

Multispectral Processing Without Spectra

Mark S. Drew	and	Graham D. Finlayson
School of Computing Science		School of Information Systems
Simon Fraser University		University of East Anglia
Vancouver, B.C.		Norwich
Canada V5A 1S6		UK NR4 7TJ
mark@cs.sfu.ca		graham@sys.uea.ac.uk

Mailing address:

Dr. Mark S. Drew

School of Computing Science,

Simon Fraser University,

Vancouver, B.C.,

Canada V5A 1S6

Telephone: 604-291-4682

Fax: 604-291-3045

Running Head: Spectra without spectra

<http://www.cs.sfu.ca/cs/~mark>

<http://www.sys.uea.ac.uk/people/graham>

Abstract

It is often the case that multiplications of whole spectra, component by component, must be carried out, for example when light reflects from or is transmitted through materials. This leads to particularly taxing calculations, especially in spectrally-based ray tracing or radiosity in graphics, making a full-spectrum method prohibitively expensive. Nevertheless, using full spectra is attractive because of the many important phenomena that can only be modelled using all the physics at hand. Here we apply to the task of spectral multiplication a method previously used in modelling RGB-based light propagation. We show that we can often multiply spectra without carrying out spectral multiplication.

In previous work we developed a method called “spectral sharpening” which took camera RGBs to a special sharp basis that was designed to render illuminant change simple to model. Specifically, in the new basis one can effectively model illuminant change using a diagonal matrix, rather than the 3×3 linear transform that results from a 3-component finite-dimensional model.¹ Here, we apply this idea of “sharpening” to the set of principal component vectors derived from a representative set of spectra which might reasonably be encountered in a given application. With respect to the sharp spectral basis we show that spectral multiplications can be modelled as the multiplication of the basis coefficients. These new product coefficients applied to the sharp basis serve to accurately reconstruct the spectral product.

Although the method is quite general, here we show how to use spectral modelling by taking advantage of metameric surfaces, ones that match under one light but not another, for tasks such as volume rendering. The use of metamers allows a user to pick out or merge different volume structures in real time, simply by changing the lighting.

1. Introduction

Given 3-dimensional models of a set of objects together with the spectral reflectance characteristics of reflectances (the object colours) and a spectral representation of the illumination field it is possible to render (or synthesize) highly accurate photo-realistic images. There are many applications of this technology. In film, it is now commonplace to see synthetically generated images, e.g., the dinosaurs in Jurassic Park are the result of computer graphics rendering as is the whole of Toy Story. Rendered images are commonplace in domains such as the Graphic Arts (for marketing), and computer games, and they are also an enabling technology for virtual reality since we can only make virtual worlds real if we can accurately synthesize the real world.

While the applications of rendered images are many, there is a problem. Modelling the light that is reflected from a scene (the light that would enter our eye if we were looking upon the same scene) is a computationally very expensive process. Full spectral representations of illumination strike full spectral surface reflectances. The result is a new spectral power distribution which is reflected and goes on to strike other reflectances, and so on. Even with the most powerful of today's computers it can take up to a day of processing to arrive at a single good looking image. It becomes clear that the cost of rendering a sequence of images (for a film) is enormous.

Due to the computation involved, computer graphics has sought ways to make the cost of multispectral rendering less (see reviews of older and newer methods in Ref.²). These methods start with the observation that colour is basically a 3-dimensional problem. All colours we see can be described using three values (e.g., redness, greenness and blueness). That this is so is due to the processing hardware of our eye: retinal cone sensors are most sensitive to Red, Green and Blue light. This said, it might be reasonable to wonder if we can represent light and surface spectral quantities by 3 numbers — redness, greenness and blueness *factors* —

and moreover, that the interaction of light and surface would be modelled by multiplying the corresponding factors of the light and surface. As an example if the illumination were 50%, 70% and 20% red, green and blue and the surface were 20%, 50%, 10% red, green and blue, then the combined (multiplied) RGB would be 10%, 35% and 2%. And in fact, this is how most of computer graphics proceeds.

This simple RGB factor strategy works quite well provided the light is fairly white³ — and we find it so here, as well. Not only can images be generated at a fraction of the computational cost but the generated images look good. However, the devil is in the details. Occasionally, the colours rendered in images will not look right and this is evidence of this *Factor Model* not working. In other applications where colour accuracy is crucial, e.g., in on-line catalogues, people expect to see true colours so even if the colours in a rendered image look good, if they are not truly accurate the customer will complain (over 50% of some types of returned goods from catalogues are due to colour mismatches). In virtual reality or medical applications it may be crucial to exactly model spectral interactions. It is well known that when we move between illuminants the relationships between colours change. In a pathological case two surfaces that look the same under one light can look different under another light. The Factor Model predicts that all colour relations are preserved across all illuminations. For such applications as rendering tissue surfaces in medical applications, the differences between true colours and those predicted by the Factor Model may become crucial.

The fundamental result presented here is to extend the simple red, green and blue factor model so that it more accurately accounts for *spectral* interactions and so ultimately produces more accurate looking images. Our idea is a simple one: suppose we capture the salient information about spectra using a principal component analysis. Using just the first few basis vectors will adequately model most spectra, and of course we can carry out spectral

multiplies using just the basis coefficients. But the problem is that if we reduce the problem to a p -dimensional space, say, we still have matrix multiplications to compute in order to determine the new set of coefficients for a product spectrum (cf.⁴). However, we propose that if we “sharpen” these basis vectors so that they are maximally concentrated around specific wavelength regions then the sharpened basis will promote the simple factor model. We effectively multiply spectra by multiplying coefficients with respect to a sharp basis (and the resulting multiplied spectra can, if necessary, be reconstructed by multiplying these coefficients times the sharp basis). The intuition of why this approach might work becomes clear if we consider using the same number of basis functions as there are sampling points. In this case the sharp basis consists of delta functions positioned at each sampling wavelength and the basis coefficients are simply the spectra themselves. In the limiting case the sharp approach to spectral multiplication results in the actual spectral multiplication.

We first carried out “spectral sharpening” on the three camera sensors for a particular device (or for the colour-matching functions of the eye): the problem is to find a 3×3 matrix such that the combinations of the three camera sensors are maximally narrowband⁵ (in a simple sense, most like delta functions). For RGB colour, coefficients in terms of the sharpened sensors obey much more accurately a Factor Model: colours for lights interacting with surfaces are approximately given in terms of the RGB colour of the light multiplied component-wise times the RGB colour of the surface, as in traditional computer graphics.

Here we find the linear combination of *spectral basis functions* which are sharpest. By analogy to the RGB Factor Model (and by rigorous proof later in the paper) two spectra, represented by their coefficients calculated with respect to a sharp basis, can be multiplied together by simply multiplying the coefficients themselves: coefficients can simply be multiplied component-wise, for an accurate determination of product spectra. The dimension of the *spectral Factor Model* is chosen to meet a particular error criterion. In the limiting

case if zero spectral error is sought we resort to the full spectral model. However, in our experiments we have found that a 5 to 7-dimensional spectral Factor Model maintains good spectral accuracy.

Rendering an image with, say, 6-dimensional basis coefficient vectors is more than five times faster as compared to the cost of rendering full multispectral images with, say, 31 sample points resulting from sampling at 10nm wavelength intervals from 400 to 700nm. We point out to the reader that this approach is much cleverer than using, say, 6 sample points. In this case we might accurately model spectral interaction at the chosen sample points but, by definition, we would be blind to what happens between the sample points (if these are obtained by sampling the input function directly, or else blind to the information between values actually used). In our method we have a simple 6 factor multiplication model which models the spectral interaction *at all wavelengths*. An alternate reasonable approximation using, say, six block functions is shown to be overwhelmingly worse than the new method.

We believe the method set out here is significant outside the domain of computer graphics. Since we can show that the spectral Factor Model is an accurate representation of the actual multispectral quantities it follows that in proposing the spectral Factor Model of image formation we are saying something fundamental about our understanding of image formation — in particular that the interaction of lights and surfaces can be modelled by a p -dimensional spectral Factor Model instead of spectra. This insight is key not only because it reduces the dimension of spectral representations of light and surface but also because the spectral Factor Model makes some sorts of image analysis tasks easy.

As an example, if we capture an image under yellow light the image colours are more yellow than they ought to be. In contrast, if we ourselves viewed the same scene we would not see the yellow cast: we possess the perceptual facility called *colour constancy*, i.e., the ability to process an image to remove colour bias due to illumination (see, e.g.,⁶⁻¹¹). In

digital photography, authors have long sought algorithms to provide colour constancy. To make the method work they adopt the 3-dimensional Factor Model — they assume RGBs across illumination are related by three scalar multiplications. They do this mostly because it simplifies the computational cost of processing. The method presented renders multispectral colour constancy equally easy to solve.

2. Spectral Sharpening and the RGB Factor Model of Colour

A. Colour Image Formation

To motivate the following discussion of filters, let us briefly consider a colour camera system. Consider the RGB triple \mathbf{r} resulting from light with spectral power distribution $E(\lambda)$ impinging, perhaps after multiple scattering events or directly or both, on a surface with spectral reflectance function $S_x(\lambda)$. For simplicity, we mention reflectance, but in general any combination of scattering events is implied.

If the camera system colour sensors have three sensitivity functions $\mathbf{q}(\lambda)$ then, for a surface point \mathbf{x} that sees this light, we have a colour

$$\mathbf{r}_{\mathbf{x}} = \int f(\mathbf{x}, \lambda) E(\lambda) S_{\mathbf{x}}(\lambda) \mathbf{q}(\lambda) d\lambda \quad (1)$$

where \mathbf{x} indexes 2D retinal coordinates and $f(\mathbf{x}, \lambda)$ encapsulates a geometric factor which is usually taken to be separable and independent of λ .

Ignoring the geometric factor, the simple colour \mathbf{b} for light $E(\lambda)$ reflected from surface $S(\lambda)$ is given by

$$\mathbf{b} \equiv \int E(\lambda) S(\lambda) \mathbf{q}(\lambda) d\lambda \quad (2)$$

When $E(\lambda)$ is an illuminant, the product $E(\lambda) S(\lambda)$ is termed the *colour signal*.¹² Working

with sampled spectra, then for example if we sample the visible spectrum from 400 nm to 700 nm in 10 nm increments, we form matrix \mathbf{Q} from samples of the three sensor functions $\mathbf{q}(\lambda)$, and each column of \mathbf{Q} has 31 entries. Thus, the colour sensor matrix \mathbf{Q} is 31×3 , the illuminant, reflectance, and product of illuminant and surface spectra — the colour signal — are 31×1 vectors, and the colour \mathbf{b} is typically a 3-vector. Denoting the illumination vector as \underline{E} , surface as \underline{S} , and colour signal as \underline{C} , we can write:

$$\underline{C} = \underline{E} \times \underline{S} \tag{3}$$

where \times multiplies corresponding vector elements together. Because the \times operator is not part of standard linear algebra we rewrite this equation as follows:

$$\underline{C} = \text{diag}(\underline{E})\underline{S}. \tag{4}$$

The *diag* function places the 31-vector \underline{E} along the diagonal of a 31 by 31 diagonal matrix. We can now rewrite (2) as:

$$\mathbf{b} = \mathbf{Q}^T \text{diag}(\underline{E})\underline{S} = \mathbf{Q}^T \underline{C}. \tag{5}$$

Now image formation is expressed algebraically. Calculating the colour signal vector takes 31 multiplications, and converting the colour signal into an RGB takes 31×3 multiplications and 30×3 additions. In rendering a multispectral scene, spectra must be multiplied many times for every pixel — rendering an image may take *billions* of calculations.

B. RGB Factor Model of Colour

In computer graphics, a simple approach to approximating (1) consists of multiplying the RGB triple for the illuminant times the RGB triple for the surface. We call this the RGB Factor Model. In terms of eq. (1) this amounts to approximating the colour equation (2) with components b_k by

$$b_k \simeq s_k e_k / w_k, \quad k = 1..3 \quad (6)$$

where s_k is the surface colour under equi-energy white light

$$\mathbf{s} = \mathbf{Q}^T \underline{\mathbf{S}} \quad (7)$$

and e_k is the colour of the illuminant,

$$\mathbf{e} = \mathbf{Q}^T \underline{\mathbf{E}} \quad (8)$$

The camera scaling term is

$$\mathbf{w} = \mathbf{Q}^T \underline{\mathbf{1}} \quad (9)$$

where $\underline{\mathbf{1}}$ is a vector of 31 ones. The intuition behind (6) through (9) can be found by setting $w_k = 1$ and thinking of the RGB sensors as having single wavelength sensitivities. In this case s_k and e_k are simply the spectral measurements for the k th sensor (for the k th delta function). The \mathbf{w} factor is a bookkeeping term that is needed to account for the magnitude of the sensor curves. Without it, e.g., delta-function sensors would not give perfect colour constancy.

Borges³ carefully considered this approximation when delta function sensitivities are not assumed. He showed that for fairly arbitrary sensor bases the RGB Factor Model is accurate provided illuminants (or surfaces) are “white enough”. In practice, the light can be relatively

non-white and still give accurate enough results under an RGB Factor Model so long as the sharpest sensor basis, the set of sensors most like delta functions, is used.⁵

C. Spectral Sharpening

Let us carry out a spectral sharpening transform⁵ in order to concentrate the “energy” of the three filters into three separate “sharpening intervals” ψ_k , $k = 1..3$, within the visible spectrum. Each sharpening interval is a range of wavelengths within which we wish to concentrate energy; e.g., for the red sensor, ψ_1 would be an interval at the red end of the spectrum. To sharpen, we wish to determine a 3×3 matrix \mathbf{T} such that the filter set \mathbf{Q} is transformed into a new set $\tilde{\mathbf{Q}}$ via

$$\tilde{\mathbf{Q}} = \mathbf{Q} \mathbf{T} . \quad (10)$$

If the visible spectrum consists of wavelengths ω , then our objective is to decrease the amount of energy for wavelengths $\phi = \omega - \psi$ *outside* the sharpening interval ψ . We may choose a different sharpening interval for each of the three filters and hence carry out a separate minimization for each of the three colour channels.

Thus, spectral sharpening consists of finding a 3-component vector \mathbf{t} , the k th column of matrix \mathbf{T} , that minimizes the least squares summation

$$\min \sum_{\lambda \in \phi_k} [\mathbf{Q}(\lambda) \mathbf{t}]^2 + \mu \left\{ \sum_{\lambda \in \omega} [\mathbf{Q}(\lambda) \mathbf{t}]^2 - 1 \right\} , \quad k = 1..3 \quad (11)$$

where μ denotes a Lagrange multiplier and \mathbf{t} is a 3-vector to be solved for. The Lagrange multiplier term forces the magnitude of the calculated sensor to equal 1 and so the minimization solves for the sensor that has minimum sensitivity outside the sharpening interval.

Of course, in carrying out the minimization there is no guarantee that the resultant sensors will be all non-negative. We can, in fact, guarantee non-negativity of the resulting

filter set by carrying out a constrained optimization:^{13,14}

$$\begin{aligned}
 & \min \sum_{\lambda \in \phi_k} [\mathbf{Q}(\lambda)\mathbf{t}]^2 \quad , \quad k = 1..3 \\
 & \text{with constraints} \left\{ \begin{array}{l} \sum_{\lambda \in \omega} [\mathbf{Q}(\lambda)\mathbf{t}]^2 = 1, \text{ L}_2 \text{ normalization} \\ \mathbf{Q}(\lambda)\mathbf{t} \geq \mathbf{0}, \quad \text{non-negative sensor result .} \end{array} \right. \quad (12)
 \end{aligned}$$

Enforcing non-negativity makes intuitive sense: we represent spectra by weighted averages in narrow wavelength bands and so the link with delta functions is clear. Moreover, positive RGBs in computer graphics usually represent colours that can be displayed. However, we point out that mathematically the spectral Factor Model does not preclude negative factors but it does preclude negatives in the spectral reconstructions.

The point of spectral sharpening is that it enhances the accuracy of a simple diagonal transform for expressing the changes in RGB required to effect a correct shift under lighting change. Simply, if we make the light redder then we can model this by simply increasing the value of all the red responses relative to the green and blue responses. One implication of this is that the ratio of RGB pairs measured across illuminants (same surface, two lights) should be constant.

Let us image a reference surface (typically a white patch) with surface spectral reflectance \underline{S}_{ref} , and adjust all RGBs \mathbf{r} by the assumption that these change as do the RGBs of the reference surface under illuminant change $E \rightarrow E'$:

$$\frac{\mathbf{b}}{\mathbf{b}'} = \frac{\mathbf{Q}^T \text{diag}(\underline{E}) \underline{S}}{\mathbf{Q}^T \text{diag}(\underline{E}') \underline{S}} \simeq \frac{\mathbf{Q}^T \text{diag}(\underline{E}) \underline{S}_{ref}}{\mathbf{Q}^T \text{diag}(\underline{E}') \underline{S}_{ref}} . \quad (13)$$

If (13) is accurate then illuminant change is modelled by three simple scalars, that is,

by a diagonal matrix. As mentioned earlier, the diagonal matrix model, or equivalently the ratio stability of (13), would hold exactly if the sensor set \mathbf{Q} consisted of very narrowband curves. This is precisely what the spectral sharpening transform (10) aims to do: in the transformed colour space, the diagonal model of change under illumination shift (the von Kries or coefficient rule) is known to hold much more accurately since the sensor curves are more narrowband.⁵ Denoting colours in the transformed colour space as $\tilde{\mathbf{b}}$, we have

$$\frac{\tilde{\mathbf{b}}}{\tilde{\mathbf{b}}'} \simeq \frac{\tilde{\mathbf{Q}}^T \text{diag}(\underline{E}) \underline{S}_{ref}}{\tilde{\mathbf{Q}}^T \text{diag}(\underline{E}') \underline{S}_{ref}} = \frac{\tilde{\mathbf{b}}_{ref}}{\tilde{\mathbf{b}}'_{ref}}. \quad (14)$$

It is simple to show that if the ratio model (14) outperforms (13) then sharp sensors must also help the RGB Factor Model (6). To see this, suppose we wish model the colour $\tilde{\mathbf{b}}'$ for light \underline{E}' reflected from surface \underline{S} . Adopting the RGB Factor Model, this colour (in the space of transformed sensors) is given by

$$\tilde{\mathbf{b}}' = \tilde{\mathbf{Q}}^T \text{diag}(\underline{E}') \underline{S} = \frac{\begin{bmatrix} \tilde{\mathbf{Q}}^T \underline{E}' \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{Q}}^T \underline{S} \end{bmatrix}}{\tilde{\mathbf{Q}}^T \text{diag}(\underline{1})}. \quad (15)$$

Now let us define a special reference surface \underline{S}_{ref} with all components equal to unity (a perfect white diffuser) and an equi-energy white light \underline{E}_w , also with all components equal to unity. Then the following holds:

$$\tilde{\mathbf{b}}' = \frac{\begin{bmatrix} \tilde{\mathbf{Q}}^T \text{diag}(\underline{E}') \underline{S}_{ref} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{Q}}^T \text{diag}(\underline{E}_w) \underline{S} \end{bmatrix}}{\tilde{\mathbf{Q}}^T \text{diag}(\underline{E}_w) \underline{S}_{ref}} = \frac{\tilde{\mathbf{b}}'_{ref} \tilde{\mathbf{b}}}{\tilde{\mathbf{b}}_{ref}}, \quad (16)$$

where $\tilde{\mathbf{b}}$ corresponds to the colour of illuminant \underline{E}_w reflected from surface \underline{S} . Rewriting, we simply have the von Kries rule (the diagonal model of change under illumination shift):

$$\frac{\tilde{\mathbf{b}}'}{\tilde{\mathbf{b}}} = \frac{\tilde{\mathbf{b}}'_{ref}}{\tilde{\mathbf{b}}_{ref}}. \quad (17)$$

The RGB Factor Model implies the von Kries rule (17) and vice versa. Thus, not unexpectedly, since we know that making the sensors more narrowband promotes the von Kries rule, we also know that it promotes the RGB Factor Model as well. The RGB Factor Model is a special application of the von Kries rule.

The RGB Factor Model is appealing in RGB graphics because of its simplicity and, importantly, simplicity leads to fast rendering.

If we could apply a reasoning similar to the RGB Factor Model of colour to the problem of multiplying spectra using a lower dimensional, coefficient space rather than full spectra then graphics calculations could be greatly sped up for spectral rendering. And in fact we can indeed accomplish just this by **sharpening the basis vectors**.¹⁵ Now we show that, again, sharpening can allow us to apply the Factor Model to this higher-dimensional problem.

3. Spectral Multiplication

In a raytracing setting, we would wish to model every spectrum as it interacts with surfaces and transmission boundaries by simply cumulatively multiplying weights in a Finite-Dimensional Model (see⁷), componentwise. While this is in fact not possible for most spectral basis sets it is accurate for a particular sharp basis. We show that a spectral basis, chosen to represent the statistics of real spectra, can be sharpened so that the resultant basis has its sensitivity concentrated in narrow wavelength bands and, furthermore, that these bands are mostly disjoint from one another. Coefficients *do* multiply for the sharp basis.

A. *Sharpening Basis Vectors*

Suppose that we form a set of colour signals, using a variety of natural lights multiplied by a collection of real spectral reflectance functions. Here, we made use of the set of standard illuminants¹⁶ A, C, D48, D55, D65, D75, and D100. These illuminants represent incandescent

lighting (illuminant A), and a variety of standard daylights at correlated colour temperatures from 4800° to 10000° . For reflectances, let us use the 170 reflectances of natural objects measured by Vrhel et al.¹⁷ Colour signals are formed from products of these illuminants and reflectances (no transmission functions were included in the set of model colour signals).

The first five SVD vectors for this set are shown in Fig. 1(a) — the first basis vector is akin to an average (cf. usage of a colour signal basis in Ref.¹⁸). To complete the analogy with the sharpening of sensor filters, let us denote by \mathbf{Q} the 31×5 matrix of SVD vectors, and denote by $\widetilde{\mathbf{Q}}$ the sharpened set. Thus, Fig. 1(a) shows the unsharpened set of basis vectors \mathbf{Q} . In comparison, Fig. 1(b) shows the sharpened set $\widetilde{\mathbf{Q}}$, where we have carried out the same optimization as in eq.(12), but for $k = 1..p$ with $p = 5$ different channels and sharpening intervals. In order to carry out the minimization (12) we need to specify sharpening intervals. Here we simply used the approximately evenly spaced intervals [410–450], [470–510], [530–570], [590–630], [650–690] nm. In order to obviate scaling terms w_k , $k = 1..5$ in the spectral Factor Model, we re-normalize each column of transformed “filter” matrix $\widetilde{\mathbf{Q}}$ so that it adds to 1, yielding scaling terms that are all unity.

There is no difference in the amount of information carried in a set of coefficients of the 31×5 matrix \mathbf{Q} than in coefficients of the sharpened set $\widetilde{\mathbf{Q}}$, since $\widetilde{\mathbf{Q}}$ is simply a matrix transform \mathbf{T} away from \mathbf{Q} , as in eq. (10). It is useful to determine just how accurately the basis set \mathbf{Q} captures spectral shapes. Suppose we model all $170 \times 7 = 1190$ colour signals using seven basis vectors and see how well spectra are modelled. Then, it turns out that the average RMS error in modelling each spectrum is 5.48%. The median RMS error over the whole set is 2.97%. Considering the illuminants alone we find a mean and median RMS error of 2.69% and 2.58%, smaller since the illuminant SPDs have lower dimensionality.¹⁹ Comparing to the most reasonable simpler model, that of using block filters to model spectra, illuminants are reconstructed with mean and median errors 33.00% and 30.62%, respectively

— the new method is substantially better than a simpler approach.

These spectral error numbers are actually deceptively high: errors in RGB *colours* formed from such approximate spectra are of course much smaller. For the colour signal set, the least squares approximations have mean RMS CIE tristimulus XYZ vector (see Ref.¹⁶) error of only 0.75% and median error of 0.42%. In terms of visible differences, one should make use of the CIELAB ΔE measure. (However, ΔXYZ errors may have more obvious meaning to rendering practitioners, so we report both.) Under highly controlled viewing conditions a CIELAB ΔE error value of 1 correlates with a single just noticeable difference in colour. In complex imagery an error up to 3 is not significant.²⁰ CIELAB units are formed by using the ratios of tristimulus XYZ values to the XYZ triple for the illuminant itself. Then the luminance channel L is basically 100 times Y raised to the power $1/3$: $L = 116Y^{1/3} - 16$. Colour content is defined in terms of differences of one-third powers of the X and Z ratios from $Y^{1/3}$. In terms of ΔE units, the mean error is 0.92, and the median is 0.42. Clearly, the basis does very well overall. However, the 95 percentile ΔE figure goes up to 3.04 units, meaning that some errors may be discernible (although not in actual rendered images we tried — see § 3.E.1). In comparison, a using a simple set of block filters on the same wavelength intervals produces figures of 7.15 and 6.74 mean and median ΔE errors. As well, the 95 percentile ΔE figure goes up to 14.72 units, clearly unacceptable. Overall, we should use the sharpening method. Moreover, a basis specialized for a particular set of lights and surfaces will do even better.

Note that this type of test is a good deal more stringent than simply comparing colour values in a rendered *image*, since the ΔE value includes magnitude. In this colour signal test, every colour is at the *maximum* magnitude possible, so the errors evaluated are also at maximum magnitude.

B. Recovering Spectra

In general, suppose we use a $31 \times p$ set of basis vectors \mathbf{Q} , with p about 5 to 7. Suppose light currently has a spectrum given in sampled terms by the 31-vector \underline{C} . Now suppose this colour signal interacts with the next reflection or transmission function \underline{S} . Then in matrix terms, the coefficients in the sharpened basis $\widetilde{\mathbf{Q}}$ for colour signal \underline{C} are equal to the p -vector \mathbf{c} where

$$\mathbf{c} = \widetilde{\mathbf{Q}}^T \underline{C}. \quad (18)$$

Similarly, coefficients for the next spectral function \underline{S} are given by

$$\mathbf{s} = \widetilde{\mathbf{Q}}^T \underline{S}. \quad (19)$$

Note that the new basis set $\widetilde{\mathbf{Q}}$ is *not* orthogonal, so that the best approximation of a spectrum is not given by simply multiplying these weights by the basis vectors. Instead, the best least squares approximation of 31-vector \underline{C} is given by the Moore-Penrose pseudoinverse of $\widetilde{\mathbf{Q}}$ acting on these weights:

$$\widehat{\underline{C}} = \widetilde{\mathbf{Q}}^+ \mathbf{c} \quad (20)$$

where

$$\widetilde{\mathbf{Q}}^+ \equiv \widetilde{\mathbf{Q}} (\widetilde{\mathbf{Q}}^T \widetilde{\mathbf{Q}})^{-1}. \quad (21)$$

The Moore-Penrose pseudoinverse is a $31 \times p$ matrix.

Similarly, the best approximation of 31-vector \underline{S} is given by

$$\widehat{\underline{S}} = \widetilde{\mathbf{Q}}^+ \mathbf{s}. \quad (22)$$

Then the spectral Factor Model reconstruction of weights for the spectrum,

after the next interaction, is simply the product of weights:

$$v_k \equiv c_k s_k \quad , \quad k = 1..p . \quad (23)$$

Written in terms of vectors \mathbf{c} and \mathbf{s} , this component-wise multiplication of p -vectors could be written in algebraic terms as

$$\mathbf{v} = \text{diag}(\mathbf{c})\mathbf{s} . \quad (24)$$

Thus the spectral Factor Model reconstruction of a spectral function after the next interaction is the result of the pseudoinverse acting on the above simple component-wise product vector of p -vectors \mathbf{c} and \mathbf{s} :

$$\text{spectrum} = \widetilde{\mathbf{Q}}^+ \mathbf{v} \quad , \quad v_k = c_k s_k . \quad (25)$$

Note that in fact no spectrum need actually be calculated unless full spectral information is necessary. For any interaction where this is not the case, simple multiplication of the p -vector “colours” may be carried out. Otherwise, the precomputed $31 \times p$ matrix $\widetilde{\mathbf{Q}}^+$ may be employed.

As well, note that in (18) we are using the sharpened basis set as a set of filters: it would be usual to define weights \mathbf{c} in terms of the pseudoinverse $\widetilde{\mathbf{Q}}^+$ applied to the spectrum. But that would destroy the parallel with spectral sharpening; while the direction we go makes no difference when modelling individual spectra, since the columns of $\widetilde{\mathbf{Q}}^+$ are not as sharp we should use the definition as stated in order to reduce spectral multiplication to a simple weight multiplication.

The method presented gives a speedup factor of about 5, and no matter how fast hardware may become this speedup is important. Compared to a method using principal com-

ponent vectors without a sharpening step,⁴ instead of each spectral interaction consisting of an $p \times p$ matrix multiply, for p -dimensional basis, we have just p multiplies; the savings compared to Peercy’s method⁴ is therefore p multiplies compared to p^2 multiplies and $p(p - 1)$ additions.

C. Recovering RGB

After all rendering (e.g., raytracing or raycasting) is accomplished, we may reconstruct RGBs trivially from p -dimensional weight vectors via a $3 \times p$ matrix. For suppose colours are generated via a matrix operation taking spectrum into 3-vector:

$$\mathbf{b}^T = \underline{\mathbf{C}}^T \mathbf{R} , \quad (26)$$

with \mathbf{R} an appropriate 31×3 colour-matching matrix. The RGB colour can be determined directly in terms of weights by a matrix transform

$$\mathbf{b} = \mathbf{R}^T \underline{\mathbf{C}} = (\mathbf{R}^T \widetilde{\mathbf{Q}}^+) \mathbf{c} \equiv \mathbf{U} \mathbf{c} , \quad (27)$$

where \mathbf{U} is $3 \times p$.

Thus to generate RGB colour for a given product spectrum, we need never carry out a spectral reconstruction.

Note that we might expect a simpler method using a set of block filters to do quite well in modelling colours for light bounced off surfaces, since block basis functions have no overlap at all and one would expect there to be some tradeoff between this asset and the optimality of the suggested basis set. But in fact using the method presented here, if we consider the 1190 colour signals arising from our 170 objects times 7 illuminants, the spectral Factor Model error using the proposed method gives mean and median errors of 2.21 and 1.14 ΔE

units, whereas a block filter method gives mean and median errors of 5.50 and 4.76 ΔE units, substantially higher values.

D. Examples

Since we might like to use a *fixed* basis set, re-used for any rendering application, we should consider testing our basis against other reflectances and colour signals, ones that were not used in forming the basis vectors. As well, this might make the most sense in applying the present model to vision-, as opposed to graphics-oriented problems.

As a concrete example of the spectral Factor Model calculation for spectra, suppose we form a colour signal using the “light skin” patch (# 2) of the Macbeth ColorChecker, which is another set of standard reflectances.²¹ Using standard illuminant D65, we arrive at the colour signal (solid curve) of Fig. 2(a). The best least squares approximation using five basis vectors is also shown, dotted.

If we apply eq. (25) for the spectral Factor Model product spectrum, as derived by simply multiplying weights, then we arrive at the dashed curve in Fig. 2(a). Clearly, little accuracy has been lost over using the best least squares fit, which is the best we can do given our basis dimensionality.

Let us now model a second bounce, this time from the Macbeth patch (# 4) approximating the spectral characteristics of foliage. Here, we are considering a surface graphics paradigm, where we wish to investigate modelling errors as the light travels through the surface model and is gradually extinguished. Fig. 2(b) shows that again we have done well enough. The two-bounce spectrum calculated according to the spectral factor model approximates the best least squares approximation. Finally, a third bounce off the Macbeth patch (# 3) for “blue sky” gives approximations as in Fig. 2(c).

For the first, second, and third bounces, the RMS error for the spectral Factor Model

approximation for *spectral components* is 7.28%, 10.14%, and 16.82%, which compare well to the best RMS errors possible in a 5–dimensional model, which are 6.50%, 8.57%, and 10.98%.

In terms of XYZ, however, errors are much less. First, we consider raw XYZ vector differences from the correct XYZ, and then phrase differences in terms of perceptual error. For the first, second, and third bounce, the correct XYZ triple (using sensor curves that each sums to 10.67) is (400.7, 374.6, 288.5), then (45.3, 49.3 17.5), and finally (8.7, 10.3 5.8). The best we can do gives raw error norms of 10.99, 0.32, and 0.14 for the first, second, and third bounce, whereas the spectral Factor Model approximation of the spectrum yields XYZ norm errors of 23.64, 3.83, and 1.74.

Of course error for XYZ responses does not have a direct perceptual meaning. However, in terms of colour difference assessment if we map XYZ to CIE Lab coordinates¹⁶ (a 3-dimensional non-linear transform) then Lab does have perceptual significance. Here we observe ΔE s of just 2.35, 3.82 and 0.84 (compared to 1.69, 0.28 and 0.28 for the full 5D linear model). Two remarks are useful here. First we notice that the error diminishes for the third bounce. This is due to the fact that as light bounces, less and less light is reflected and so the spectral power distribution is attenuated — and the ΔE measure includes magnitude. (The increase for the second bounce is likely due to the approximation error being greater than this attenuation.) In the limit after infinite bounces no light is reflected and so in the limit we have a zero error. Second, that in the context of complex images a ΔE of 3 or even 5 is not noticeable to an observer.²⁰ This small test indicates that the method can produce reasonably good results.

As well, we could in fact do a good deal better by simply creating a “specialized” basis for colour signals formed from just a target set of illuminants and reflectances. In graphics, this could well be a reasonable approach if we wished to use particular lights such as sodium

lamps, say, that have a spiky component. For example, using a basis set created from only fluorescent light “F2”¹⁶ and the 24 Macbeth reflectances, then a 5D sharpened linear model gives ΔE errors of 3.71, 3.74, and 0.57 ΔE units over 1, 2, and 3 bounces, compared to 0.71, 0.20, and 0.14 units for the best finite-dimensional model available. This is certainly acceptable performance for the Factor Model of spectra.

However the simple expedient of utilizing a single basis for any application, appealing in its simplicity, is also quite workable.

E. Multi-reflection results

1. Random reflections

Now let us consider the behaviour of the algorithm in modelling spectra and colours for a more exhaustive test. Let us use the 170 reflectances and 7 illuminants to first generate all possible colour signals, and then model them using the spectral Factor Model method we have set out, and compare to the best that can be done using 5 basis vectors. The results are that, for the output XYZ vectors, the best approximation yields a median percentage error of 0.99%, with error defined as $(actual - approximation) / \|actual\|$. The spectral Factor Model version, modelling spectra by multiplication of sharpened basis coefficients and then reconstituting spectra, yields a median XYZ error of 3.6%. This is an acceptable error level.

In general, we are interested in how random bounces produce errors, indexed both in terms of how many bounces, and as well in terms of how many dimensions we use in our finite dimensional model. Table 1 shows how the *best* approximation performs, using a given basis set dimensionality. Let us perform one test for each of 1 to 5 random bounces, starting from each material in our set of 170 reflectances, and with an illuminant randomly drawn from our set of 7 lights and accumulating the colour signal as bounces proceed. Further bounces are calculated from randomly drawn reflectances from the set of 170. Here, we are

considering a volume graphics paradigm, where we wish to add up light coming through a volume, with light interacting as it comes through each new material, always accumulating light along an “eye ray”. That is, we first create a colour signal starting from a random illuminant and bouncing from a random reflectance. This forms the main contribution into a virtual camera. We then keep bouncing, each time adding more contributions to the signal received at the virtual camera. We consider basis set dimensionality from 3 to 9.

Median CIE ΔE values are tabulated, measuring the error from the XYZ triple derived from the full, correct spectrum to that derived from the spectrum produced by our spectral Factor Model rule in eq. (23). The top of Table 1 shows ΔE values for the best available spectral model, for a given dimensionality — after all, for any algorithm based on a Finite-Dimensional Model, we cannot hope to do better than the underlying dimensional reduction allows, and so it is important to compare to how well we could do at base. The second part of the Table gives error values for the proposed model. As generally expected, the spectral Factor Model median errors are worse than the “best” errors.

We see that, again as expected, the proposed model does best for higher dimensionality. Overall, naturally we could do better, using a Finite-Dimensional basis, if we in fact knew the colour signal and then applied the basis. But the spectral Factor Model method set out does well enough, for dimensionality 7, and not poorly even for dimensionality 5. Hence, we can carry out graphics using roughly twice the number of “colours”, but with much greater accuracy in terms of spectral information than simply using RGB.

As well, the third part of Table 1 gives ΔE performance using a simple, 3-component colour component-wise multiplication version of Computer Graphics. This model simply multiplies colour and is therefore independent of basis dimension. For dimension 7 or higher, the spectral Factor Model always outperforms simple colour multiplication, considering only overall statistics and not the large errors that will occur for special cases for the simple RGB

method.

In fact, the simple RGB-multiplication model does work reasonably well, over many cases. However, sometimes simple RGB multiplies will result in colours that are completely wrong (see § 3.E.2 below on the use of metamers as an application of the fact that we have spectra available).

Borges³ showed that if we define an error measure for an RGB Factor Model of tristimulus values of products of functions f and g (light and reflectance, say) via $\varepsilon(f, g) = \int (f(\lambda)g(\lambda)q(\lambda)d\lambda - \int f(\lambda)q(\lambda)d\lambda \int g(\lambda)q(\lambda)d\lambda)$ (where sensor q is normalized so that it integrates to unity) then an error bound exists $[\varepsilon(f, g)]^2 \leq \varepsilon(f, f)\varepsilon(g, g)$. That is, a low error bound depends on the RGB Factor Model working well for the product of these functions with themselves. So Table 1 is an empirical validation of Borges' theoretical result: the error bound is reasonably small much of the time.

But we note that in fact the same theorem applies if we make use of our sharpened basis functions, in place of sensors q , and in a higher dimension. So the information in Table 1 is that, if we wished to do best, in a low dimension, with respect to the human visual system then our sharpened basis is indeed not the best choice since it covers the whole spectrum uniformly and the human visual system does not. But by its definition we are in fact trying to do best not with respect to colour per se, but with respect to an error bound over the whole spectrum — we are trying to obtain the best representation for a spectrum, for use in tasks for which a complete spectrum is required (see, e.g.,²²). And for any dimensionality our choice will do best for that objective.

Note that in respect of Table 1, of course the first bounce colour is usually the most important, since it will contribute the most in magnitude. Nevertheless the other bounces are important as well, since these can substantively contribute light when summed over many bounces — just as in the radiosity method of graphics, or in fact in the actual lighting

conditions of most rooms. A good deal of the lighting in a room with light-coloured walls comes from the sum of a series of bounces.

Table 2 show CIELAB errors at the 90-percentile level for the same experiment as above. Generally, ΔE errors for the spectral Factor Model with dimensionality of 7 or above are acceptable. And in fact, colour itself and the ΔE measure are arguably not the right error metric for what we are trying to accomplish, since we need an accurate spectrum more than an accurate colour, for the tasks at hand (see the next section), and the naive model of multiplying RGBs fails dismally for these however reasonable errors are in general for the simple approach.

2. Using spectral models for visualisation

Importantly, we can make use of the spectra the present method generates to our advantage in certain types of modelling and visualization (where the RGB model alone would not suffice). To begin with, consider the RMS errors for whole spectra, for random lighting and reflectances. Table 3 shows that we do remarkably well using the spectral Factor Model, even accumulating over many bounces, and not too much worse than the best we could possibly do. Thus the spectral Factor Model produces a very reasonable approximation of whole spectra for use in graphics or vision.

Metamers present an interesting problem for graphics rendering. A pair of metameric surfaces have the property that they look the same under one light but different under a second. Of course, an RGB model for rendering cannot predict metameric effects — using non-spectral graphics, if RGBs are the same (or different) under one light, they must be the same (or different) under another light. The RGB mapping is bijective. This is not true for the spectral Factor Model. We carried out the following experiment to see how Factor Model spectral rendering and the RGB model fare in the face of metamerism.

Consider the three spectral pairs shown in Fig. 3. Here, an original and a metameric spectrum are shown as a solid and a dashed curve. Each pair produces the same XYZ triple under illuminant D65, but each pair yields widely different colour values under tungsten lighting, represented by standard illuminant A (these curves are from Ref.¹⁶ ; and see Ref.²³ for colour images of these metamers). We would like to see how the usual RGB-multiplication model of Computer Graphics does on these metamers under illumination change. Since, in the simple model, colour is formed from the product of surface colour and illuminant colour, one necessarily has that if surfaces match under one light, they match under any other light. And that is precisely the incorrect behaviour in these cases.

Applying the 3-component RGB Factor Model to the separate XYZ vectors for surface and illuminant, under D65 the XYZ pairs do not actually match precisely: ΔE values for differences between the pairs are 6.41, 4.30, and 4.14 units. I.e., full spectrum colour signals match (with $\Delta E=0$), but the RGB Factor Model colour does not match. However, the colour is close nonetheless. Now changing the illuminant to tungsten, we again find that ΔE values for the RGB Factor Model of colour are small: 3.69, 2.12, and 2.18 units. But for the full spectrum version, these values should actually be 34.51, 24.34, and 22.31 units. In other words, the simple RGB model fails dramatically.

In Fig. 4 the spectral Factor Model derived approximation of the colour signal under tungsten lighting, for an N=7 basis, is shown dashed, along with a solid curve for the correct spectrum. The ΔE values for the differences between correct and approximate spectra have median value 7.42 units. Whereas the usual RGB colour model is not only completely wrong and does not yield a spectral approximation, the present method does reasonably well and also calculates a spectrum.

Fig. 5 shows how these surfaces appear in a standard surface graphics rendering system.²⁴ Here the three metamer pairs are first shown under light D65 as three sets of contiguous

spheres, with a horizontal plane of Macbeth colour patch #2 reflecting the sphere pairs. Rendering uses a standard raytracing approach with up to 30 scattering events per ray. The top-left image uses full-spectrum calculations followed by computation of XYZ values, as a form of ground truth. In comparison, the Factor Model approach to approximating spectra is shown in the central image, and the RGB-graphics version of the image is shown on the right. The bottom row shows the same scene under standard illuminant A. The RGB-graphics version in the right-hand column is, of course, completely wrong since surfaces that match under one light will again match under another light. But the full-spectrum and Factor Model versions agree. Notice that since raytracing is employed, in fact these images show not just a single bounce, but in fact multiple bounces of light.

Table 4 shows how a “specialized” basis performs. In graphics, we may wish to use only a few spectra that we have designed, and use only a few lights. Since we are interested in the metameric reflectances, let us use these to form colour signals, without regard to their metameric pair structure. Here we make use of standard illuminants D65 and A to synthesize colour signals for the six metamer spectra (three pairs) used in Fig. 5 (although in fact we did not use a specialized basis to create Fig. 5 — we used a general-purpose basis). Table 4 gives errors, for the two lights used and over 6 bounces, using a specialized basis with dimension 7. In fact, these metamer reflectances are difficult to capture in a low-dimensional basis in that generally some are not very smooth. In comparison to ordinary RGB multiplication the spectral Factor Model does very well, with CIELAB errors about 1/3 the size, on average. Ordinary colour multiplication fails spectacularly for standard light A, under which metamers are not supposed to match.

Note that even in a sharpened colour space a simple RGB diagonal colour change to correct an overall colour cast stemming from the illumination^{25,26} cannot accurately deal with spectral effects, in contradistinction to the method presented here which indeed can.

In particular, if two patches match under one light, then using the method of white-point correction using a 3×3 matrix,^{25,26} in any colour space, to transform colour values to those under another illuminant will still produce matching patches, and this will be wrong for metameric surfaces.

To consider spectral error, Table 5 shows how the spectral Factor Model fares in terms of RMS whole-spectrum percentage error, again for the specialized basis aimed at the metamer reflectances (with CIELAB errors given above in Table 4). Comparing to Table 3, we note that indeed we do worse than for the relatively smoother non-metameric reflectance set. Nevertheless, one of the strengths of the present method is that it can indeed handle a situation in which one uses a specialized basis, as opposed to point-sampling methods (e.g.,²⁷) which necessarily must fail for some spiky source or absorption spectra.

Fig. 6 shows the accumulated reconstituted spectra for the colour signals with errors as in Tables 4 and 5. We see that even for challenging metameric spectra and changing light we still have a relatively small spectral error.

It is evident that the best FDM model significantly outperforms the spectral Factor Model approach. Notice that in Table 4 for one and two bounces the spectral Factor Model delivers an error rate of about $6 \Delta E$ (just about imperceptible in complex scenes) compared with 18 for the RGB model (very noticeable). Moreover, as the number of bounces increases, the error rate for the spectral Factor Model is between one third and one half of that for the RGB model. As a final point we remark that the error rates achieved for the best FDM model require the spectral calculation of the colour signal which is then projected to the lower-dimensional basis. That is, Table 4 values for “Best FDM” illustrate performance that can only be achieved by full spectral rendering, which is what we are trying to avoid.

3. Spectral models in volume visualization navigation

Perhaps the most appropriate use for the method within Graphics would likely be in scientific visualization using Volume Graphics (see Ref.²⁸). Here, a full-spectrum approach can well make use of the properties of metamers to enable the disclosure or hiding of 3D features when rendering such data sets. As we saw in Fig. 5, RGB graphics cannot form a vehicle for effects that require full spectra. In Fig. 7 we show a volume rendering result derived from MRI slice images for biological data. In the first instance, in Fig. 7(a) we render the visualization using a standard illuminant D65, with materials assigned spectral reflectance values in a volume-rendering transfer function such that most structures in the image are mapped to the same or a similar colour: the reflectances are metamers under D65. In the second instance, in Fig. 7(b), we instead use tungsten lighting to illuminate, with the result that most structures fade into black, and a single structure (the frog’s stomach) is prominent. Only by spectral design can such effects be created. The interesting feature of this approach is that the lighting can be inserted at any point in the rendering pipeline; in particular, here we render the visualization images as a *post-illumination* step, after all raytracing is completed. This means that the user can interact with the visualization in *real time* (cf. Ref.²⁸). These images were created using a general-purpose basis, not a specialized one.

4. Conclusions

We have set out a new method for spectral multiplication based on sharpening of a set of basis functions for colour signals. Since sharpening promotes a spectral Factor Model, the new method gives a good approximation of spectra, even under multiple bounces of light. As well, but not included in this paper, we have determined that a full-spectrum solution for radiosity²⁹ can be phrased in terms of a lower-dimension statement of the problem, with resulting savings. Naturally, false colour rendering of medical volume data³⁰ in fact has

nothing to do with accurate colour since such colour is unphysical. However, we could not have achieved real-time feature highlighting and obscuring without the facility of spectral response via changing a lighting slider²⁸). Moreover, the present method does open up the possibility actually rendering medical structures correctly, using spectral measurements of tissues.

Computer Graphics is certainly the application most readily amenable to full-spectrum calculations, using the present method. But other applications, in Computer Vision, for example, can also be addressed using the mechanism set out here and we intend to pursue these other applications elsewhere.

5. Acknowledgements

The authors are indebted to the Natural Sciences and Engineering Research Council of Canada (M.S.D) and the Hewlett Packard corporation (G.D.F.) for funding. As well, we thank Mr. Steven Kiltbau for help in producing the spheres raytracing images and Mr. Stephen Bergner and Dr. Torsten Möller for the volume-rendering images of frog data. Finally, the reviewers and the Topical Editor provided invaluable suggestions that greatly improved the paper.

References

1. G. Healey and D. Slater. Global color constancy: Recognition of objects by use of illumination invariant properties of color distributions. *J. Opt. Soc. Am. A*, 11:3003–3010, 1994.
2. Eds.: P. Robertson and J. Shönhut. Special Issue on Color in Computer Graphics. In *Computer Graphics and Applications*, volume 19, 4: July/August, pages 18–67. IEEE, 1999.

3. C.F. Borges. Trichromatic approximation method for surface illumination. *J. Opt. Soc. Am. A*, 8:1319–1323, 1991.
4. M.S. Peercy. Linear color representations for full spectral rendering. In *Computer Graphics (SIGGRAPH '93)*, number 2, pages 191–198, 1993.
5. G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *J. Opt. Soc. Am. A*, 11(5):1553–1563, May 1994.
6. L. Arend and A. Reeves. Simultaneous color constancy. *J. Opt. Soc. Am. A*, 3:1743–1751, 1986.
7. L.T. Maloney and B.A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *J. Opt. Soc. Am. A*, 3:29–33, 1986.
8. D.A. Forsyth. A novel approach to color constancy. In *Int. Conf. on Computer Vision '88*, pages 9–18, 1988.
9. D. H. Brainard, B. A. Wandell, and E.-J. Chichilnisky. Color constancy: From physics to appearance. *Cur. Dir. Psychol. Sci.*, 2:165–170, 1993.
10. D.H. Foster and S.M.C. Nascimento. Relational colour constancy from invariant cone-excitation ratios. *Proc. Roy. Soc. London B*, 257:115–121, 1994.
11. D.H. Brainard and W.T. Freeman. Bayesian color constancy. *J. Opt. Soc. Am. A*, 14:1393–1411, 1997.
12. G. Buchsbaum. Color signal coding: Color vision and color television. *Color Research and Application*, 12:266–269, 1987.
13. P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.
14. M.S. Drew and G.D. Finlayson. Spectral sharpening with positivity. *J. Opt. Soc. Am. A*, 17:1361–1370, 2000.
15. M.S. Drew and G.D. Finlayson. Representation of colour in a colour display system. UK Patent Application No. 0206916.9. Under review, British Patent Office, March 23, 2002.

16. G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. Wiley, New York, 2nd edition, 1982.
17. M.J. Vrhel, R. Gershon, and L.S. Iwan. Measurement and analysis of object reflectance spectra. *Color Research and Application*, 19:4–9, 1994.
18. M.S. Drew and B.V. Funt. Natural metamers. *CVGIP:Image Understanding*, 56:139–151, 1992.
19. D.B. Judd, D.L. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *J. Opt. Soc. Am.*, 54:1031–1040, August 1964.
20. M. Stokes, M.D. Fairchild, and R.S. Berns. Precision requirements for digital color reproduction. *ACM Trans. Graphics*, 11(4):406–422, 1992.
21. C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart. *J. App. Photog. Eng.*, 2:95–99, 1976.
22. Y. Sun, F.D. Fracchia, M.S. Drew, and T.W. Calvert. Rendering iridescent colors of optical disks. In *11th Eurographics Workshop on Rendering*, pages 341–352. Eurographics/ACM, June 26-28, Brno, Czech Republic 2000.
23. G.M. Johnson and M.D. Fairchild. Full-spectral color calculations in realistic image synthesis. *Computer Graphics and Applications*, 19(4: July/August):47–53, 1999.
24. A colour version of all figures may be found at <http://www.cs.sfu.ca/~mark/ftp/Josa03> .
25. G.D. Finlayson. *Coefficient Color Constancy*. PhD thesis, Simon Fraser University, 1995.
26. G. Ward and E. Eydelberg-Vileshin. Picture perfect RGB rendering using spectral pre-filtering and sharp color primaries. In *13th Eurographics Workshop on Rendering*. Eurographics/ACM, June 26-28, Pisa, Italy 2002.
27. G.W. Meyer. Wavelength selection for synthetic image generation. *Comp. Vision, Graphics, and Image Proc.*, 41:57–79, 1988.

28. S. Bergner, T. Möller, M.S. Drew, and G.D. Finlayson. Interactive spectral volume rendering. In *IEEE Visualization*, pages 101–108. IEEE, Boston 2002. Selected for proceedings cover.
29. C.M. Goral, K.E. Torrance, D.P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics*, 18:213–222, 1984.
30. H.J. Noordmans, H.T.M. van der Voort, and A.W.M. Smeulders. Spectral volume rendering. *IEEE Trans. Visualization and Comp. Graphics*, 6:196–207, 2000.

	Reflections				
	1	2	3	4	5
Dimension	Best FDM				
3	3.06	3.24	3.62	4.27	4.86
5	0.95	0.96	1.12	1.34	1.70
7	0.44	0.55	0.70	1.02	1.26
9	0.13	0.16	0.19	0.21	0.24
Dimension	Factor Model				
3	3.92	4.39	5.48	7.04	8.40
5	2.51	2.92	3.58	4.35	6.25
7	1.13	1.31	1.44	1.78	2.09
9	0.88	1.04	1.15	1.25	1.81
	Multiply RGB				
	1.76	1.99	2.27	2.67	3.16

Table 1. Median CIELAB ΔE values for 170 Object reflectances comparing the best spectral approximation available using a Finite-Dimensional basis set of dimension N with the spectral Factor Model approach, accumulating over 1 to 5 recursive reflections. The illuminant in each case, and the further reflectance functions, are drawn randomly. The last line compares to a simple multiplication of XYZ values.

	Reflections				
	1	2	3	4	5
Dimension	Best FDM				
3	3.31	3.66	3.85	4.57	5.19
5	0.98	1.03	1.21	1.36	1.77
7	0.46	0.63	0.81	1.13	1.37
9	0.14	0.18	0.21	0.22	0.25
Dimension	Factor Model				
3	4.12	4.95	5.75	7.44	9.45
5	2.67	3.06	3.75	4.74	6.96
7	1.19	1.40	1.48	1.83	2.21
9	0.92	1.18	1.26	1.31	1.94
	Multiply RGB				
	1.86	2.02	2.35	2.82	3.58

Table 2. Ninetieth percentile level for CIELAB ΔE values over 170 Object reflectances with randomly chosen lights.

	Reflections				
	1	2	3	4	5
Dimension	Best FDM (%)				
3	9.11	9.28	9.76	10.42	11.14
5	4.80	4.83	4.75	4.80	5.26
7	3.03	3.29	3.77	4.28	4.83
9	2.26	2.32	2.64	2.77	3.17
Dimension	Factor Model (%)				
3	10.35	10.59	10.97	11.65	13.10
5	7.59	7.50	7.35	7.90	8.61
7	5.17	5.61	5.98	6.75	7.96
9	5.04	5.68	6.21	6.58	7.28

Table 3. RMS percentage errors for whole spectra over 170 Object reflectances with randomly chosen lights.

	Reflections					
	1	2	3	4	5	6
	Best FDM (dim=7)					
D65	2.20	1.02	4.36	2.89	6.60	5.41
A	1.03	2.39	0.86	0.62	3.65	3.86
	Factor Model (dim=7)					
D65	1.69	1.18	4.72	3.40	9.23	4.51
A	6.22	6.16	9.48	10.39	17.39	20.28
	Multiply RGB					
D65	3.95	4.20	8.22	10.08	17.50	15.99
A	18.24	18.94	27.83	29.90	43.92	37.65

Table 4. CIELAB errors using specialized 7D basis from lights D65 and A applied to the metamer reflectances, over both lights and 6 bounces, accumulating the signal over bounces.

	Reflections					
	1	2	3	4	5	6
Best FDM (%) (dim=7)						
D65	3.01	2.64	5.65	5.28	8.88	9.09
A	2.48	3.95	4.41	4.04	8.71	9.11
Factor Model (dim=7)						
D65	4.20	3.84	7.40	6.44	11.70	11.37
A	17.00	21.90	14.01	16.45	14.68	16.63

Table 5. RMS whole-spectrum percentage errors for lights D65 and A applied to the metamer reflectances, over both lights and 6 bounces, accumulating the signal over bounces.

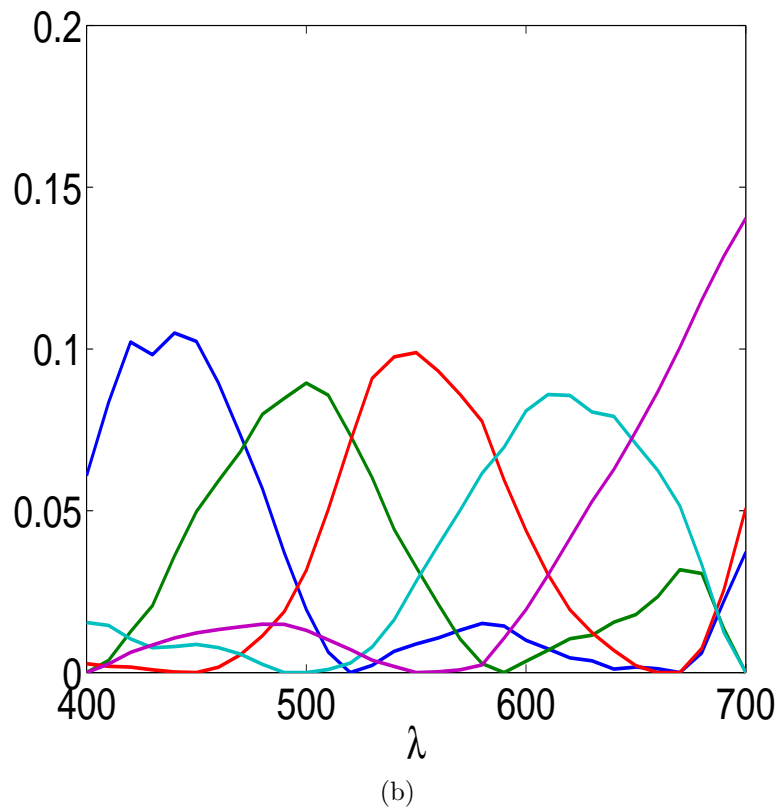
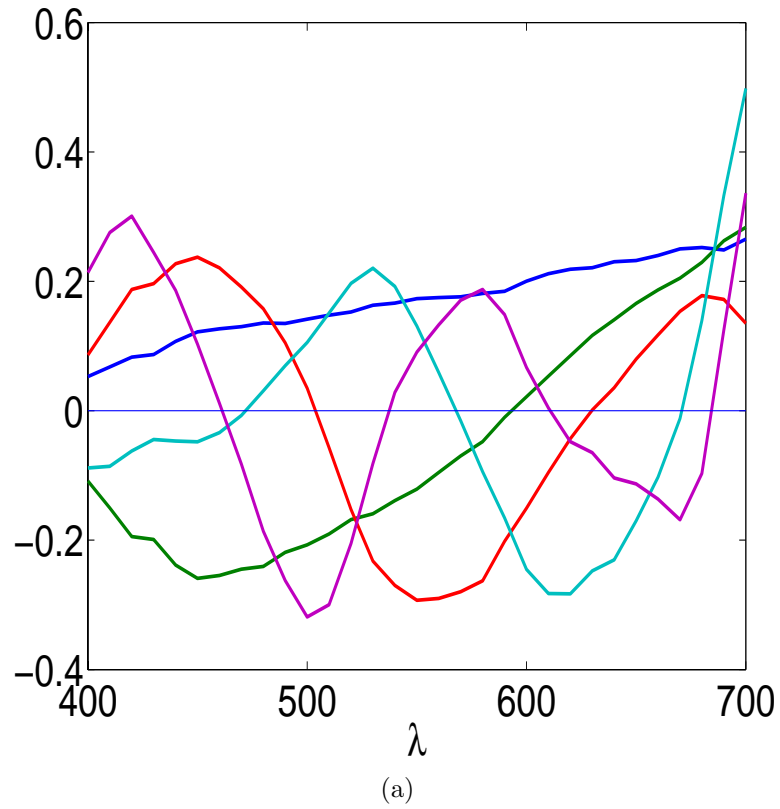
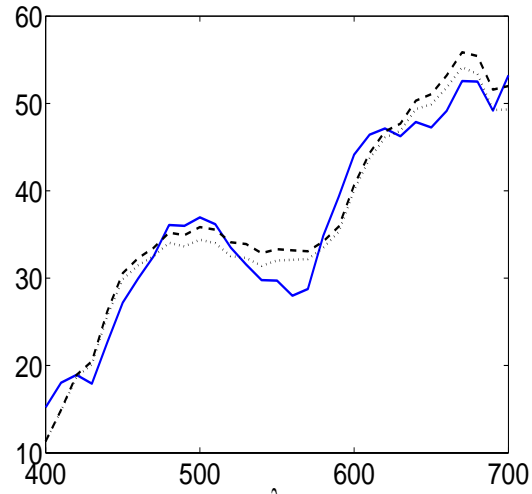
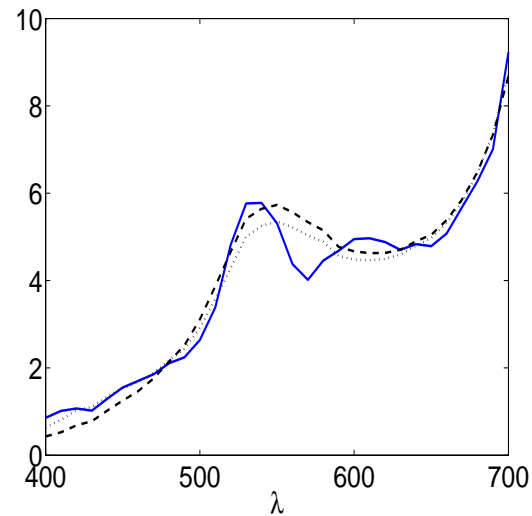


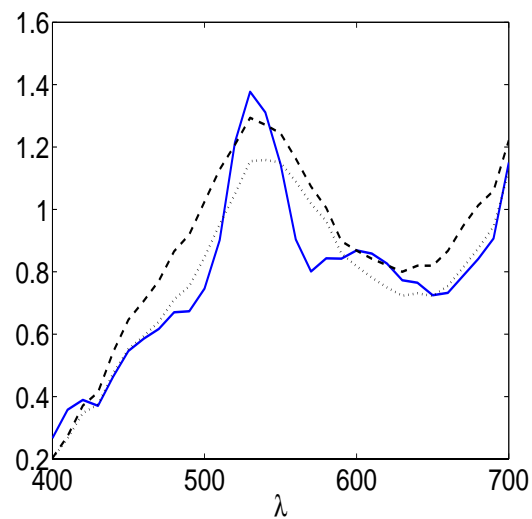
Fig. 1. (a): First five SVD vectors. (b): Sharpened version of basis vectors.



(a)



(b)



(c)

Fig. 2. (a): One bounce. Exact: solid line; best least squares: dotted line; spectral Factor Model: dashed line. (b): Two bounces. (c): Three bounces.

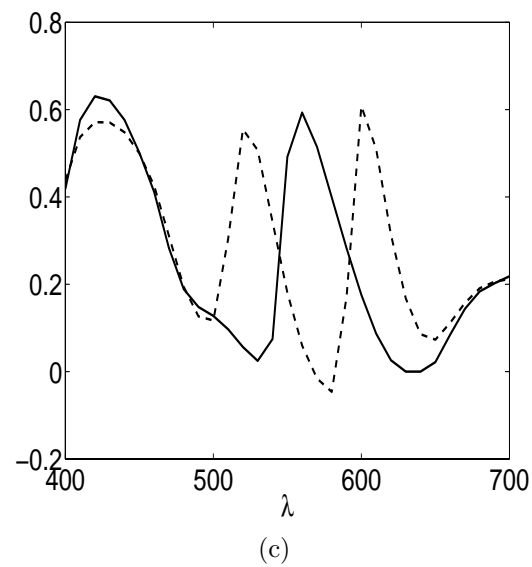
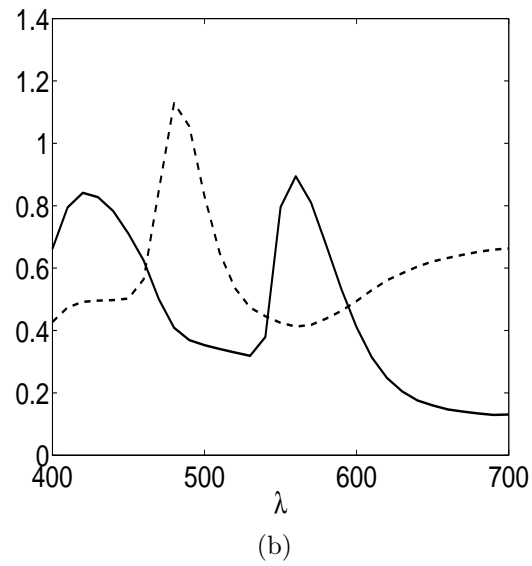
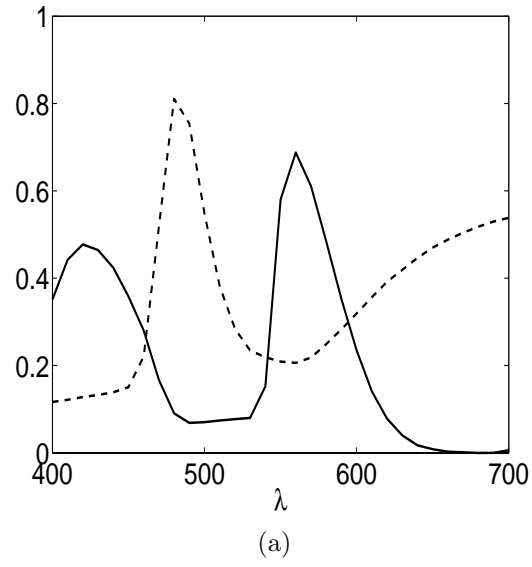
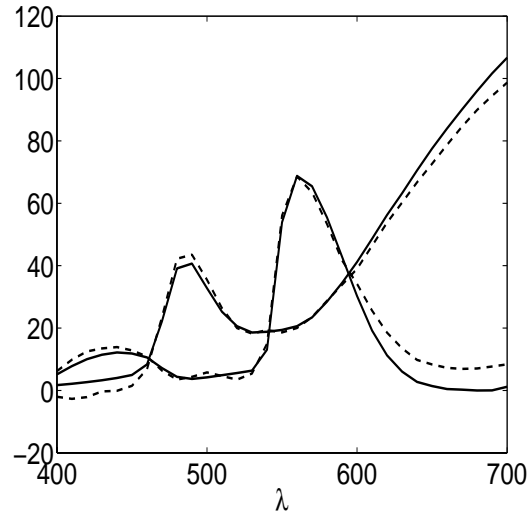
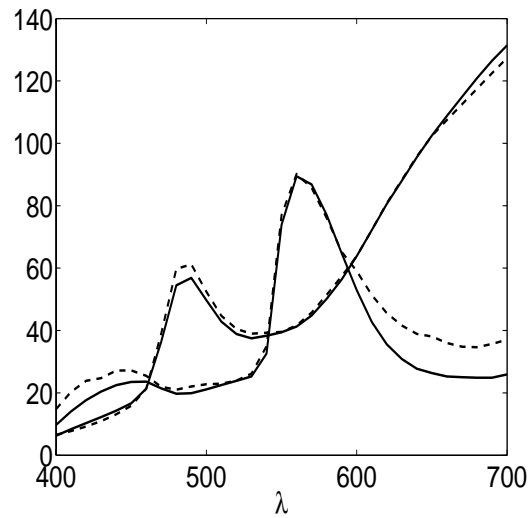


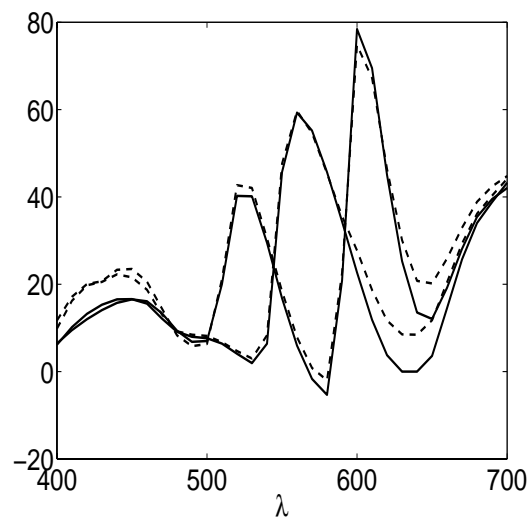
Fig. 3. Three pairs of metameric surface spectral reflectance functions.



(a)



(b)



(c)

Fig. 4. Colour signals for the three metameric pairs, under tungsten lighting (solid) with the spectral Factor Model approximation (dashed).

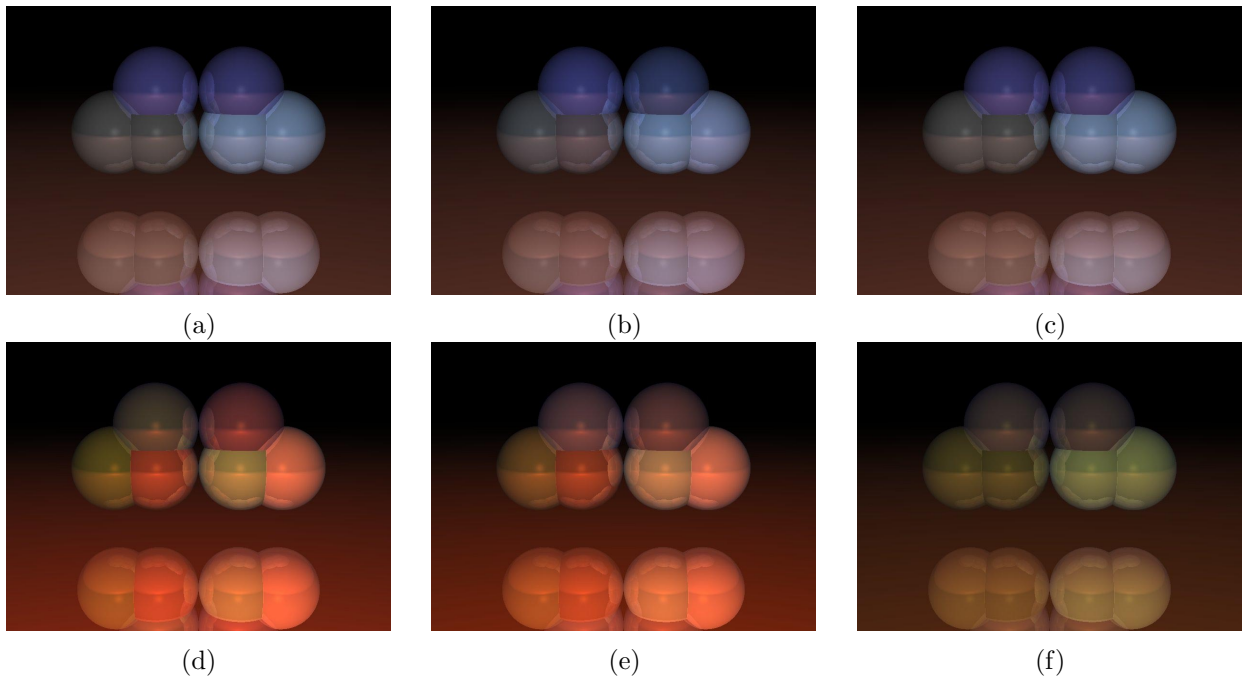


Fig. 5. Spheres painted with metameric surface reflectance functions. Top row is under D65, bottom row is under illuminant A. Left column uses full-spectrum reflectances and illuminants in a full-spectrum 31-component raytracer, middle column shows results using the present spectral Factor Model approach, and right-hand column shows results using standard 3-component graphics. (Colour versions of figures can be seen at <http://www.cs.sfu.ca/~mark/ftp/Josa03> .)

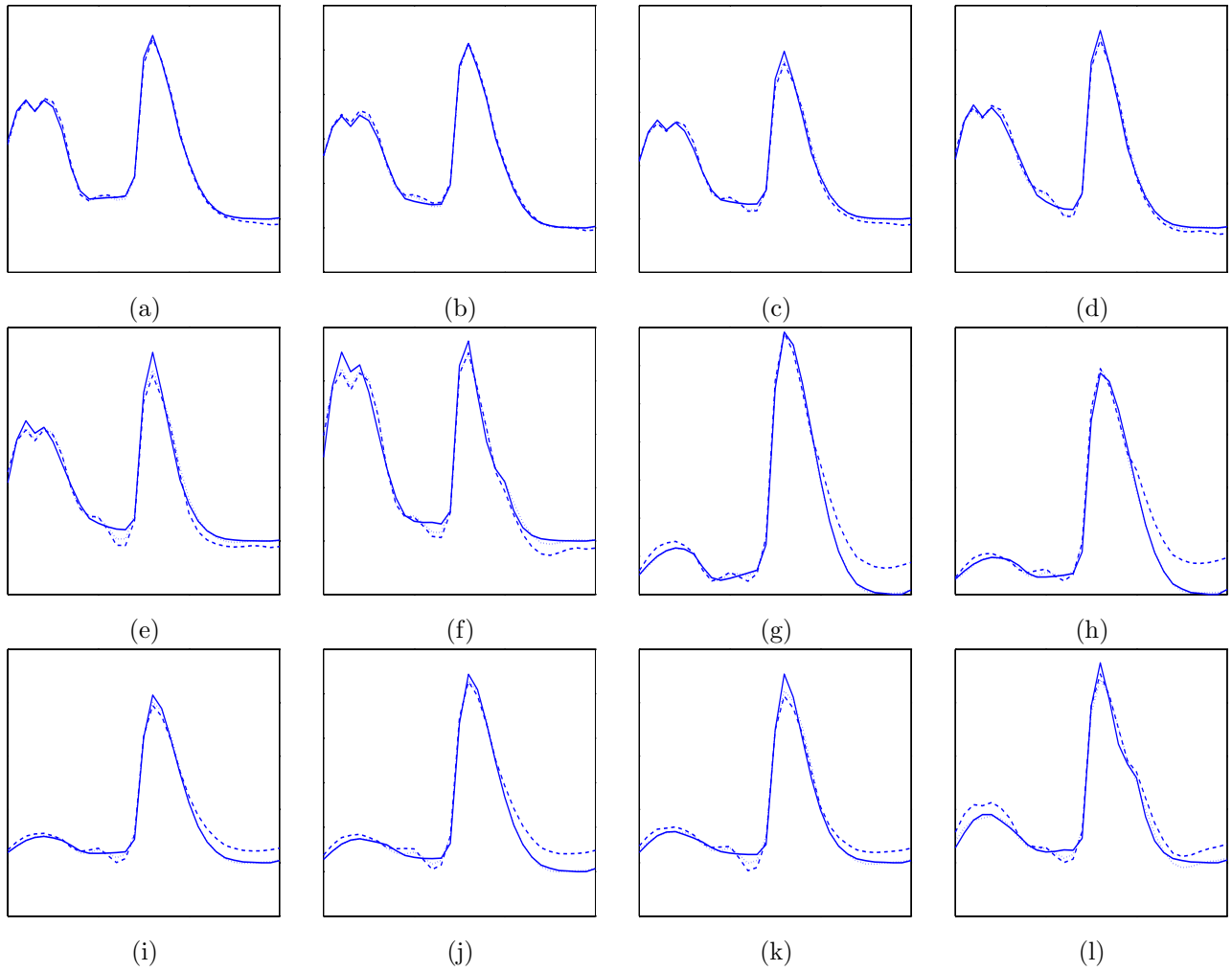
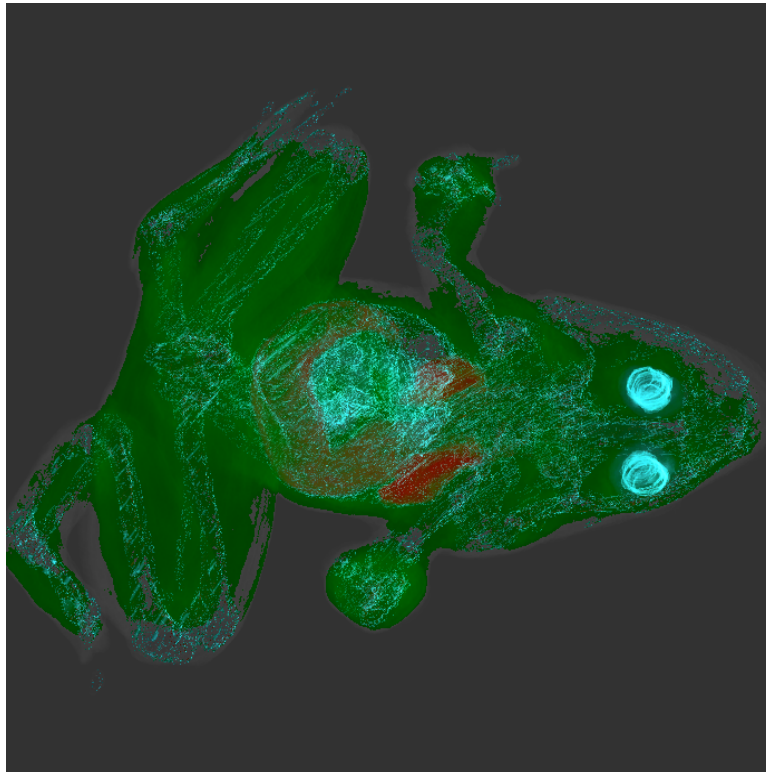
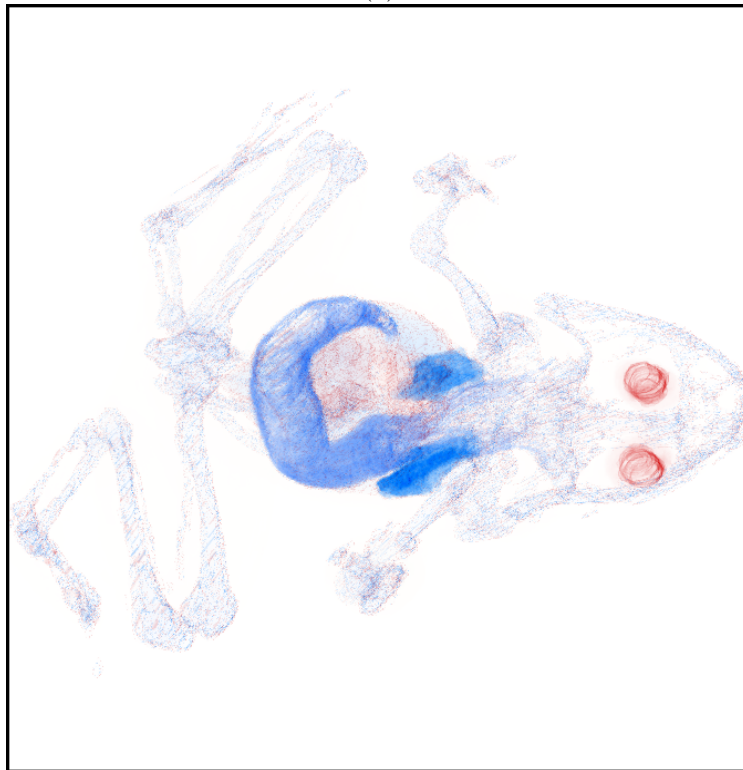


Fig. 6. Colour signals for 1 to 6 bounces of lights D65 (figures (a) to f)) and A (figures (g) to l)) from the six metameric reflectances in order, accumulating the signal over bounces. Solid: actual colour signal; dashed: spectral Factor Model approximation.



(a)



(b)

Fig. 7. Volume rendering of 3D biological data. (a): Under illuminant D65, most materials are metamereric. (b): Under illuminant A, most structures fade from view and just the stomach is prominent (image shown in inverse colour).

Figure Captions

Fig. 1

(a): First five SVD vectors. (b): Sharpened version of basis vectors.

Fig. 2

(a): One bounce. Exact: solid line; best least squares: dotted line; spectral Factor Model: dashed line. (b): Two bounces. (c): Three bounces.

Fig. 3

Three pairs of metameric surface spectral reflectance functions.

Fig. 4

Colour signals for the three metameric pairs, under tungsten lighting (solid) with the spectral Factor Model approximation (dashed).

Fig. 5

Spheres painted with metameric surface reflectance functions. Top row is under D65, bottom row is under illuminant A. Left column uses full-spectrum reflectances and illuminants in a full-spectrum 31-component raytracer, middle column shows results using the present spectral Factor Model approach, and right-hand column shows results using standard 3-component graphics. (Colour versions of figures may be viewed at <http://www.cs.sfu.ca/~mark/ftp/Josa03> .)

Fig. 6

Colour signals for 1 to 6 bounces of lights D65 (figures (a) to f)) and A (figures (g) to l)) from the six metameric reflectances in order, accumulating the signal over bounces. Solid: actual colour signal; dashed: spectral Factor Model approximation.

Fig. 7

Volume rendering of 3D biological data. (a): Under illuminant D65, most materials are

metameric. (b): Under illuminant A, most structures fade from view and just the stomach is prominent (image shown in inverse colour).