithms of *k*NN classifier design using reduced sample sets. Particularly noteworthy are the condensed NN (CNN) [5], the reduced NN (RNN) [6], and the edited NN (ENN) [7]. In these algorithms, iterative processes are used to test the effect on the classification performance as each individual point is moved in and out of the design set, and only the "good" points are retained. For very large design sets, these methods are often tedious and difficult to implement since a new classifier is in fact built and evaluated every time a point is moved in or out of the design set. The most serious disadvantage is that the final reduction rate is usually low and not under the control of the algorithms, e.g., it depends entirely on the nature of the sample set to be reduced.

Recently, two nonparametric data reduction algorithms were proposed by Fukunaga *et al.* for the Parzen‘s kernel classifier and the NN classifier design, respectively [1], [8], [9]. Their algorithms find the optimal reduced design set from the original design set in the sense that the difference between the probability density function estimated from the reduced set and that estimated from the original set is minimized. Bearing some similarities to the traditional reduced data *k*NN algorithms, their algorithms iteratively move each individual point in and out of a tentatively chosen reduced sample set and test the resultant effect on the criterion function. To avoid an exhaustive search of all possible subsets, which is impractical, the optimization scheme used in Fukunaga's algorithms can achieve a local optimum. The computational complexities of these algorithms are considerable. Moreover, the initial guess of the reduced sample set is of crucial importance in Fukunaga's reduced NN algorithm. Thus far, only an intuitively developed initial assignment procedure for the 2:1 reduction rate case has been published [9].

## II. Two New Nonparametric Classifiers Using Vector Quantization Technique

In this article, we introduce a new approach for nonparametric data reduction using the vector, or block, quantization technique. As a mathematical process, optimal vector quantization has already been

Manuscript received September 23, 1991; revised June 2, 1992. Recommended for acceptance by Editor-in-Chief A. K. Jain.

The authors are with the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada V6T 1Z4.

IEEE Log Number 9211830.

0162–8828/93$03.00 © 1993 IEEE