Unsupervised filtering of color spectra

Reiner Lenz and Mats Österberg

*Image Processing Group, Department of Electrical Engineering,
Linköping University, S-58183 Linköping, Sweden*

Jouni Hiltunen and Timo Jaaskelainen

Väisälä Laboratory, Department of Physics, University of Joensuu, FIN-80101 Joensuu, Finland

Jussi Parkkinen

*Department of Information Technology, Lappeenranta University of Technology,
FIN-53851 Lappeenranta, Finland*

Received April 17, 1995; revised manuscript received January 22, 1996; accepted February 21, 1996

We describe a class of unsupervised systems that extract features from databases of reflectance spectra that sample color space in a way that reflects the properties of human color perception. The systems find the internal weight coefficients by optimizing an energy function. We describe several energy functions based on second- and fourth-order statistical moments of the computed output values. We also investigate the effects of imposing boundary conditions on the filter coefficients and the performance of the resulting systems for the databases with the reflectance spectra. The experiments show that the weight matrix for one of the systems is very similar to the eigenvector system, whereas the second type of system tries to rotate the eigenvector system in such a way that the resulting filters partition the spectrum into different bands. We also show how the system can be forced to use weight vectors with positive coefficients. Systems consisting of positive weight vectors are then approximated with Gaussian quadrature methods. In the experimental part of the paper we investigate the properties of three databases consisting of reflectance spectra. We compare the statistical structure of the different databases and investigate how these systems can be used to explore the structure of the space of reflectance spectra. © 1996 Optical Society of America

1. UNSUPERVISED PROCESSING OF SENSORY DATA

The automatic processing of color images is receiving a much greater interest recently than in the past. This is certainly due to the rapid performance increase of computer hardware, but it is also motivated by new application areas such as pattern recognition and automatic quality control, for example, in the printing industry. In this paper we describe some investigations of the space of color spectra that are relevant for human color vision. Sometimes we will motivate our approaches with findings obtained in the study of the human visual system and we will also compare some of our results with facts from biology, but our main interest is in the investigation and the effective processing of the color information as such.

The color of an object is a production of the brain, and therefore it is not possible to measure color directly. The cause of the color sensation, the electromagnetic spectrum reaching the eye, is, however, physically measurable. In biological color sensor systems there are usually several different sets of response function.[1] Processing of color in the human sensory system is done with three different types of photoreceptor, with total range of 380–780 nm and sensitivity maxima at 440, 545, and 590 nm. The wavelength response curves of these photoreceptors can be found in Ref. 2. This is the reason why three-dimensional representations have dominated color research and applications. These three-dimensional representations are mostly based on human tristimulus values,

$$\begin{aligned} X &= k\int S(\lambda)\overline{x}(\lambda)\mathrm{d}\lambda\,, \\ Y &= k\int S(\lambda)\overline{y}(\lambda)\mathrm{d}\lambda\,, \\ Z &= k\int S(\lambda)\overline{z}(\lambda)\mathrm{d}\lambda\,, \end{aligned} \tag{1}$$

where $S(\lambda)$ is the input spectrum, $\overline{x}(\lambda)$, $\overline{y}(\lambda)$, and $\overline{z}(\lambda)$ are the sensitivity functions of the human eye, and k is a normalizing factor. The CIE tristimulus functions are shown in Fig. 1. These tristimulus functions are derived from psychophysical measurements of human color vision, and they form the basis for most color coordinate systems. Therefore we compare our positive sensory functions with these tristimulus functions. Since these positive sensory functions are detector filters, it would also be reasonable to compare them with the cone responses.

These models describe each spectrum with the three values X, Y, and Z, and thus there are an infinite number of different spectra with the same description (X, Y, Z). This may be insufficient for many purposes that require a more detailed analysis of the spectra. Examples of investigations based on the full spectral data can be found in Refs. 3–6.

In the study of color processing it is of great interest to know how many basis vectors (corresponding to

0740-3232/96/071315-10$10.00

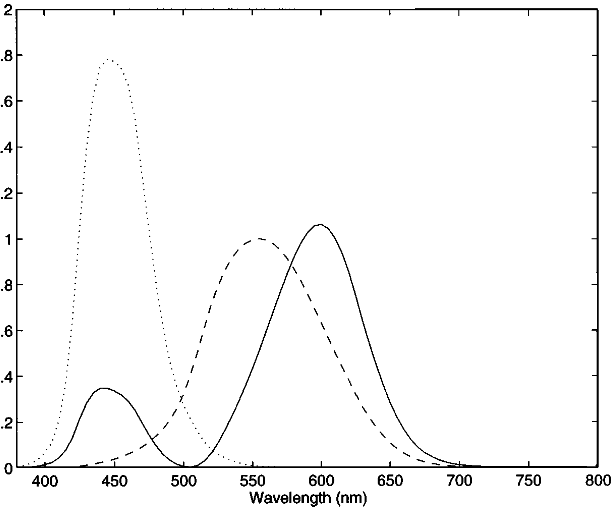

Fig. 1. CIE tristimulus functions.

the sensitivity functions) are needed for representing the color information and what the optimal basis vectors for three-dimensional color representation are. Investigations of the first problem can be found in Refs. 4 and 7. A neural-network-based system that tries to find a three-dimensional encoding of the information contained in the color spectra is described in Refs. 8 and 9. There a multilayer network is trained with spectral data, and the internal structure of the resulting network is then compared with that of conventional color systems.

A study of these problems requires access to all of the color information, and we will therefore use the color spectra as initial information. We will not take into account spatiospectral properties of the data, although they are important. We will use three databases of spectral data. The content of these databases should reflect the properties of the space of those colors that are perceivably by the human visual system. We will first describe the contents of these databases and compare their statistical properties.

We will then describe our strategy to compute the basis functions from the raw spectral data. The basic idea is to formulate a number of desirable properties of the resulting system. These properties are then incorporated into a quality or energy function, and the best basis functions are found by solving the optimization problem. Sometimes we will refer to the iterative optimization process as learning and to the resulting filter functions as the learned filter functions. Related methods are used in Refs. 10–14 to investigate properties of biological vision systems. We will describe several different types of systems based on different quality functions and illustrate their properties.

The raw reflectance spectrum of the object gives us information about the physically measurable color properties of an object. In some of our experiments we weighted these reflectance spectra by the $\overline{y}$ function given in Eqs. (1). By decreasing the importance of the red and the blue parts of the visible spectrum, we generate a rough approximation of the spectral data that are analyzed by the human visual system.

This study will give more information about the basic problem of the structure of the color space. This has been investigated by the authors of this paper in a series of previous studies.[4,7,15] In these studies we have shown that the color space can be described accurately by a subspace method. An aim of the present study is to find a set of positive basis functions that span the same color space as the subspace method does. Knowledge about different methods to describe the color space will help in the search for an efficient color analysis method, e.g., for color image analysis and transmission. Furthermore, it will give information for the research on human color vision and on modeling color vision systems in general.

Parallel optical inner product calculations are needed in some illumination systems for novel microscopy methods, in optical pattern recognition, and generally in systems producing illumination with arbitrary spectral shape (e.g., solar simulators).[16,17] These systems use spatial light modulators (SLM's) to filter a spectrum in a certain wavelength region. Since negative filter values cannot be handled by SLM's directly, it is necessary to consider the positive and negative filter values separately or to use some other multiplexing method. Methods to derive filters with nonnegative filter values are thus especially important in this context, since the nonnegative filters can be implemented directly in the SLM. Furthermore, avoiding the division into positive and negative filter parts doubles the number of channels that can be used on the SLM, and the necessary electronic manipulations are also minimized.

Finally we will describe a fast digital implementation of the filtering operation. The method described uses a Gaussian quadrature method to approximate, by means of a finite sum, the integral over a function weighted by a positive weight function. The application of such an approximation is possible only when positive filter functions are used. When Gaussian quadrature methods are used, we have to consider two kinds of error: the approximation error introduced by using only a small number of filter functions and the errors originating in the quadrature approximation. The relation between these two types of error is investigated in the final experiments.

2. DATABASES

In our experiments we used three different databases: two are based on the *Munsell Book of Color*[18] and one on the Natural Color System (NCS) developed by the Scandinavian Color Institute AB. The old Munsell database consisted of 1253 spectra represented by 61 measurements each. We often ignored the first five measurements in these spectra, since the illumination in this spectral band was too noisy. The remaining values represent the reflectance in the range from 425 nm to 700 nm with a step length of 5 nm. We refer to this database as the Munsell-I database if necessary. We will call the new database based on the Munsell colors the Munsell database. This database consisted of 1269 spectra of 421 measurements each. They are in the range from 380 nm to 800 nm in 1-nm steps. The NCS database consisted of 1513 spectra based on the NCS color system. The spectral range was 380 nm to 780 nm in 5-nm steps.

In our first experiment we selected from the three databases the measurements in the 425-nm to 700-nm

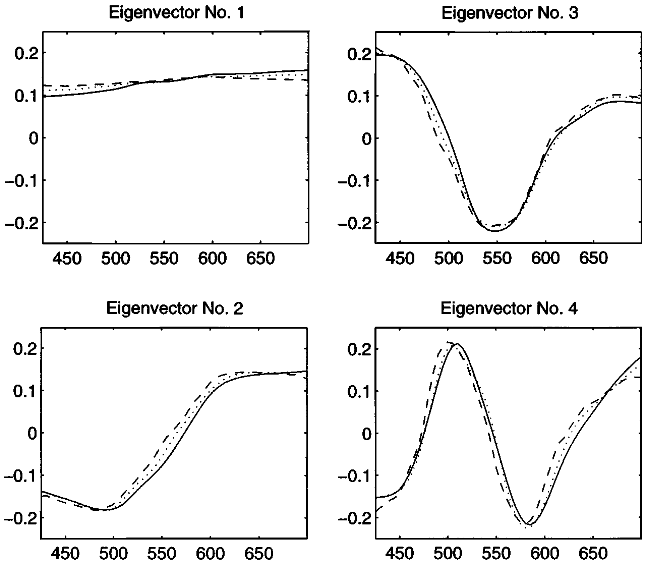

Fig. 2. First four eigenvectors for the three databases: Munsell-I (dashed), Munsell (dotted), NCS (solid).

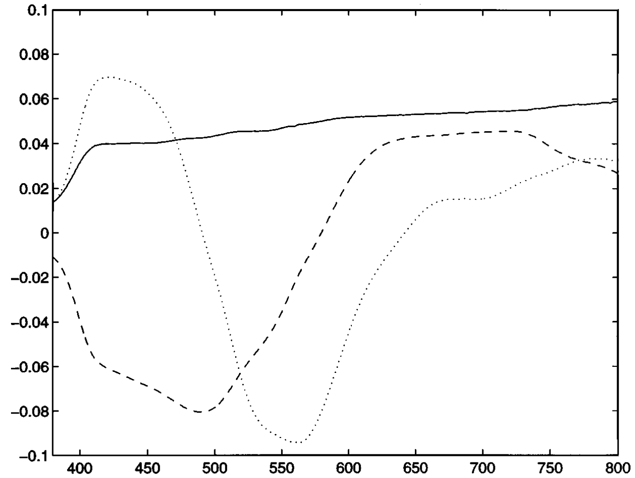

Fig. 3. First three eigenvectors: first (solid), second (dashed), third (dotted).

band and computed from them the eigenvectors and the eigenvalues of the correlation matrix.

As an example we show in Fig. 2 the first four eigenvectors from the three different databases. The dashed curves show the eigenvectors from the old Munsell database, the dotted curves were computed from the new Munsell data, and the solid curves are from the NCS set. We found that the first few eigenvectors are almost identical for all databases. In Fig. 3 we see the first three eigenvectors from the Munsell database. The solid curve shows the first eigenvectors, the dashed curve shows the second eigenvector, and the dotted curve shows the third eigenvector. Here we used the full spectral range of the new database. Table 1 contains the correlation coefficients between the first four eigenvectors from the Munsell databases and the NCS system. These vectors were calculated from the 425-nm to 700-nm range with 5-nm steps. The eigenvalues from the full Munsell spectra are listed in Table 2. The conclusion that we can draw from these experiments is that the databases have very similar statistical properties. Therefore we will mainly work with the Munsell databases in the following sections.

3. OPTIMIZATION AND UNSUPERVISED FILTER SYSTEMS

Methods to construct filter systems for raw, unclassified data are known as unsupervised learning procedures in the theory of neural networks. Such systems are often specified by their learning rules, i.e., how they change their internal weights or filter coefficients. In this paper we will use a different approach to find the filter functions. Here we select first some properties of the output vectors computed by the system. Then we design a quality (or energy) function that measures these properties. Finally we use an iterative optimization procedure to find the filters. This has the advantage that the huge existing knowledge in optimization theory can be applied to find efficient implementations of the learning procedure.

In all our investigations we start from the requirement that the extracted feature vectors should convey as much information about the space of input signals as possible. The term information as used here is not the entropy-based definition used in information theory; rather, it is used in the Karhunen–Loéve sense of minimum mean squared error. It turns out that a useful class of energy functions can be built around the determinant of

Table 1. Correlations of Eigenvectors from the Different Databases

Number	NCS Versus Munsell-I	NCS Versus Munsell	Munsell Versus Munsell-I
1	0.9940	0.9982	0.9987
2	0.9847	0.9965	0.9953
3	0.9835	0.9946	0.9942
4	0.9662	0.9939	0.9780
5	0.9316	0.9810	0.9742
6	0.8692	0.9287	0.9666
7	0.7312	0.9018	0.6725
8	0.7847	0.9102	0.6934
9	0.4676	0.9138	0.6185
10	−0.1552	0.8906	−0.1923

Table 2. Eigenvalues from the Full Munsell Database

Number	Eigenvalue
1	47.5567
2	2.3453
3	0.8268
4	0.1814
5	0.0635
6	0.0331
7	0.0319
8	0.0170
9	0.0058
10	0.0047

the output correlation matrix. We describe this as follows: The patterns in the database are given by the vectors p. The correlation matrix C_p of these signals is $C_p = E(p \cdot p')$, where the prime denotes transposition and $E(x)$ is the expected value of the stochastic variable x. The output of the system for the input p is given by the vector $o = f(p)$, and the correlation matrix of the output is $C_o = E(o \cdot o')$. We choose as the first term in our energy function the determinant of the output correlation matrix,

$$Q_D = -\det(C_o)\,. \tag{2}$$

In the statistical literature (see, for example, Ref. 19) this is known as the generalized variance of the stochastic process o. In contrast to other variance measures (like the trace of the matrix, also known as the total variance) it has the advantage that it is based on combinations of all second-order statistics of the process. It excludes, for example, the case in which there are linearly dependent filter vectors in the final system (see Ref. 20 for further information).

In the case in which the function f is linear we have a weight matrix W such that $o = W \cdot p$. In this case it can be shown that the matrices W that lead to an extremum of Q_D are connected to the first eigenvectors of C_p through a rotation. This implies that we can multiply W by a rotation matrix and still obtain the same energy value. In our previous investigations[21–23] we used this freedom and introduced new requirements for the output vectors. These additional properties were introduced by modifying the energy function to functions of the form $Q = Q_D Q_A$, where Q_A is an additional term. We found that terms depending on fourth-order statistical moments of the output values produced especially interesting results. At about the same time and independent of our work the importance of fourth-order moments in neural network research was recognized in the context of sparse coding.[13] It is also interesting to note that systems that maximize fourth-order moments like the kurtosis have received a lot of attention recently in connection with blind deconvolution techniques. The goal here is to recover signals from their degraded observations even if the degradation process is unknown (see Ref. 24 for an introduction).

The variant of the fourth-order function that worked best and that was easiest to implement is given by the variance of the squared output values, Q_V [Eq. (3) below], but we experimented also with other variants like Q_4, described in Eq. (4) below. This particular choice of the quality function was investigated because the function $x(1 - x)$ and the entropy function $-x \ln x$ are similar for x in the interval [0, 1]. The function Q_4 thus measures a kind of entropy for the output values o_n. Filter systems derived from this quality function try to produce binary output values -1, 1, and 0. The output values are thus a type of sparse coding of the input signals. In a number of cases they also support unsupervised classification by simple thresholding: $|o_n| \geq T$ signals the presence of a certain feature, whereas $|o_n| < T$ signals its absence. Note, however, that the Q_4 term requires that the output values be in the range $[-1, 1]$. (A more detailed description of this type of system can be found in Refs. 20 and 22.) These functions are

$$Q_A = Q_V = \sum_{n=1}^{N} \mathrm{var}({o_n}^2)\,, \tag{3}$$

$$\frac{1}{Q_A} = Q_4 = \sum_{n=1}^{N} E[{o_n}^2(1 - {o_n}^2)]\,. \tag{4}$$

In many applications it might be necessary to impose certain restrictions on the values of the weights in the system. An example is the optical implementation of the filters, where it is desirable to have only positive filter coefficients. This will also be a key requirement for the approximation method based on Gaussian quadrature that is described in Section 4. Other useful conditions are weight vectors of unit length or weights in the interval [0, 1].

These boundary conditions are relatively easy to incorporate into the optimization process. In our implementation we enforce unit-length weight vectors by using polar coordinates. Optimization is then done in the angular variables. In a similar way we enforce positive weights by using weights of the form $w_{ij} = \mathrm{abs}(x_{ij})$. Optimization takes place in the x variables, but computation of the output values is performed in the w variables.

4. GAUSSIAN QUADRATURE APPROXIMATIONS

In Section 3 we described unsupervised systems that can find filter (or weight) vectors $w_i = (w_{i1}, \ldots, w_{iN})$, $i = 1, \ldots, M$, from examples. If we collect these vectors in a matrix W, then an input vector x is mapped to the feature vector y with $y = Wx$. The weight matrix will no longer change after optimization has been completed. In many applications we will then use this fixed weight matrix W and compute the products for an large number of different input vectors x. In such applications it is important to evaluate the matrix–vector product $y_i = w_i \cdot x$ as efficiently as possible. We mentioned above that the products $w_i \cdot x$ can be computed optically if all the entries in w_i are nonnegative. But positivity is also important in digital implementations, since it makes it possible to use Gaussian quadrature methods. Gaussian quadrature has been used earlier in color image processing,[25] but since it is not a standard method and since we will use it in its most general form, we will give here a summary of the main facts. More information on Gaussian quadrature can be found in the standard texts on numerical mathematics, such as Ref. 26.

We first give a summary of some of the results from the theory of orthogonal polynomials. We consider a nonnegative weight function $w(\lambda)$, which is defined on the real axis and which will be fixed from now on. It can then be shown that there is a system of polynomials $p_k(\lambda)$ with the following properties:

1. p_k is of degree k.
2. If we define the scalar product as

$$\langle f,\, g\rangle = \int_{-\infty}^{\infty} f(\lambda) g(\lambda) w(\lambda) \mathrm{d}\lambda\,,$$

then they are orthogonal, i.e., $\langle p_k,\, p_l\rangle = 0$ if $k \neq l$.

3. There is a recurrence relation:

$$p_{k+1}(\lambda) = (\lambda - a_k)p_k(\lambda) - b_k p_{k-1}(\lambda),$$

where a_k and b_k are real constants and $p_{-1}(\lambda) = 0$ and $p_0(\lambda) = 1$. The leading coefficient is always unity.

4. The polynomial p_k has exactly k different roots.

5. The roots are interlaced; i.e., between two roots of p_k there is exactly one root of p_{k+1}.

Now assume that we want to compute the integral $\int_{-\infty}^{\infty} f(\lambda)w(\lambda)\mathrm{d}\lambda$ for a large number of different functions f. The Gaussian quadrature rule of order n approximates the integral by a finite sum:

$$\int_{\infty}^{\infty} f(\lambda)w(\lambda)\mathrm{d}\lambda \approx \sum_{k=1}^{n} f(x_k)\omega_k, \tag{5}$$

where the n points x_k and the n weights ω_k $(k = 1, \ldots, n)$ are computed from the weight function $w(\lambda)$ alone. These Gaussian quadrature rules are optimal in the sense that the approximation is exact if f is a polynomial of degree less than $2n$. The fundamental theorem of Gaussian quadrature states that the points x_k, the nodes, are the roots of the orthogonal polynomial p_n constructed from the weight function w. The weights are given by $\omega_k = \langle p_{n-1}, p_{n-1}\rangle / p_{n-1}(x_k)p_n'(x_k)$, where p_n' is the derivative of p_n.

For the classical weight functions all the quantities involved, i.e., a_k, b_k, x_k, and p_k, are known, and an application of the theory is straightforward. In the case of a nonstandard weight function the situation is more complicated, and numerical stability problems are important. In our implementation we followed the approach in Ref. 27 and basically used the routines described there. In this approach the weights and the nodes are computed in two steps: first the coefficients a_k and b_k in the recurrence relation are computed, and then the weights and the nodes are computed from them. The method used in our implementation is based on so-called modified moments. These are defined as follows: Assume that you have a given, known system of orthogonal polynomials π_k derived from another weight function. Then the modified moments ν_k are defined as $\nu_k = \int_{-\infty}^{\infty} \pi_k(\lambda)w(\lambda)\mathrm{d}\lambda$. The recurrence coefficients are then computed from these modified moments. For the details the reader might consult Ref. 27 and the references cited there.

In our application we want to approximate the scalar product $w' \cdot x$ for a given positive weight vector w and varying data vectors x. In the following experiments we make the assumption that the vectors w and x represent piecewise-constant functions W and X that are defined on the interval $[-1, 1]$. The scalar product is then given by the integral $\int_{-1}^{1} X(\lambda)W(\lambda)\mathrm{d}\lambda$. The modified moments are then the sums: $\nu_l = \sum_{k=1}^{N} w_k \int_{I_k} \pi_l(\lambda)\mathrm{d}\lambda$, where the interval I_k is equal to $[-1 + 2(k-1)/N, -1 + 2k/N]$. In our implementation we used as polynomials π_k the Legendre polynomials, which are derived from the weight function $\tilde{w}$ that is equal to unity on the interval $[-1, 1]$ and zero elsewhere. For these integrals involving the Legendre polynomials we find the values

$$\int \pi_l(\lambda)\mathrm{d}\lambda = \frac{1}{2l+1}[\pi_{l+1}(\lambda) - \pi_{l-1}(\lambda)].$$

The assumption of a piecewise-constant weight function could be weakened by considering other, smoother versions of the weight functions like splines, but we did not explore these possibilities in our experiments.

5. EXPERIMENT RESULTS

In most of our experiments we used our optimization programs based on an implementation of the CONMIN algorithm together with the line-search algorithm developed by Moré and Thuente. (For more information see Refs. 28–31 and the references in Ref. 22.) We found that the optimization was very efficient. Using these algorithms has furthermore the advantage that they can be implemented in parallel systems in which the different modules are updated in parallel. In most of our experiments we used the subsampled data from the new Munsell database. Sampling in 5-nm steps describes the spectra with 85 measurements each. Using the reduced data set speeded up optimization considerably. We also compared the results obtained from these reduced spectra with the results computed from the full data set, and we found that the resulting filter functions are nearly identical.

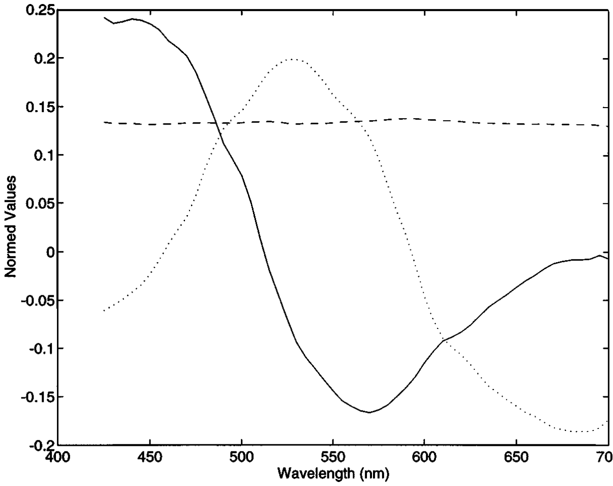

Fig. 4. Learned filter functions: CONMIN with use of Q_4.

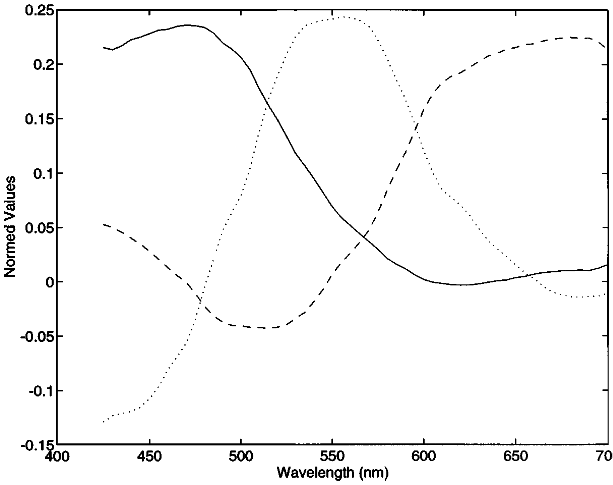

Fig. 5. Learned filter functions: CONMIN with use of Q_V.

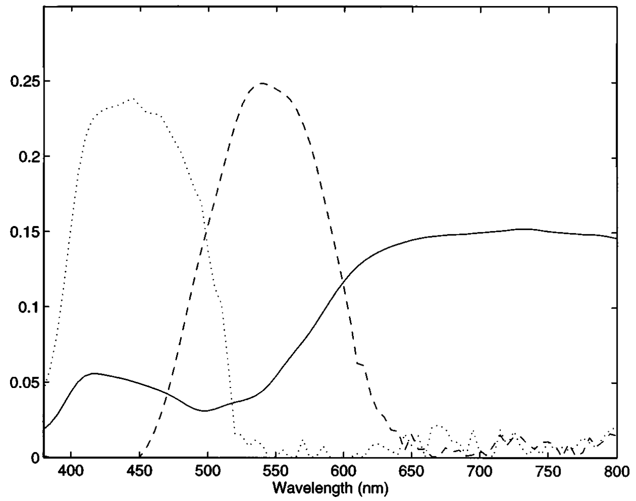

Fig. 6. Learned filter functions: CONMIN with use of Q_V and positive weights.

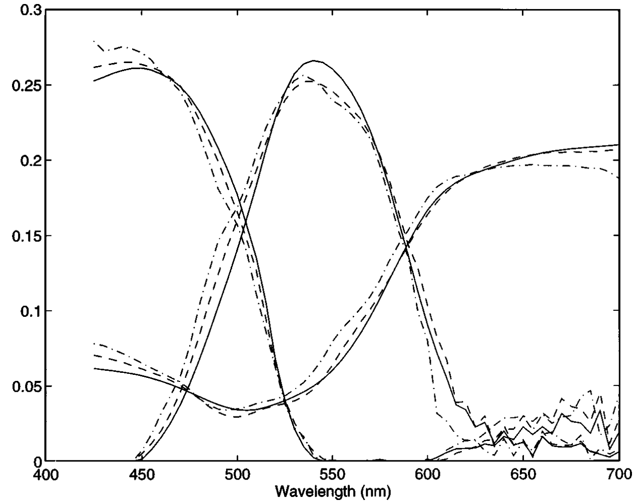

Fig. 7. Learned filter functions: CONMIN with use of Q_V and positive weights: NCS (solid), Munsell-I (dashed–dotted), Munsell (dashed).

In the first series of experiments we compared the performance of systems based on the energy functions Q_D/Q_4 and Q_DQ_V [Eqs. (4) and (3), respectively]. In these experiments we trained a system consisting of three units. We used the CONMIN optimization program and described the weights in polar coordinates, thus enforcing the strict boundary condition that the rows of the weight matrix should have unit length. We used the Munsell-I database and normed each individual entry in it to unit length, since the function Q_D/Q_4 requires outputs in the range $[-1, 1]$. The results are shown in Figs. 4 and 5.

These figures show that the system based on the Q_4 function stabilizes in a state that is qualitatively similar to the three eigenvectors of the correlation matrix (note, however, that each input vector was normed to unit length in these experiments). The system based on the Q_V function, on the other hand, tries to split the domain into three separate spectral bands.

In another experiment we used the unnormed entries from the new Munsell database and the CONMIN program together with the Q_V function. Here we optimized in the polar coordinates, but we used as weights the absolute values of the Cartesian coordinates x: $w = \text{abs}(x)$. This enforced positive weights. The resulting filter vectors are shown in Fig. 6. This shows that the positivity constraint enhances the ability of the original system to partition the spectrum into different bands. The resulting filters could be characterized as red–green–blue filters. These filter functions can be compared with the tristimulus functions $\overline{x}$, $\overline{y}$, and $\overline{z}$ mentioned in Section 1. These functions are shown in Fig. 1, and we see that the main difference lies in their behavior in the region with the long wavelengths; here the tristimulus functions fall to zero, but one of the learned filter functions has relatively high values. An inspection of the spectra in the database shows that the spectra have comparatively high values in this region, and since the filter functions are designed to have high correlations with the input patterns, it is understandable why they have high coefficients in this region. It is, however, still an open question why the human visual system chooses to ignore information in this spectral region.

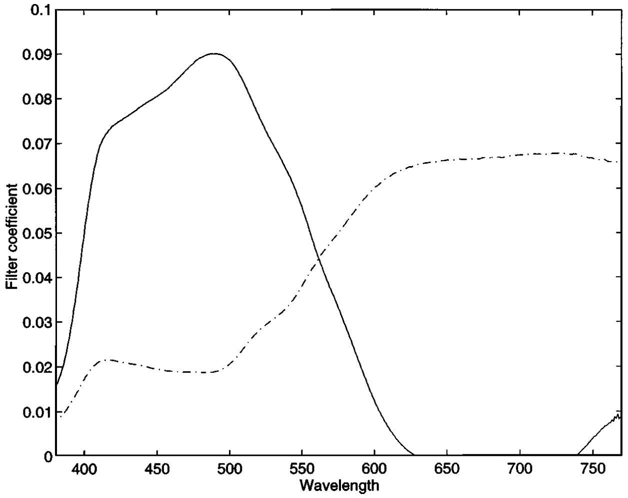

Fig. 8. System with two filter functions.

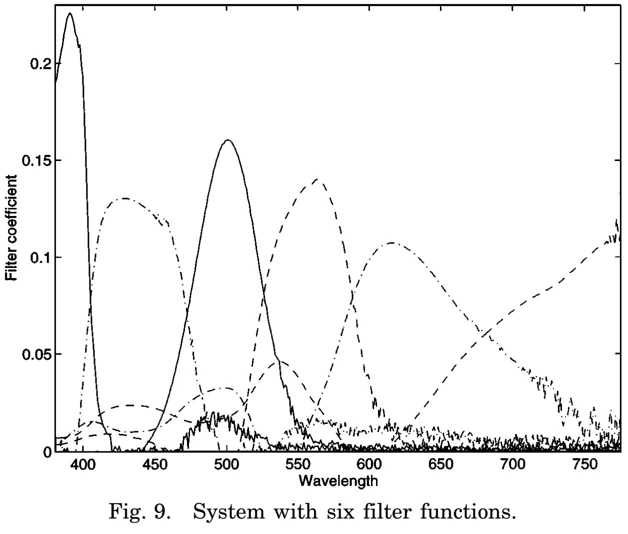

Fig. 9. System with six filter functions.

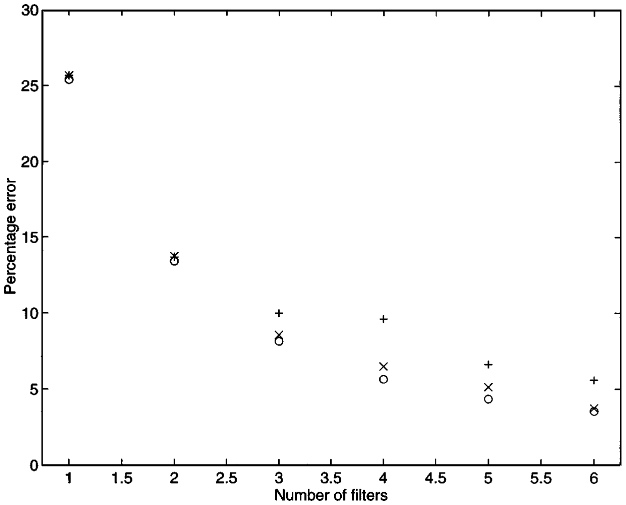

Fig. 10. Approximation errors: eigenvector system (○), unrestricted filter systems learned from the Munsell database (×), positive filter functions computed from the same data set (+).

The statistical properties (described in Section 2) of the three databases are very similar. This similarity is reflected in the structure of the filter systems computed from the three database. In Fig. 7 we see three filter systems consisting of three filter functions each. The filter functions plotted with the solid curves are from the NCS database, the dashed-curve filters are from the new Munsell database, and the dashed–dotted-curve filters are from the old Munsell database. In all three cases we used the spectral band from 425 to 700 nm with a 5-nm sampling.

Figures 8 and 9 illustrate the behavior of the filter systems when we vary the number of filter functions. In Fig. 8 we train a system consisting of two filters with positive weights with the spectral data in the range 381 to 800 nm with 1-nm sampling. In Fig. 9 we use the same spectra to construct a system with six positive functions. The two-filter system splits the spectral band roughly into two adjacent regions. The six-filter system has also a bandpass characteristic, but it is interesting to see that several of the resulting filter functions are bimodal and that one has even a third local maximum.

The computation of the output vectors o [or the (X, Y, Z) values] mentioned above is a projection from the high-dimensional space of the spectra to a low-dimensional coordinate space. Obviously some information is lost in this process, and there are therefore many

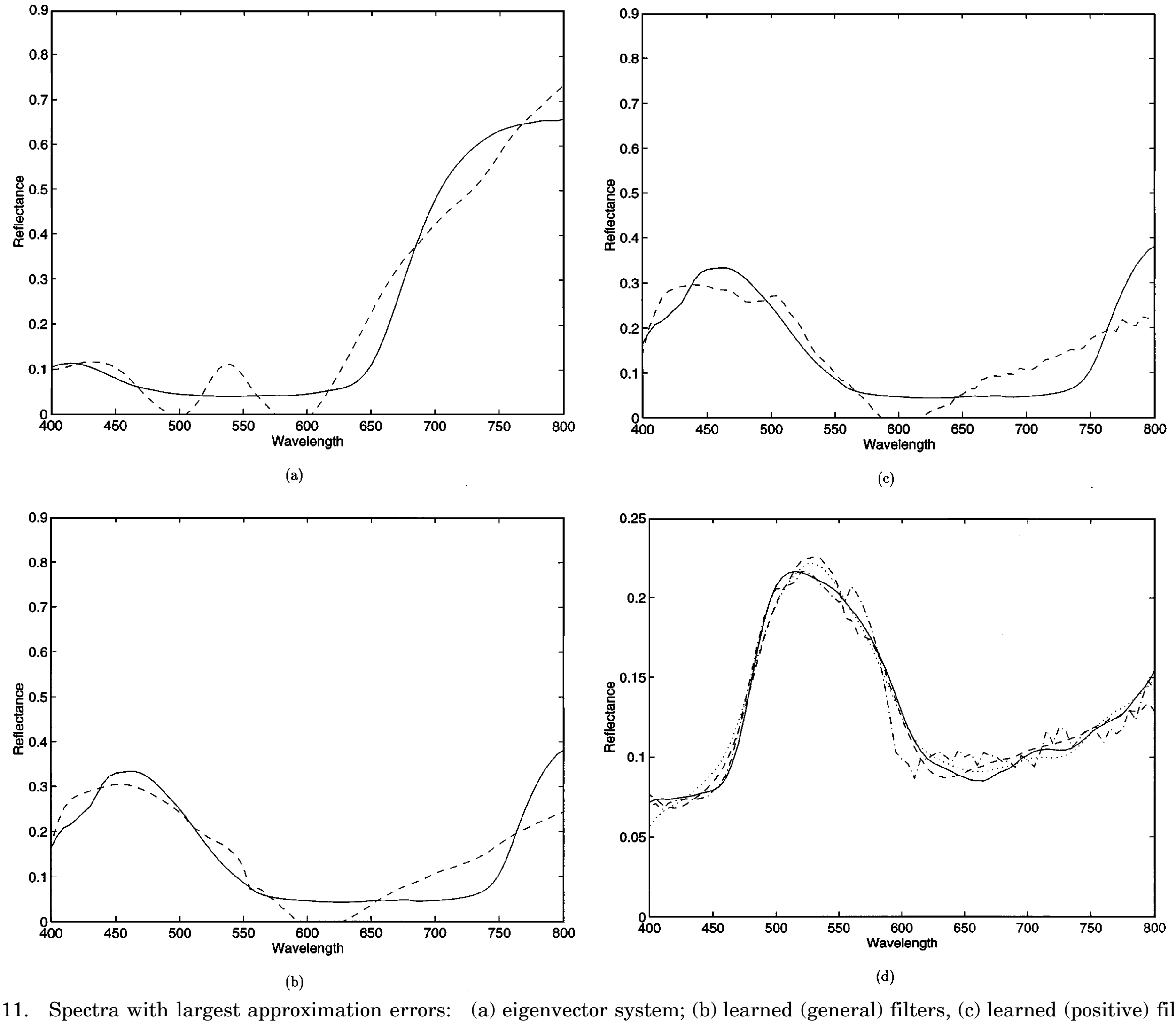

Fig. 11. Spectra with largest approximation errors: (a) eigenvector system; (b) learned (general) filters, (c) learned (positive) filters, (d) typical approximation error for learned (positive) filters. See the text for an explanation of the different curves.

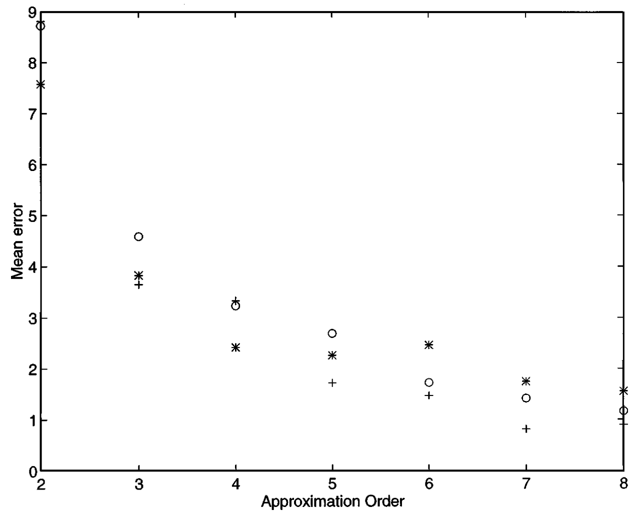

Fig. 12. Approximation errors for Gaussian quadrature: weight vectors (∗), 2 (+), and 3 (○).

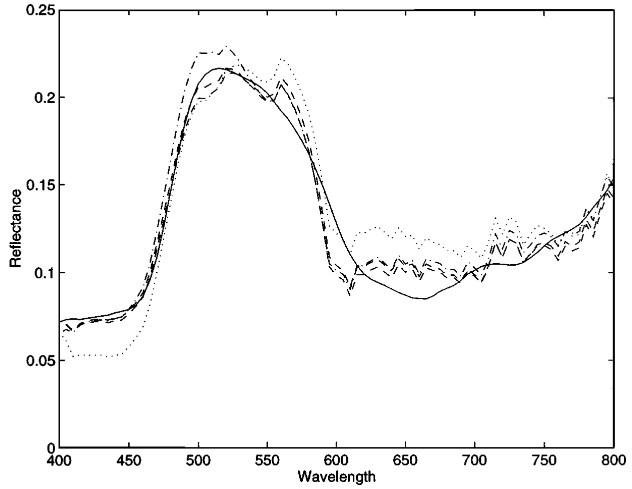

Fig. 13. Gaussian quadrature approximations: original spectrum (solid); reconstructions based on full vectors and eight-node approximation (dashed), two-node approximation (dotted), four-node approximation (dashed–dotted).

different spectra that map to the same coordinate vector. In many applications it would, however, be interesting to compute from the coordinate values [for example, the *o* or the (X, Y, Z) vector] an approximation of the underlying spectrum. Such an approximation is easy if the filter functions are the eigenvectors, since these vectors form an orthonormal system. In this case we collect the eigenvectors in a matrix B, and since they are orthonormal, we find that $x = BB'x$. In the general, nonorthonormal case we can express the new system $\tilde{B}$ as a linear combination of the orthonormal system B, i.e., $\tilde{B} = BM$, where the matrix M describes the expansion of the new basis vectors in terms of the orthonormal system. If we introduce the matrix $S = (M'M)^{-1} = (M'B'BM)^{-1} = (\tilde{B}'\tilde{B})^{-1}$, then we see that $x = \tilde{B}S\tilde{B}'x = \tilde{B}Sy$. In our application the vector y contains features extracted with the filter system, S describes the deviation from orthonormality, and $\tilde{B}$ is the filter system. In Fig. 10 we summarize the results of some of our experiments. In these experiments we use the spectra in the range 380 nm to 800 nm with 5-nm steps. We trained a three-filter system with positive filter functions and a general three-filter system and compared the filter systems obtained with the eigenvector system computed from the same data set. Figure 10 shows the performance of the approximations based on the eigenvectors system (○), the (unrestricted) filter systems learned from the Munsell database (×), and the positive filter functions computed from the same data set (+). The error was computed as the mean value over the quantity: 100 × (reconstruction error/signal energy). We see that the learned filter systems and the eigenvectors systems are comparable, whereas the system with the positivity constraint has a slightly worse performance.

In Fig. 11 we illustrate the approximation errors for the different systems. In this experiment we used filter systems with six filter functions. We computed for each spectrum the approximation and the corresponding error. In the figure we see the spectral data for which the approximation error was maximal. In the case of the eigenvector expansion (a) this was spectrum 204, and for the learned filter systems (b) and (c) it was spectrum 892. In Figs. 11(a)–11(c) the solid curves show the measured spectral data, and the dashed curves show the approximation. In Fig. 11(d) we see an example of an approximation with an approximation error that is near the average error. This approximation shows spectrum 526, and the approximation error for this measurement vector and the system with positive coefficients was 5.59%. The solid curve is the original spectrum, the dotted curve uses the first six eigenvectors, the dashed curve is based on the general learned filter system, and the dashed–dotted curve comes from the positive filters.

Finally we describe the results of our experiments with the Gaussian quadrature methods. In these experiments we used a filter system with three positive weight vectors that we learned from the spectra in the range from 380 to 800 nm with 5-nm sampling. For each of the filter functions we computed the nodes and the weights for the Gaussian quadrature method for orders 2–8. Then we computed for each spectrum in the data set the error of the approximation. This error is computed as $100|(y_{\text{true}} - y_G)/y_{\text{true}}|$, where y_{true} is the value computed from the full-weight and data vectors and y_G is the approximated value computed with the Gaussian quadrature method. The mean values of this error are shown in Fig. 12 for the weight vectors 1 (∗), 2 (+), and 3 (○) and approximation orders 2–8.

In Fig. 13 we show the reconstruction of spectrum 526, which was used in Fig. 11(d). Here we use the same six-filter system as that in Fig. 11, but we compute the six coefficients in four different ways: We compute it using the full-weight vectors, and we use Gaussian quadrature with two, four, and eight nodes. The original spectrum is shown as the solid curve, and the reconstructions based on the full vectors and the eight-node approximation are shown as dashed curves. The dotted curve is based on two nodes, and the dashed–dotted curve is based on four nodes. We see that the error introduced by the eight-node approximation is negligible and that the four-node approximation is probably still acceptable. Since the original vectors consisted of

85 coefficients each, we have a achieved a tenfold speedup in the eight-node case and a 20-fold speedup in the four-approximation.

6. SUMMARY AND CONCLUSIONS

We investigated three databases containing reflectance spectra describing the colors relevant for human perception: the Munsell colors measured by an acousto-optical-modulator-based spectrometer, the Munsell colors measured by an accurate spectrophotometer, and the NCS colors measured by a an accurate spectrophotometer. Table 1 (Munsell versus NCS) shows that our previous results (derived from the acousto-optical modulator data set) are valid and that the Munsell and NCS color sets describe approximately the same color space.

We then introduced a class of systems that find a set of basis functions from the raw input data. These systems are derived from energy functions that included second- and fourth-order statistics of the output vectors. We also demonstrated how additional constraints (like the positivity of the weights) can be incorporated into the network design.

We found that the performance of the learned filter systems was in general comparable with that of the eigenvectors system. This is also true for the systems in which we required that all filter coefficients should be nonnegative. These filters are especially interesting for optical realizations of filters. Because of its parallel nature, optical computing is an interesting option for operations in the optical part of the spectrum. The experiments with the positive filter functions demonstrate that such systems can be used for accurate analysis and reconstruction of the color space used by the human visual system. The study of such positive filter systems is also interesting in the study of biological vision systems, since these color detectors are also based on positive response functions. In biological vision systems one can find a varying number of color-sensitive receptors, and it is therefore of interest to study the effect of the number of available receptor types. Figures 8 and 9 show a two-filter and a six-filter system, respectively, and the shape of the optimized positive filter functions in such color detection systems. They demonstrate a general bandpass characteristic of the positive filter functions, but they also show that such systems will usually contain a number of multimodal filter functions.

For filter functions with positive weights it is also possible to use Gaussian quadrature methods to compute the filter responses. For systems consisting of three filter functions we found that an approximation based on three to eight nodes gave satisfactory approximation results.

ACKNOWLEDGMENTS

Reiner Lenz and Mats Österberg were supported by the Swedish Research Council for Engineering Sciences (TFR) under grant TFR-95-467 and by the Swedish National Board for Industrial and Technical Development (NUTEK) under grant P1839-1. The NCS database was obtained from the Scandinavian Color Institute in Stockholm and provided by B. Kruse. J. Hiltunen is supported by the Machine Vision program of the Technology Development Centre (TEKES) . J. Hiltunen and M. Österberg received travel grants from the Nordic Research Network in Computer Vision (NORVIC).

Correspondence should be addressed to Reiner Lenz. Tel: +46-13-281337; fax: +46-13-282599; e-mail: reiner@isy.liu.se.

REFERENCES

1. J. F. W. Nuboer, "A comparative view on colour vision," Neth. J. Zoo. **36,** 344–380 (1986).
2. G. Wyszecki and W. S. Stiles, *Color Science,* 2nd ed. (Wiley, London, 1982).
3. J. Cohen, "Dependency of the spectral reflectance curves of the Munsell color chips," Psychon. Sci. **1,** 369–370 (1964).
4. J. P. S. Parkkinen, J. Hallikainen, and T. Jaaskelainen, "Characteristic spectra of Munsell colors," J. Opt. Soc. Am. A **6,** 318–322 (1989).
5. 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).
6. M. D'Zmura and G. Iverson, "Color constancy. III. General linear recovery of spectral descriptions for lights and surfaces," J. Opt. Soc. Am. A **11,** 2389–2400 (1994).
7. T. Jaaskelainen, J. Parkkinen, and S. Toyooka, "Vector subspace model for color representation," J. Opt. Soc. Am. A **7,** 725–730 (1990).
8. S. Usui, S. Nakauchi, and M. Nakano, "Reconstruction of Munsell color space by a five-layer neural network," J. Opt. Soc. Am. A **9,** 516–520 (1992).
9. S. Usui and S. Nakauchi, "Computational color vision models by neural networks," in *Computational Intelligence, Imitating Life,* T. M. Zurada, R. J. Marks II, and C. J. Robinson, eds. (Institute of Electrical and Electronics Engineers, Piscataway, N.J., 1994), pp. 252–263.
10. F. Attneave, "Some informational aspects of visual perception," Psychol. Rev. **61,** 183–193 (1954).
11. H. B. Barlow, "The coding of sensory messages," in *Current Problems in Animal Behavior,* W. H. Thorpe and O. L. Zangwill, eds. (Cambridge. U. Press, Cambridge, 1961), pp. 331–360.
12. J. J. Atick, "Could information theory provide an ecological theory of sensory processing?" Network **3,** 213–251 (1992).
13. D. J. Field, "What is the goal of sensory coding," Neural Comput. **6,** 559–601 (1994).
14. J. G. Taylor and M. D. Plumbley, "Information theory and neural networks," in *Mathematical Applications to Neural Networks,* J. G. Taylor, ed. (Elsevier, New York, 1993), pp. 307–340.
15. K. Mantere, J. Parkkinen, M. Mäntyjärvi, and T. Jaaskelainen, "An eigenvector interpretation of the Farnsworth Munsell 100-hue test," J. Opt. Soc. Am. A **12,** 2237–2243 (1995).
16. T. Jaaskelainen, S. Tyooka, S. Izawa, and H. Kadono, "Color classification by vector subspace method and its implementation using liquid crystal spatial light modulator," Opt. Commun. **89,** 23–29 (1992).
17. N. Hayasaka, S. Toyooka, and T. Jaaskelainen, "Iterative feedback method to make a spatial filter on a liquid crystal spatial light modulator for 2D spectroscopic pattern recognition," Opt. Commun. **119,**643–651 (1995).
18. *Munsell Book of Color, Matte Finish Collection* (Munsell Color, Baltimore, 1976).
19. B. Flury, *Common Principal Components and Related Multivariate Models* (Wiley, New York, 1988).
20. R. Lenz and M. Österberg, "Computing the Karhunen–Loéve expansion with a parallel, unsupervised filter system," Neural Comput. **4,** 382–392 (1992).
21. R. Lenz and M. Österberg, "A new method for unsupervised linear feature extraction using fourth order moments," Pattern Recognition Lett. **13,** 827–836 (1992).
22. M. Österberg, "Quality functions for parallel selective principal component analysis," Ph.D. dissertation (Linköping University, Linköping, Sweden, 1994).

23. M. Österberg, and R. Lenz, "Unsupervised parallel feature extraction from first principles," in *Advances in Neural Information Processing Systems 6,* J. D. Cowan, G. Tesauro, and J. Alspector, eds. (Morgan, Kaufmann, San Francisco, Calif., 1994), pp. 136–144.
24. J. A. Cadzow and X. Li, "Blind deconvolution," Digital Signal Process. **5,** 3–20 (1995).
25. C. F. Borges, "Numerical determination of tristimulus values," J. Opt. Soc. Am. A **11,** 3152–3161 (1994).
26. J. Stoer and R. Bulirsch, *Introduction of Numerical Analysis* (Springer-Verlag, New York, 1980).
27. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes* (Cambridge U. Press, Cambridge, 1992).
28. J. Nocedal, "Theory of algorithms for unconstrained optimization," Acta Numer. **1,** 199–242 (1992).
29. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, N.J., 1983).
30. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," ACM Trans. Math. Software **20,** 286–307 (1994).
31. D. F. Shanno and K. H. Phua, "Remark on algorithm 500," ACM Trans. Math. Software **6,** 618–622 (1980).