CHAPTER II

BASIC BACKGROUND AND RELATED TOPICS

In this chapter, the fundamental knowledge of the two-dimensional subspace analysis algorithm is described. First of all, the traditional 1D subspace is represented in vector form. Next, subspace analysis with tensor PCA and the Multilinear Principal Component Analysis (MPCA) is introduced. Finally, we review the basic knowledge of color system in image processing.

2.1 Principal Component Analysis (PCA) Subspace Analysis

Linear dimensionality reduction techniques have been widely used in pattern recognition and computer vision, such as face hallucination, image retrieval, etc. Principal Components Analysis (PCA) is the one of unsupervised subspace method, which is used to reduce multidimensional data sets to lower dimensions for analysis. Let **A** be the m by n matrix of pixels intensity of the image and the image vector, γ, is the vector of **A** which was previously transformed by column-stack vectorization. Thus, the dimension of γ is $mn \times 1$. The average of γ can be found as

$$\psi = \frac{1}{N}\sum_{i=1}^{N}\gamma_i, \tag{2.1}$$

where N is the number of training images. The zero-mean normalization is applied to all image vectors by

$$\phi_i = \gamma_i - \Psi, i = 1, 2, 3, ..., M, \tag{2.2}$$

where ϕ_i is the i^{th} zero-mean normalized of γ_i. The covariance matrix, **C**, of these image vectors can be calculated as

$$\mathbf{C} = \Phi\Phi^T, \tag{2.3}$$

where Φ = $[\phi_1\ \phi_2\ ...\ \phi_N]$

The PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the principal component directions. This transformation is therefore equivalent to finding the eigenvalue decomposition of the matrix **C**.

According to the dimension of ϕ, the dimension of **C** will be $mn \times mn$ which can be normally quite large for calculating the eigenvalue decomposition. The number of training samples is normally smaller than the dimension of ϕ then the non-zero eigenvalues of this covariance matrix can be found in another way via a new matrix.

$$\mathbf{L} = \Phi^T\Phi, \tag{2.4}$$

The dimension of $\mathbf{L}$ is only $m \times n$, thus the eigenvalue decomposition of $\mathbf{L}$ can be done easier than $\mathbf{C}$. The eigenvalue decomposition of $\mathbf{L}$,

$$\mathbf{L} = \mathbf{F}\Lambda\mathbf{F}^T, \tag{2.5}$$

where Λ is the diagonal matrix which contains the eigenvalues of $\mathbf{L}$ and $\mathbf{F}$ contains a set of eigenvectors of $\mathbf{L}$. Finally, the eigenvectors of $\mathbf{C}$ which correspond to the non-zero eigenvalues of $\mathbf{C}$ can be determined by

$$\mathbf{U} = \Phi\mathbf{F}, \tag{2.6}$$

where $\mathbf{U}$ is the matrix that contains a set of eigenvectors of $\mathbf{C}$.

The eigenvector associated with the largest eigenvalue has the same direction as the first principal component, the eigenvector associated with the second largest eigenvalue determines the direction of the second principal component, and so on. Since the lower-order principal components often contain the most important aspects of the data, the dimension of projected space can be reduced by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones.

2.2 Subspace Learning based on Tensor Analysis

Recently, multilinear algebra, the algebra of higher-order tensors, was applied for analyzing the multifactor structure of image ensembles [59]. Vasilescu and Terzopoulos have proposed a novel face representation algorithm called Tensorface [60]. Tensorface represents the set of face images by a higher-order tensor and extends traditional PCA to higher-order tensor decomposition. In this way, the multiple factors related to expression, illumination and pose can be separated from different dimensions of the tensor. However, Tensorface still considers each face image as a vector instead of 2-dimensional (2D) tensor. Thus, Tensorface is computationally expensive. Moreover, it does not encode discriminating information, thus it is not optimal for recognition.

2.2.1 Tensor PCA

Let $\mathbf{A} \in \mathbf{R}^{m \times n}$ denote an image of size $m \times n$. Mathematically, $\mathbf{A}$ can be thought of as the 2^{nd} order tensor (or, 2-tensor) in the tensor space $\mathbf{R}^m \otimes \mathbf{R}^n$. Let $(\mathbf{u}_1, ..., \mathbf{u}_m)$ be a set of orthonormal basis functions of $\mathbf{R}^m$. Let $(\mathbf{v}_1, ..., \mathbf{v}_n)$ be a set of orthonormal basis functions of $\mathbf{R}^n$. Thus, an 2-tensor $\mathbf{A}$ can be uniquely written as:

$$\mathbf{A} = \sum_{ij}(\mathbf{u}_i^T\mathbf{A}\mathbf{v}_j)\mathbf{u}_i\mathbf{v}_j^T, \tag{2.7}$$

This indicates that $\mathbf{u}_i\mathbf{v}_j^T$ forms a basis of the tensor space $\mathbf{R}^m \otimes \mathbf{R}^n$. Define two matrices $U = [\mathbf{u}_1, ..., \mathbf{u}_{l1}] \in \mathbf{R}^{m \times l_1}$ and $V = [\mathbf{v}_1, ..., \mathbf{v}_{l2}] \in \mathbf{R}^{n \times l_2}$. Let $\mathbf{U}$ be a subspace of $\mathbf{R}^m$ spanned by $\mathbf{u}_{i}{}_{i=1}^{l1}$ and $\mathbf{V}$ be a subspace of $\mathbf{R}^n$ spanned by $\mathbf{v}_{j}{}_{j=1}^{l2}$. Thus, the tensor product $\mathcal{U}\otimes\mathcal{V}$ is a subspace of $\mathbf{R}^m\otimes\mathbf{R}^n$. The projection of $\mathbf{A} \in \mathbf{R}^{m\times n}$ onto the space $\mathcal{U}\otimes\mathcal{V}$ is $\mathbf{Y} = \mathbf{U}^T\ \mathbf{A}\ \mathbf{V} \in \mathbf{R}^{l_1 \times l_2}$

Suppose we have N images, $\mathbf{A}_1$,..., $\mathbf{A}_N \in \mathbf{R}^{m\times n}$. These images belong to k categories $C_1, ..., C_k$. For the i-th category, there are n_i images. The mean of each category $\mathbf{M}_i^{\mathbf{A}}$ is computed by taking the average of $\mathbf{A}$ in category i, i.e.,

$$\mathbf{M}_i^{(\mathbf{A})} = \frac{1}{n_i}\sum_{j\in C_i}\mathbf{A}_j \tag{2.8}$$

and the global mean $\mathbf{M}^{(\mathbf{A})}$ is defined as

$$\mathbf{M}^{(\mathbf{A})} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{A}_j. \tag{2.9}$$

Let $\mathbf{Y}_i = \mathbf{U}^T\mathbf{A}_i\mathbf{V} \in \mathbb{R}^{l_1 \times l_2}$. Likewise, we can define

$$\mathbf{M}_i^{(\mathbf{Y})} = \frac{1}{n_i}\sum_{j\in C_i}\mathbf{Y}_j \tag{2.10}$$

and

$$\mathbf{M}^{(\mathbf{Y})} = \frac{1}{N}\sum_{j=1}^{n}\mathbf{Y}_j, \tag{2.11}$$

It is easy to check that $\mathbf{M}_i^{\mathbf{Y}} = \mathbf{U}^T\mathbf{M}_i^{(\mathbf{A})}\mathbf{V}$ and $\mathbf{M}^{\mathbf{Y}} = \mathbf{U}^T\mathbf{M}^{(\mathbf{A})}\mathbf{V}$.

The tensor subspace learning problem aims at finding the $(l1 \times l2)$ dimensional space $\mathcal{U}\otimes\mathcal{V}$ based on the specific objective functions. Particularly, we will introduce a novel algorithms called TensorPCA in this section.

TensorPCA is fundamentally based on PCA. It tries to project the data to the tensor subspace of maximal variances so that the reconstruction error can be minimized. The objective function of TensorPCA can be described as follows:

$$MAX_{\mathbf{U},\mathbf{V}}\sum_i ||\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})}||^2 \tag{2.12}$$

Note that we use tensor norm of the difference of two tensors to measure the distance of two images. Since order two tensor is essentially matrix, we use Frobenius norm of a matrix as our 2-d tensor norm.

Since $||\mathbf{A}||^2$ = tr($\mathbf{A}\mathbf{A}^T$), we have

$$\begin{aligned}\sum_{i=1}^{n}||\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})}||^2 &= \sum_{i=1}^{n} tr((\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})^T) \\ &= \sum_{i=1}^{n} tr(\mathbf{U}^T(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})\mathbf{V}\mathbf{V}^T(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})^T\mathbf{U}) \\ &= tr(\mathbf{U}^T\sum_{i=1}^{n}(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})\mathbf{V}\mathbf{V}^T(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})^T)\mathbf{U}) \\ &= tr(\mathbf{U}^T\mathbf{M}_{\mathbf{V}}\mathbf{V}) \end{aligned} \tag{2.13}$$

where $\mathbf{M_V} = \sum_{i=1}^{n}(\mathbf{U}^T(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})\mathbf{V}\mathbf{V}^T(\mathbf{Y}_i - \mathbf{M}^{(\mathbf{Y})})^T)\mathbf{U})$. Similarly, $||\mathbf{A}||^2 = \mathrm{tr}(\mathbf{A}\mathbf{A}^T)$, so we also have

$$\begin{aligned} &= tr(\mathbf{V}^T(\sum_{i=1}^{n}(\mathbf{A}_i - \mathbf{M}^{(\mathbf{A})})\mathbf{U}\mathbf{U}^T(\mathbf{A}_i - \mathbf{M}^{(\mathbf{A})})^T)\mathbf{V}) \\ &= tr(\mathbf{V}^T\mathbf{M_U}\mathbf{V}) \end{aligned} \tag{2.14}$$

where $\mathbf{M_U} = \sum_{i=1}^{n}((\mathbf{A}_i - \mathbf{M}^{(\mathbf{A})})^T\mathbf{U}\mathbf{U}^T(\mathbf{A}_i - \mathbf{M}^{(\mathbf{A})}))$. Thus, the optimal projection **U** should be the eigenvectors of $\mathbf{M_V}$ and the optimal projection **V** should be the eigenvectors of $\mathbf{M_U}$.

One might notice that **U** and **V** can not be computed independently. In our algorithm, we try to find an optimal coordinate system of $\mathbf{R}^m \otimes \mathbf{R}^n$. That is, we assume that both **U** and **V** are orthonormal, i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. In such case,

$$\mathbf{M}'_{\mathbf{V}} = \sum_{i=1}^{n}((\mathbf{A}_i - \mathbf{M}^{\mathbf{A}})(\mathbf{A}_i - \mathbf{M}^{\mathbf{A}})^T) \tag{2.15}$$

and

$$\mathbf{M}'_{\mathbf{U}} = \sum_{i=1}^{n}((\mathbf{A}_i - \mathbf{M}^{\mathbf{A}})^T(\mathbf{A}_i - \mathbf{M}^{\mathbf{A}})). \tag{2.16}$$

It is clear that $\mathbf{M}'_{\mathbf{V}}$ no longer depends on **V** , and $\mathbf{M}'_{\mathbf{U}}$ no longer depends on **U**. Therefore, the matrix **U** can be simply computed as the eigenvectors of $\mathbf{M}'_{\mathbf{V}}$ and the matrix **V** can be computed as the eigenvectors of $\mathbf{M}'_{\mathbf{U}}$. Note that, both $\mathbf{M}'_{\mathbf{U}}$ and $\mathbf{M}'_{\mathbf{V}}$ are symmetric, hence their eigenvectors are orthonormal. This is consistent with our assumptions. If we try to reduce the original tensor space to a $l_1 \times l_2$ tensor subspace, we choose the first l_1 column vectors in **U** and the first l_2 column vectors in **V**.

2.3 Multilinear Principal Component Analysis (MPCA)

For the theoretically inclined reader, it should be noted that there are some recent developments in the analysis of higher order tensors, then this section introduces a new MPCA framework for tensor object dimensionality reduction and feature extraction using tensor representation. This framework is introduced from the perspective of capturing the original tensors variation. It provides a systematic procedure to determine effective representations of tensor objects. This contrasts to previous work such as those reported in [61], where vector, not tensor, representation was used, and the works reported in [59], [62], where matrix representation was utilized. Furthermore, unlike previous attempts, such as the one in [29], design issues of paramount importance in practical applications, such as the initialization, termination, convergence of the algorithm, and the determination of the subspace dimensionality, are discussed in details. The basic idea in MPCA solution to the problem of dimensionality reduction for tensor objects is introduced [33, 63].

2.3.1 Multilinear Projection of Tensor Objects

In this section, we review some basic multilinear concepts used in the MPCA framework development and introduces the multilinear projection of tensor objects for the purpose of dimensionality reduction.

Throughout this paper, the discussion is restricted to real-valued vectors, matrices, and tensors since the targeted applications, such as holistic gait recognition using binary silhouettes, involve real data only. The extension to the complex valued data sets is out of the scope of this work and it will be the focus of a forthcoming research.

An Nth-order tensor is denoted as $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \ldots \times I_N}$ for $I_n = 1, ..., N$. It is addressed by N indices $i_n, n = 1, ..., N$ and each i_n addresses the n-mode of $\mathcal{A}$. The n-mode vectors of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n while keeping all the other indices fixed. Unfolding $\mathcal{A}$ along the n-mode is denoted as $\mathbf{A}_{(n)}$ $\in \mathbf{R}^{I_n \times (I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N)}$ and the column vectors of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$ which are illustrated in Fig. 2.1. Let the set of tensors be $\{\mathcal{A}_m, m = 1, ..., M\}$ and the total scatter of these tensors is defined as

$$\Psi_{\mathcal{A}} = \sum_{m=1}^{M} \|\mathcal{A}_m - \bar{\mathcal{A}}\|^2, \tag{2.17}$$

where $\bar{\mathcal{A}}$ is the mean tensor calculated as $\bar{\mathcal{A}} = (1/M) \sum_{m=1}^{M} \mathcal{A}_m$. The n-mode total scatter matrix of these samples can be defined as

$$\mathbf{C}_{\mathbf{A}} = \sum_{m=1}^{M} (\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)})(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)})^T, \tag{2.18}$$

where $\mathbf{A}_{m(n)}$ is the n-mode unfolded of $\mathcal{A}$ and $\bar{\mathbf{A}}$ is sample mean. The n-mode unfolded matrix can be illustrated in Fig. 2.1. In Fig. 2.1, a third-order tensor can be unfolded in 1-mode vector.

2.3.2 MPCA Algorithm

A set of M tensor objects $\{\mathcal{X}_m, m = 1, ..., M\}$ is available for training with each tensor object $\mathcal{X}_m \in \mathbf{R}^{I_1 \times I_2 \times \ldots \times I_N}$ assuming values in a tensor space $\mathbf{R}^{I_1} \otimes \mathbf{R}^{I_2} \ldots \otimes \mathbf{R}^{I_N}$, where $\otimes$ denotes the Kronecker product. The main objective of MPCA is to define a multilinear transformation $\widetilde{\mathbf{U}}^{(n)}$ which denoted $I_n \times P_n$ matrix containing the orthonormal n-mode basis vectors and the matrix $\widetilde{\mathbf{U}}^{(n)}$ is nth projection matrix, $n = 1, ..., N$. It can map the original tensor space $\mathbf{R}^{I_1} \otimes \mathbf{R}^{I_2} \ldots \otimes \mathbf{R}^{I_N}$ into a tensor subspace $\mathbf{R}^{P_1} \otimes \mathbf{R}^{P_2} \ldots \otimes \mathbf{R}^{P_N}$ with ($P_n < I_n$, for n = 1,..., N):

We can define the projection of n-mode vector of $\mathcal{X}_m$ as

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \widetilde{\mathbf{U}}^{(1)^T} \times_2 \widetilde{\mathbf{U}}^{(2)^T} \ldots \times_N \widetilde{\mathbf{U}}^{(N)^T}. \tag{2.19}$$

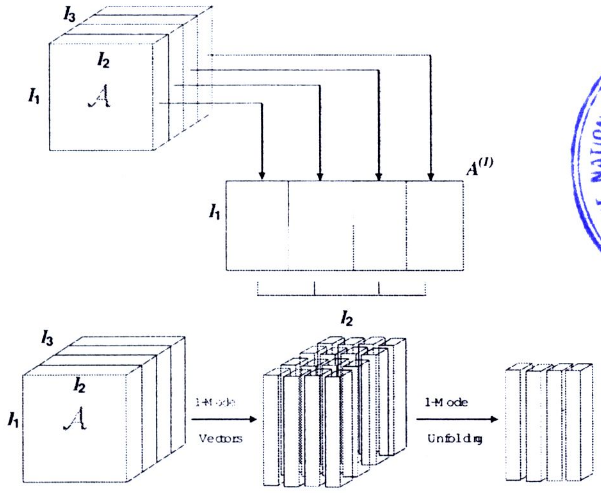

Figure 2.1: The visual illustration of the 1-mode unfolding of a third-order tensor $\mathcal{A}$ to matrix $\mathbf{A}^{(1)}$

The tensor $\mathcal{Y}_m$ can capture most of the variations observed in the original tensor objects, assuming that these variations are measured by the total tensor scatter. The objective of MPCA is the determination of the N projection matrices $\widetilde{\mathbf{U}}^{(n)}$ that maximize the total tensor scatter $\Psi_{\mathcal{Y}}$ as

$$\{\widetilde{\mathbf{U}}^{(n)}, n = 1, ..., N\} = \underset{\widetilde{\mathbf{U}}^{(1)},\widetilde{\mathbf{U}}^{(2)},...,\widetilde{\mathbf{U}}^{(N)}}{\operatorname{argmax}} \quad \Psi_{\mathcal{Y}}. \tag{2.20}$$

There is no known optimal solution which allows for the simultaneous optimization of the N projection matrices. Since the projection to an Nth-order tensor subspace consists of N projections to N vector subspaces, N optimization subproblems can be solved by finding $\widetilde{\mathbf{U}}^{(n)}$ that maximizes the scatter in the n-mode vector subspace.

The dimensionality P_n for each mode is assumed to be known or predetermined. The matrix $\widetilde{\mathbf{U}}^{(n)}$ consists of the P_n eigenvectors corresponding to the largest P_n eigenvalues of the matrix and it can be expressed as

$$\Phi^{(n)} = \sum_{m=1}^{M} (\mathbf{X}_m^{(n)} - \bar{\mathbf{X}}^{(n)}) \cdot \widetilde{\mathbf{U}}_{\Phi^{(n)}} \cdot \widetilde{\mathbf{U}}_{\Phi^{(n)}}^{T} \cdot (\mathbf{X}_m^{(n)} - \bar{\mathbf{X}}^{(n)})^T, \tag{2.21}$$

where

$$\widetilde{\mathbf{U}}_{\Phi^{(n)}} = (\widetilde{\mathbf{U}}^{(n+1)} \otimes \widetilde{\mathbf{U}}^{(n+2)} \otimes ... \otimes \widetilde{\mathbf{U}}^{(1)} \otimes \widetilde{\mathbf{U}}^{(2)} \otimes ... \widetilde{\mathbf{U}}^{(n-1)}). \tag{2.22}$$

The optimization of $\widetilde{\mathbf{U}}_{\Phi^{(n)}}$ depends on the projections in other modes, so there is no closed-form solution. Therefore an iterative procedure is proposed to solve (2.22). The projection matrices are calculated one by one, keeping all the others fixed (local optimization).

2.3.3 Full Projection

The term full projection refers to the multilinear projection for MPCA with $P_n = I_n$ for $n = 1, ..., N$. In this case, we can see that $\widetilde{\mathbf{U}}_{\Phi^{(n)}} \cdot \widetilde{\mathbf{U}}_{\Phi^{(n)}}^{T}$ is an identity matrix. As a

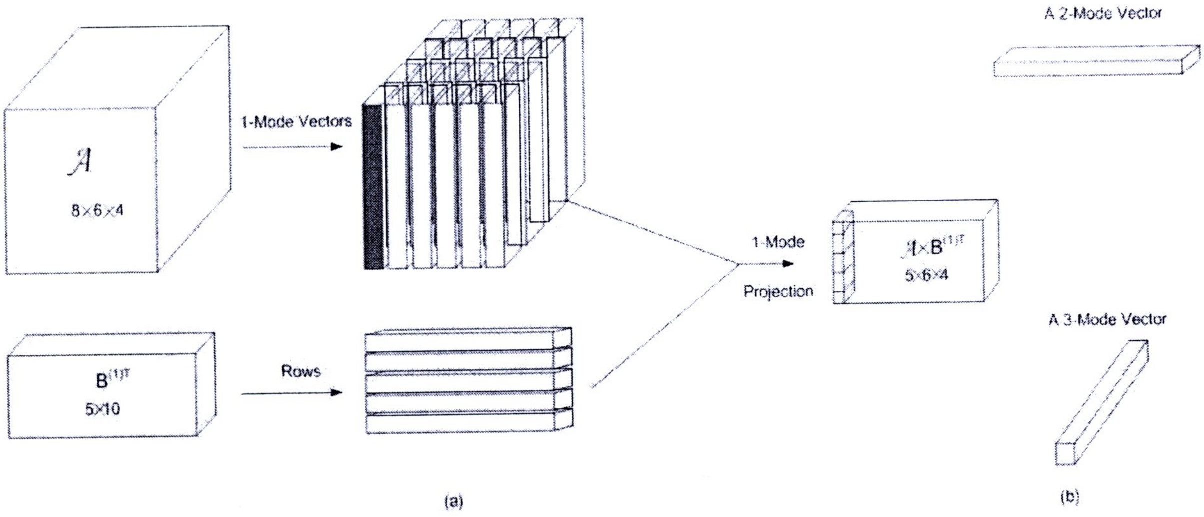

Figure 2.2: Visual illustration of multilinear projection: (a) projection in the 1-mode vector space and (b) 2-mode and 3-mode vectors.

result, $\Phi^{(n)}$ reduces to $\Phi^{(n)*} = \sum_{m=1}^{M}(\mathbf{X}_{m(n)}-\bar{\mathbf{X}}_{(n)})(\mathbf{X}_{m(n)}-\bar{\mathbf{X}}_{(n)})^T$, with $\Phi^{(n)}$ determined by the input tensor samples only and independent of other projection matrices. The optimal $\tilde{\mathbf{U}}^{(n)} = \mathbf{U}^{(n)*}$ is then obtained as the matrix comprised of the eigenvectors of $\Phi^{(n)*}$ directly without iteration, and the total scatter $\Psi_{\mathcal{X}}$ in the original data is fully captured. However, there is no dimensionality reduction through this full projection. From the properties of eigendecomposition, it can be concluded that if all eigenvalues per mode are distinct, the full projection matrices (corresponding eigenvectors) are also distinct and that the full projection is unique (up to sign).

To interpret the geometric meanings of the n-mode eigenvalues, the total scatter tensor $\mathcal{Y}^*_{var} \in \mathbf{R}^{I_1 \times I_2 \times ... \times I_N}$ of the full projection is introduced as an extension of the total scatter matrix. Each entry of the tensor $\mathcal{Y}^*_{var}$ is defined as

$$\mathcal{Y}^*_{var}(i_1, i_2, ..., i_N) = \sum_{m=1}^{M}[(\mathcal{Y}^*_m - \bar{\mathcal{Y}}^*)(i_1, i_2, ..., i_N)]^2 \tag{2.23}$$

where

$$\mathcal{Y}^*_m(i_1, i_2, ..., i_N) = \mathcal{X}_m \times_1 \mathbf{U}^{(1)} *^T ... \times_N \mathbf{U}^{(N)*^T} \tag{2.24}$$

and

$$\bar{\mathcal{Y}}^* = (1/M)\sum_{m=1}^{M}\mathcal{Y}^*_m \tag{2.25}$$

Using the previous definition, it can be shown that for the so-called full projection (P_n=I_n for all n), the i_nth n-mode eigenvalue $\lambda^{(n)*}_{i_n}$ is the sum of all the entries of the i_nth n-mode slice of $\mathcal{Y}^*_{var}$

$$\lambda^{(n)*}_{i_n} = \sum_{i=1}^{I_1} ... \sum_{i_{n-1}=1}^{I_{n-1}} \sum_{i_{n+1}=1}^{I_{n+1}} ... \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{var}(i_1, ..., i_{n-1}, i_n, i_{n+1}, ..., i_N) \tag{2.26}$$

In this paper, the eigenvalues are all arranged in a decreasing order. Fig. 2.3 shows visually what the n-mode eigenvalues represent. In this graph, third-order tensors, e.g., short sequences (four frames) of images with size 6×5, are projected to a tensor space of size $6 \times 5 \times 4$ (full projection) so that a total scatter tensor $\mathcal{Y}^*_{var} \in \mathbf{R}^{6 \times 5 \times 4}$ is obtained.

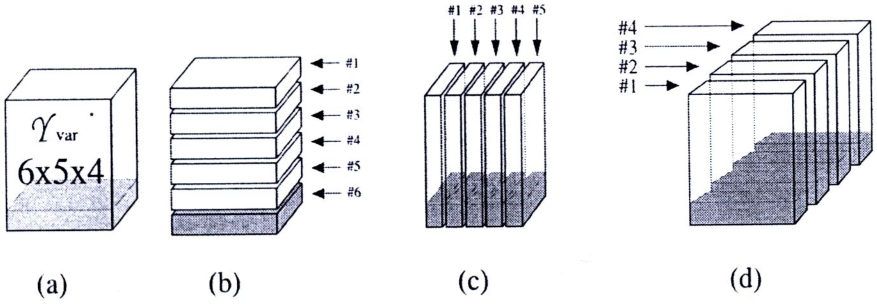

Figure 2.3: Visual illustration of (a) total scatter tensor, (b) 1-mode eigenvalues, (c) 2-mode eigenvalues, and (d) 3-mode eigenvalues.

2.3.4 MPCA Versus PCA and 2-D PCA Solutions

It is not difficult to see that the MPCA framework generalizes not only the classical PCA solution but also a number of the so-called 2-D PCA algorithms.

Indeed, for $N = 1$, the input samples are vectors $\mathbf{x}_m \in \mathbf{R}^{I_1}$. There is only one mode and MPCA is reduced to PCA. For dimensionality reduction purposes, only one projection matrix $\mathbf{U}$ is needed in order to obtain $\mathbf{y}_m = \mathbf{x}_m \times_1 \mathbf{U} = \mathbf{U}^T \mathbf{x}_m$. In this case, there is only one $\Phi^{(n)} = \Phi^{(1)} = \sum_{m=1}^{M} (\mathbf{x}_m \text{-} \bar{\mathbf{x}}) \cdot (\mathbf{x}_m - \bar{\mathbf{x}})^T$, which is the total scatter matrix of the input samples in PCA [36]. The projection matrix maximizing the total scatter (variation) in the projected space is determined from the eigenvectors of $\Phi^{(1)}$. Thus, MPCA subsumes PCA.

In the so-called 2-D PCA solutions, input samples are treated as matrices, in other words second-order tensors. Two (left and right) projection matrices are sought to maximize the captured variation in the projected space. The proposed MPCA algorithm is equivalent to the 2-D PCA solution of [62], with the exception of the initialization procedure and termination criterion. Other 2-D PCA algorithms such as those discussed in [64] can be viewed as variations of the method in [62] and thus they can be considered special cases of MPCA when second-order tensors are considered.

2.4 TensorFaces: Multilinear Analysis of Facial Images

Multilinear algebra offers a natural approach to the analysis of the multifactor structure of image ensembles and to addressing the difficult problem of disentangling the constituent factors or modes then an image formation depends on scene geometry, viewpoint, and illumination conditions [60]. We apply multilinear analysis to the facial image data using the N-mode decomposition algorithm described in Section 2.3.

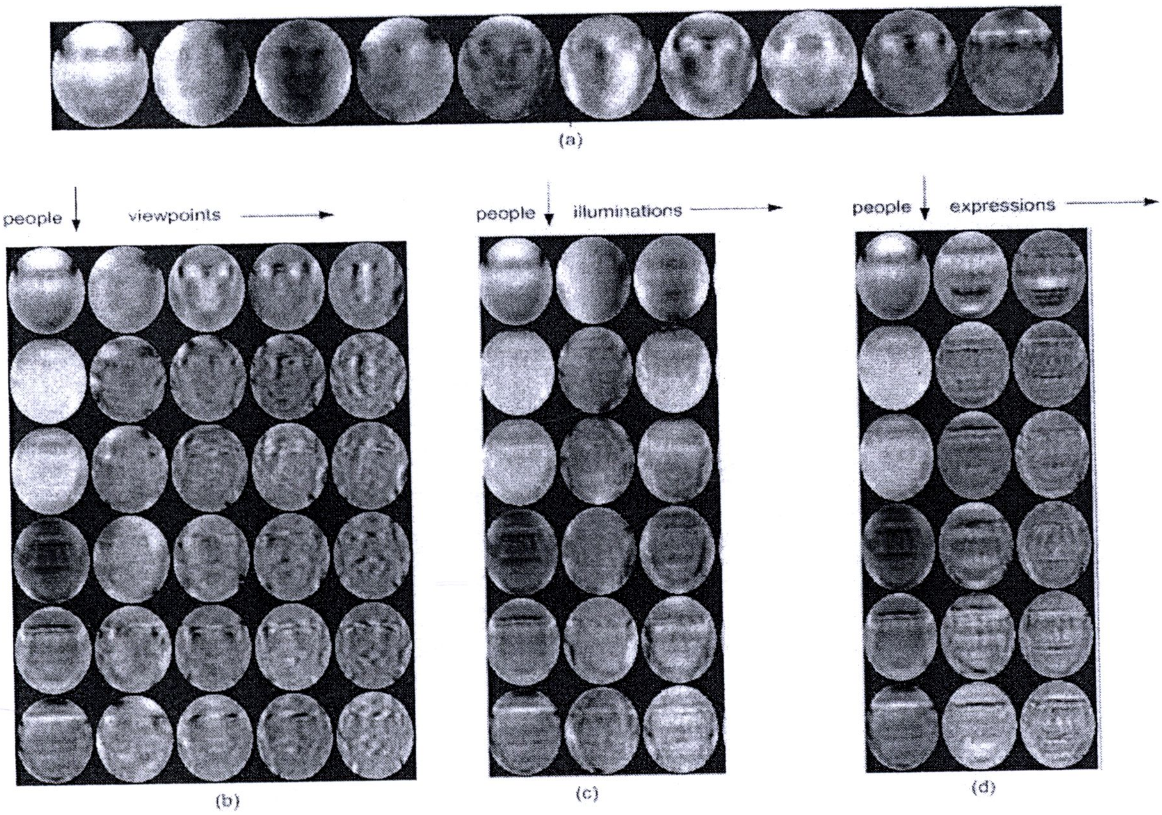

Figure 2.4: Some of the basis vectors resulting from the multilinear analysis of the facial image data tensor $\mathcal{D}$ [1,2].

In a concrete application of our multilinear image analysis technique, we employ the Weizmann face database of 28 male subjects photographed in 15 different poses under 4 illuminations performing 3 different expressions. The 5-mode decomposition of $\mathcal{D}$ is

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{people} \times_2 \mathbf{U}_{views} \times_3 \mathbf{U}_{illums} \times_4 \mathbf{U}_{expres} \times_5 \mathbf{U}_{pixels}. \tag{2.27}$$

Using a global rigid optical flow algorithm, we roughly aligned the original 512×352 pixel images relative to one reference image. The images were then decimated by a factor of 3 and cropped as shown in Fig. 2.4, yielding a total of 7943 pixels per image within the elliptical cropping window. Our facial image data tensor D is a $28 \times 5 \times 3 \times 3 \times 7943$ tensor. The number of modes is $N = 5$. The core tensor $\mathcal{Z}$ governs the interaction between the factors represented in the 5 mode matrices: The 28×28 mode matrix $\mathbf{U}_{people}$ spans the space of people parameters, the 5×5 mode matrix $\mathbf{U}_{views}$ spans the space of viewpoint parameters, the 3×3 mode matrix $\mathbf{U}_{illums}$ spans the space of illumination parameters and the 3×3 mode matrix $\mathbf{U}_{expres}$ spans the space of expression parameters. The 7943×7943 mode matrix $\mathbf{U}_{pixels}$ orthonormally spans the space of images.

Our multilinear analysis, which we call TensorFaces, subsumes linear, PCA analysis or conventional eigenfaces. Each column of $\mathbf{U}_{pixels}$ is an eigenimage. These eigenimages are identical to conventional eigenfaces [13, 17], since the former were computed by performing an SVD on the mode-5 flattened data tensor $\mathcal{D}$ which yields the matrix $\mathbf{D}_{pixels}$ whose columns are the vectorized images. To further show mathematically that PCA is a special case of our multilinear analysis, we write the latter in terms of matrix notation. A matrix representation of the N-mode SVD can be obtained by unfolding $\mathcal{D}$ and $\mathcal{Z}$ as follows:

$$\mathbf{D}_{(n)} = \mathbf{U}_{(n)}\mathbf{Z}_{(n)}(\mathbf{U}_{(n-1)} \otimes ... \otimes \mathbf{U}_{(1)} \otimes \mathbf{U}_{(N)} \otimes ... \otimes \mathbf{U}_{(n+2)} \otimes \mathbf{U}_{(n+1)})^T, \tag{2.28}$$

Using 2.28 we can express the decomposition of $\mathcal{D}$ as

$$\underbrace{\mathbf{D}_{(pixels)}}_{imagedata} = \underbrace{\mathbf{U}_{(pixels)}}_{basisvectors} \underbrace{(\mathbf{Z}_{(pixels)}\mathbf{U}_{(expres)} \otimes \mathbf{U}_{(illums)} \otimes \mathbf{U}_{(views)} \otimes \mathbf{U}_{(people)})^T}_{coefficients} \tag{2.29}$$

The above matrix product can be interpreted as a standard linear decomposition of the image ensemble, where the mode matrix $\mathbf{U}_{(pixels)}$ is the PCA matrix of basis vectors and the associated matrix of coefficients is obtained as the product of the flattened core tensor times the Kronecker product of the people, viewpoints, illuminations, and expressions mode matrices. Thus, as we stated above, our multilinear analysis subsumes linear, PCA analysis.

The advantage of multilinear analysis is that the core tensor Z can transform the eigenimages present in the matrix $\mathbf{U}_{(pixels)}$ into eigenmodes, which represent the principal axes of variation across the various modes (people, viewpoints, illuminations, expressions) and represents how the various factors interact with each other to create an image. This is accomplished by simply forming the product $\mathcal{Z}\times 5\ \mathbf{U}_{(pixels)}$. By contrast, PCA basis vectors or eigenimages represent only the principal axes of variation across images. To demonstrate, Fig. 2.4 illustrates in part the results of the multilinear analysis of the facial image tensor $\mathcal{D}$. Fig. 2.4(a) shows the first 10 PCA eigenimages contained in $\mathbf{U}_{(pixels)}$. Fig. 2.4(b) illustrates some of the eigenmodes in the product $\mathcal{Z}\times 5\ \mathbf{U}_{(pixels)}$. A few of the lower-order eigenmodes are shown in the three arrays. The labels at the top of each array indicate the names of the horizontal and vertical modes depicted by the array. Note that the basis vector at the top left of each panel is the average over all people, viewpoints, illuminations, and expressions, and that the first column of eigenmodes (people mode) is shared by the three arrays.

2.5 Color Image Processing

Over the last three decades, we have seen several important contributions in the field of color image processing. While there have been many early papers that address various aspects of color images, it is only recently that a more complete understanding of color vision, colorimetry, and color appearance has been applied to the design of imaging systems and image processing methodologies. The first contributions in this area were those that changed the formulation of color signals from simple algebraic equations to matrix representation [65], [66], [67]. More powerful use of the matrix algebraic representation was presented in [68], where set theoretic methods were introduced to color processing. The first overview extending signal processing concepts to color was presented in IEEE Signal

Processing Magazine in 1993 [69]. This was followed by a special issue on color image processing in IEEE Transactions on Image Processing in July 1997, where a complete review of the state of the art at that time was found in [70, 71].

At this point, the focus of the issue shifts from hardware and system centric to image processing techniques that form the backbone of many color systems and applications. The first of these is discussed in Detection and Classification of Edges in Color Images [72], where the emphasis is placed on detecting discontinuities, i.e., transitions from one region to another, instead of similarities in a given image. One of the most fundamental steps in many applications and in the design of image processing techniques is to ensure that a given image is optimized for noise and enhanced in quality. This is outlined in the article Vector Filtering for Color Imaging [73], where the authors discuss and compare the various techniques outlining their strength, areas of improvements, and future research directions. Finally, the article titled Digital Color Halftoning [74] provides a complete review of the methods employed by printers to reproduce color images and the challenges they face in ensuring that these images are free of visual artifacts.

2.5.1 Mathematical Definition of Color Matching

A vector space approach to describing color is useful for expressing and solving complex problems in color imaging. For this reason, we will use this notation to describe the fundamentals of color matching. Let the $N \times 3$ matrix $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$ represent the response of the eye, where the N vectors, $\mathbf{s}_i$, correspond to the response of the ith type sensor (cone) in Fig. 2.5. A given visible spectrum can be represented by an N vector, $\mathbf{f}$, a function whose value is radiant energy. Hence, the response of the sensors to the input spectrum is a three vector, $\mathbf{c}$, obtained by

$$\mathbf{c} = \mathbf{S}^T\mathbf{f} \tag{2.30}$$

Two visible N-vectors spectra $\mathbf{f}$ and $\mathbf{g}$ are said to have the same color if they appear the same to a human observer. In our linear model, this implies that if $\mathbf{f}$ and $\mathbf{g}$ represent different spectral distributions, they portray equivalent colors if

$$\mathbf{S}^T\mathbf{f} = \mathbf{S}^T\mathbf{g} \tag{2.31}$$

From this, it can be easily seen that many different spectra can result in the same color appearance to a given observer. This fascinating phenomena is known as metamerism (meh tam er ism), and the two spectra are termed as metamers. In essence, metamerism is basically color aliasing and can be described by generalizing the well-known Shannon sampling theorem frequently encountered in communications and digital signal processing. It should be noted, however, that the level of metamerism may vary across various observers, dependent on their individual cone sensitivities.

In practice, it is desirable to have a matrix of color matching functions that are non-negative, so they can be physically realized as optical filters. This problem was addressed by the Commission Internationale de lEclairage (CIE), in 1931, yielding the XYZ color matching functions shown in Fig. 2.6 as solids lines. Hence, the matrix **A** can now be used to represent these functions. The Y value was chosen to be the luminous efficiency function, making it equivalent to the photometric luminance value. This standardization led to the precise definition of colorimetric quantities, such as tristimulus values and chromaticity.

The term tristimulus values refers to the vector of values obtained from a radiant spectrum, **r**, by $\mathbf{t} = [X, Y, Z]^T = \mathbf{A}^T\mathbf{r}$ (we recognize the inconsistency of denoting the elements of **t** by X, Y, Z, but since the color world still uses the X, Y, Z terms, we use it here). The chromaticity is then obtained by normalizing the tristimulus values yielding

$$\begin{aligned} x &= X/(X+Y+Z) \\ y &= Y/(X+Y+Z) \\ z &= Z/(X+Y+Z) \end{aligned} \tag{2.32}$$

Since $x + y + z = 1$, any two chromaticity coordinates are sufficient to characterize the chromaticity of a spectrum. In general, as a matter of convention, the x and y terms are used as the standard.

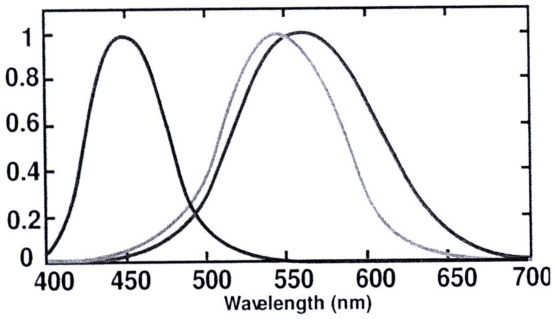

Figure 2.5 Cone sensitivities.

2.5.2 Mathematics of Color Reproduction

To reproduce a color image, it is necessary to generate new vectors in N space (spectral space) from those obtained by a given multispectral sensor. Since the eye can be represented as a three-channel sensor, it is most common for a multispectral sensor to use three types of filters. Hence, the characteristics of the resulting multispectral response functions associated with the input and output devices are critical aspects for color reproduction. Output devices can be characterized as being additive or subtractive. Additive devices, such as cathode ray tubes (CRTs), produce light of varying spectral composition as viewed by the human observer. On the other hand, subtractive devices, such as ink-jet printers, produce filters that attenuate portions of an illuminating spectrum. We will discuss both types in the following, clearly highlighting their differences.

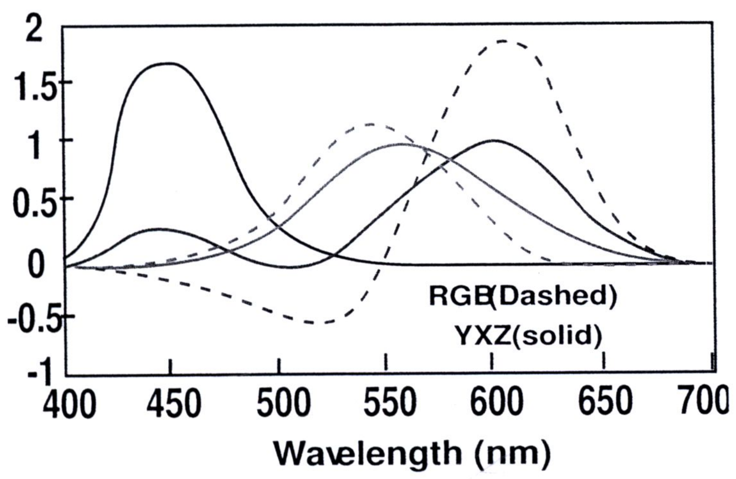

Figure 2.6: CIE RGB and XYZ color matching functions: RGB are shown in dashed lines, and XYZ are shown in solid lines.

2.5.3 Additive Color Systems

In additive devices, various colors are generated by combining light sources with different wavelengths. These light sources are known as primary. An example of this is illustrated in Fig. 2.7. In the Fig. 2.7, it can be easily seen that cyan, magenta, yellow, and white are generated by combining blue and green; red and blue; red and green; and red, green, and blue, respectively. The red, green, and blue channels of an example color image are also shown for illustration purposes. Other colors can be generated by varying the intensities of the red, green, and blue primaries. For instance, the screen of a television, or CRT, is covered with phosphoric dots that are clustered in groups. Each group contains these primary colors: red, green, and blue, which are combined in a weighted fashion to produce a wide range of colors. Additive color systems are characterized by their corresponding multispectral output response. For instance, a three-color monitor is represented by the $N \times 3$ matrix, **E**, which serves the same purpose as the primaries in the color matching experiment. The amount of each primary is controlled by a three-vector **c**. The spectrum of the output is then computed as follows:

$$\mathbf{f} = \mathbf{E}\mathbf{c}, \tag{2.33}$$

Hence, the tristimulus values associated with a standard observer who is viewing the screen are given by

$$\mathbf{t} = \mathbf{A}^T\mathbf{f} = \mathbf{A}^T\mathbf{E}\mathbf{c}, \tag{2.34}$$

There are several challenges that need to be considered when dealing with additive systems. One is to choose the control values so that the output matches the intended target values. This is not feasible for all possible colors due to the power limitations of the output device.

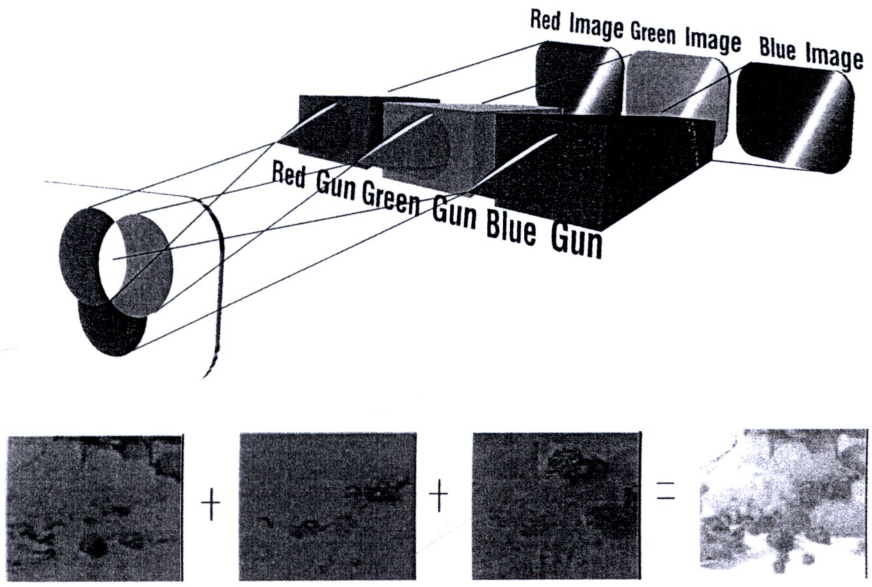

Figure 2.7 Additive color system.

Furthermore, the control values cannot be negative, i.e., we cannot produce negative light. In addition, we have the question of estimating the best values of c from some recorded data.

2.5.4 Subtractive Color Systems

Subtractive systems are characterized by the fundamental property that color is obtained by removing (subtracting) selected portions of a source spectrum. This is illustrated in Fig. 2.8, where cyan, magenta, and yellow colorants are used to absorb, in this respect, the red, green, and blue spectral components from white light. The cyan, magenta, and yellow channels of a color image are also shown for illustration purposes. Hence, each colorant absorbs its complementary color and transmits the remainder of the spectrum. The amount of light removed, by blocking or absorption, is determined by the concentration and material properties of the colorant. The color is generally presented on a transmissive medium like transparencies or on a reflective medium like paper. While the colorant for subtractive systems may be inks, dyes, wax, or toners, the same mathematical representation outlined in previously can be used to approximate them. The main property of interest for imaging in subtractive systems is the optical density. The transmission of an optically transmissive material is defined as the ratio of the intensity of the light that passes through the material to the intensity of the source. This is illustrated by

$$T = \frac{I_{out}}{I_{in}}. \tag{2.35}$$

As a result, the optical density is defined by

$$d = -log_{10}(T). \tag{2.36}$$

and is related to the physical density of the material. The inks can be characterized by their density spectra, the $N \times 3$ matrix $\mathbf{D}$. Hence, the spectrum that is seen by the observer is the product of an illumination source, the transmission of the ink, and the reflectance of the paper. Since the transmissions of the individual inks reduce the light proportionately, the output at each wavelength, λ, is given by

$$g(\lambda) = l(\lambda)t_1(\lambda)t_2(\lambda)t_3(\lambda) \tag{2.37}$$

where $t_i(\lambda)$ is the transmission of the ith ink and $l(\lambda)$ is the intensity of the illuminant. For simplification, the reflectance of the paper is assumed perfect and is assigned the value of 1.0. The transmission of a particular colorant is related logarithmically to the concentration of the ink on the page. The observed spectrum is obtained mathematically by

$$g = \mathbf{L}[10^{-\mathbf{Dc}}] \tag{2.38}$$

where $\mathbf{L}$ is a diagonal matrix representing an illuminant spectrum and $\mathbf{c}$ is the concentration of the colorant. The concentration values are held between zero and unity and the matrix of density spectra, $\mathbf{D}$, represents the densities at the maximum concentration. The exponential term is computed componentwise, i.e.,

$$10^{r} = [10^{r_1}10^{r_2}...10^{r_N}]^T. \tag{2.39}$$

This simple model ignores nonlinear interactions between colorant layers. For a reflective medium, the model requires an additional diagonal matrix, which represents the reflectance spectrum of the surface. For simplicity, this can be conceptually included in the illuminant matrix $\mathbf{L}$. The actual process for subtractive color reproduction is much more complicated and cannot, in general, be comprehensively modeled by the equations described here. Hence, these systems are usually characterized by look-up tables (LUTs) that capture their input-output relationships empirically. The details of handling device characterizations via LUTs are described in [75].

2.5.5 Color Spaces

The proper use and understanding of color spaces is necessary for the development of color image processing methods that are optimal for the human visual system. Many algorithms have been developed that process in an RGB color space without ever defining this space in terms of the CIE color matching functions, or even in terms of the spectral responses of R, G, and B. Such algorithms are nothing more than multichannel image processing techniques applied to a three-band image, since there is no accounting for the perceptual aspect of the problem. To obtain some relationship with the human visual system, many color image processing algorithms operate on data in hue, saturation, lightness (HSL) spaces. Commonly, these spaces are transformations of the aforementioned RGB color space and hence have no visual meaning until a relationship is established back to a CIE color space. To further confuse the issue, there are many variants of these color spaces, including hue saturation

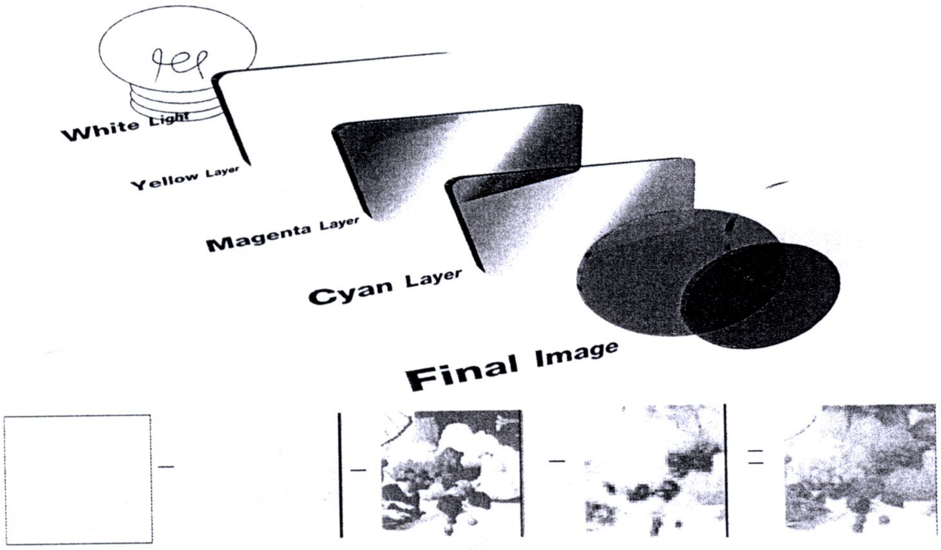

Figure 2.8 Subtractive color system.

value (HSV), hue saturation intensity (HSI), and hue chroma intensity (HCI), some of which have multiple definitions in terms of transforming from RGB. Since color spaces are of such importance and a subject of confusion, we will discuss them in details.

There are two primary aspects of a color space that make it more desirable and attractive for use in color devices: 1) its computational expediency in transforming a given set of data to the specific color space and 2) conformity of distances of color vectors in the space to that observed perceptually by a human subject, i.e., if two colors are far apart in the color space, they look significantly different to an observer with normal color vision. Unfortunately, these two criteria are antagonistic. The color spaces that are most suited for measuring perceptual differences require complex computation, and vice versa.

2.5.6 Uniform Color Spaces

It is well publicized that the psychovisual system is nonlinear and extremely complex. It cannot be modeled by a simple function. The sensitivity of the system depends on what is being observed and the purpose of the observation. A measure of sensitivity that is consistent with the observations of arbitrary scenes is well beyond our present capabilities. However, much work has been done to determine human color sensitivity in matching two color fields that subtend only a small portion of the visual field. In fact, the color matching functions (CMFs) of Figure 2.6 are more accurately designated by the solid angle of the field of view that was used for their measurement. A two-degree field of view was used for those CMFs.

It is well known that mean square error is, in general, a poor measure of error in any phenomenon involving human judgment. A common method of treating the nonuniform error problem is to transform the space into one where Euclidean distances are more closely correlated with perceptual ones. As a result, the CIE recommended, in 1976, two transformations in an attempt to standardize measures in the industry. Neither of these standards achieve the goal of a uniform color space. However, the recommended transformations do reduce the variations in the sensitivity ellipses by a large degree. In addition, they have another major feature in common: the measures are made relative to a reference white point. By using the reference point, the transformations attempt to account for the adaptive characteristics of the visual system. The first of these transformation is the CIELAB space defined by

$$\tag{2.40}$$

$$L^* = 116(\frac{Y}{Y_n})^{\frac{1}{3}} - 16 \tag{2.41}$$

$$a^* = 500[(\frac{X}{X_n})^{\frac{1}{3}} - (\frac{Y}{Y_n})^{\frac{1}{3}}] \tag{2.42}$$

$$b^* = 200[(\frac{Y}{Y_n})^{\frac{1}{3}} - (\frac{Z}{Z_n})^{\frac{1}{3}}] \tag{2.43}$$

for $(\frac{X}{X_n}), (\frac{Y}{Y_n}), (\frac{Z}{Z_n}) > 0.01$. The values X_n, Y_n, Z_n are the CIE tristimulus values of the reference white under the reference illumination, and X, Y, Z are the tristimulus values, which are to be mapped to the CIELAB color space. This maps the reference white to (L^*, a^*, b^*) = (100,0,0). The requirement that the normalized values be greater than 0.01 is an attempt to account for the fact that at low illumination the cones become less sensitive and the rods (monochrome receptors) become active. Hence, a linear model is used at low light levels.