R. D. Dony "Karhunen-Loève Transform"
The Transform and Data Compression Handbook
Ed. K. R. Rao and P.C. Yip.
Boca Raton, CRC Press LLC, 2001

Chapter 1

Karhunen-Loève Transform

R.D. Dony

University of Guelph

1.1 Introduction

The goal of image compression is to store an image in a more compact form, i.e., a representation that requires fewer bits for encoding than the original image. This is possible for images because, in their "raw" form, they contain a high degree of redundant data. Most images are not haphazard collections of arbitrary intensity transitions. Every image we see contains some form of structure. As a result, there is some correlation between neighboring pixels. If one can find a reversible transformation that removes the redundancy by decorrelating the data, then an image can be stored more efficiently. The Karhunen-Loève Transform (KLT) is the linear transformation that accomplishes this.

In Section 1.2 we show how pixels are correlated in typical images. With the pixel values forming the axes of a vector space, a rotation of this space can remove this correlation. The basis vectors of the new space define the linear transformation of the data. The basis vectors of the KLT are the eigenvectors of the image covariance matrix. Its effect is to diagonalize the covariance matrix, removing the correlation of neighboring pixels.

As presented in Section 1.3, the KLT minimizes the theoretical bound on bit rate as given by the signal entropy. The entropy for both discrete random variables and continuous random processes is defined. The KLT also maximizes the coding gain defined as the ratio of the arithmetic mean of the coefficient variances to their geometric mean. Further, the effects of truncation, block size, and interblock correlation are also presented. Section 1.4 presents the results of using the KLT for a number of examples.

1.2 Data Decorrelation

Data from neighboring pixels are highly correlated for most images. Fig. 1.1 shows a typical gray scale image. The image is 512×512 pixels in size with each gray level brightness value of pixel being represented by an 8-bit value for a range of [0–255]. This particular image is commonly used in evaluations and is often referred to as the Lena image. Even with a large degree of detail in many regions, the gray level value of any given pixel tends to be similar to its neighboring pixels. To illustrate this relationship, one can plot the gray level values of pairs of adjacent pixels as shown in Fig. 1.2. Each dot represents a pixel in the image with the x coordinate being its gray level value and the y coordinate being the gray level value of its neighbor to the right. The strong diagonal relationship about the $x = y$ line clearly shows the strong correlation between neighboring pixels.

If we were to block the image into nonoverlapping 1×2 pixel blocks as shown in Fig. 1.3, we can represent an image by a collection of two-dimensional vectors $\mathbf{x}_i$. The scatter plot of this collection is equivalent to Fig. 1.2. Looking at the distributions of the values for each of the two components as shown in Fig. 1.4, we see that they are relatively wide and cover most of the 0–255 range. In fact, the distributions of each component would be quite similar to the overall distribution of individual pixels in the image.

Now, what would happen if we rotated the distribution shown in Fig. 1.2 by 45° about the center? The result is shown in Fig. 1.5. The two components are now decorrelated, i.e., knowing the value of the first component does not help in estimating the value of the second. The distributions of the new components are shown in Fig. 1.6. The first component, save for the shift and a scaling factor of $\sqrt{2}$, is still quite similar to the previous distributions — quite broad and covering most of the dynamic range of the original individual pixels. The second component, however, is quite different. It is much narrower, with a strong peak at 0. Because it has a smaller dynamic range, we could encode its value with fewer bits. So even with a decorrelation by a simple rotation of the axis, we can reduce the number of bits required for encoding an image.

In general, a process is decorrelated when, for zero mean random variables x_i and x_j, the expectation of their product, the covariance, is zero if $i \neq j$, i.e.,

$$E\left(x_i x_j\right) = \begin{cases} 0 & i \neq j \,, \\ \sigma_i^2 & i = j \,, \end{cases} \tag{1.1}$$

where $E(\cdot)$ is the expectation operator. Using vector notations, we may define the vector of the values of an image block of N pixels as

$$\mathbf{x} = [x_1 \; x_2 \; \ldots \; x_N]^T \;. \tag{1.2}$$

We can then define the covariance matrix as

$$[\mathbf{C}]_x = E\left[(\mathbf{x} - \mathbf{m})\,(\mathbf{x} - \mathbf{m})^T\right] \,, \tag{1.3}$$

FIGURE 1.1
Example "Lena" image. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

where $\mathbf{m} = E(\mathbf{x})$ is the mean. For notational convenience, we will assume zero mean input for the rest of this chapter. In practice, the mean can simply be removed from the data before processing.

We wish to find a linear transformation matrix, $[\mathbf{W}]$, whose transpose, $[\mathbf{W}]^T$, will rotate $\mathbf{x}$ to produce a diagonal covariance matrix for the transformed variable $\mathbf{y}$,

$$\mathbf{y} = [\mathbf{W}]^T \mathbf{x} \,. \tag{1.4}$$

Each column vector, $\mathbf{w}_i$, of $[\mathbf{W}]$ is a basis vector of the new space. So, alternatively, each element, y_i, of $\mathbf{y}$ is calculated as

$$y_i = \mathbf{w}_i^T \mathbf{x} \,. \tag{1.5}$$

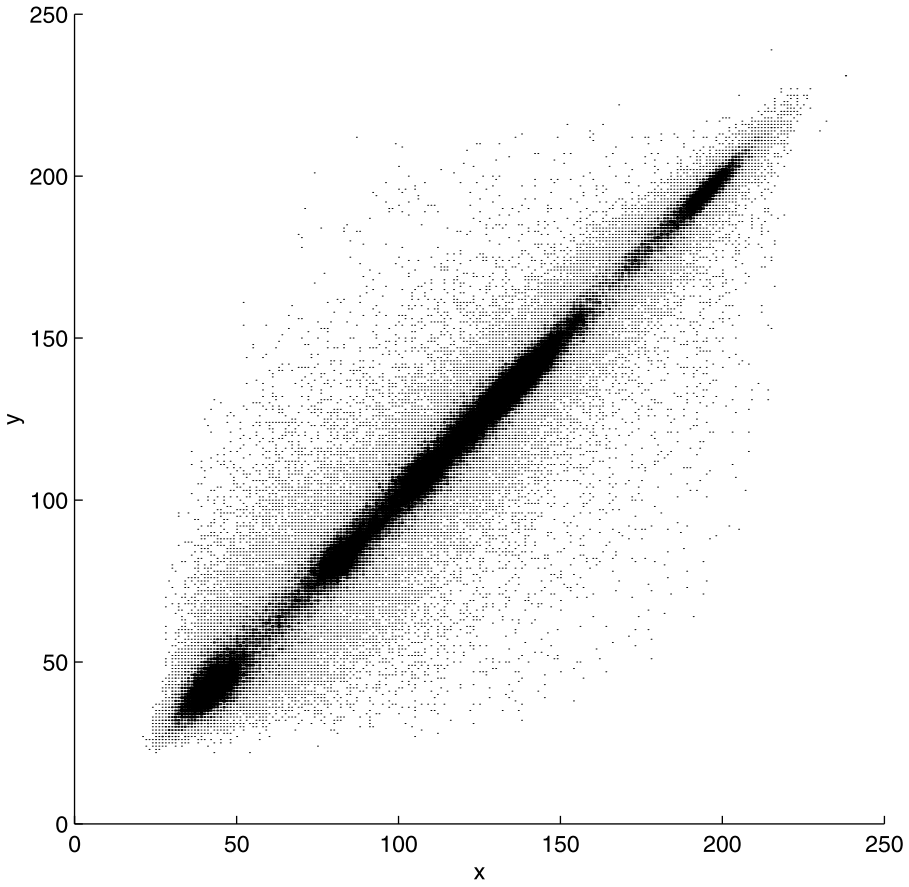

FIGURE 1.2
Scatter plot of adjacent pixel value pairs.

For simple rotations with no scaling, the matrix [**W**] must be orthonormal, that is

$$[\mathbf{W}]^T[\mathbf{W}] = [\mathbf{I}] = [\mathbf{W}][\mathbf{W}]^T \tag{1.6}$$

where [**I**] is the identity matrix. This means that the column vectors of the matrix are mutually orthogonal and are of unit norm. From Eq. (1.6), it follows that the inverse of an orthonormal matrix is simply its transpose, $[\mathbf{W}]^T = [\mathbf{W}]^{-1}$. The inverse transformation is then calculated as

$$\mathbf{x} = [\mathbf{W}]\mathbf{y} \; . \tag{1.7}$$

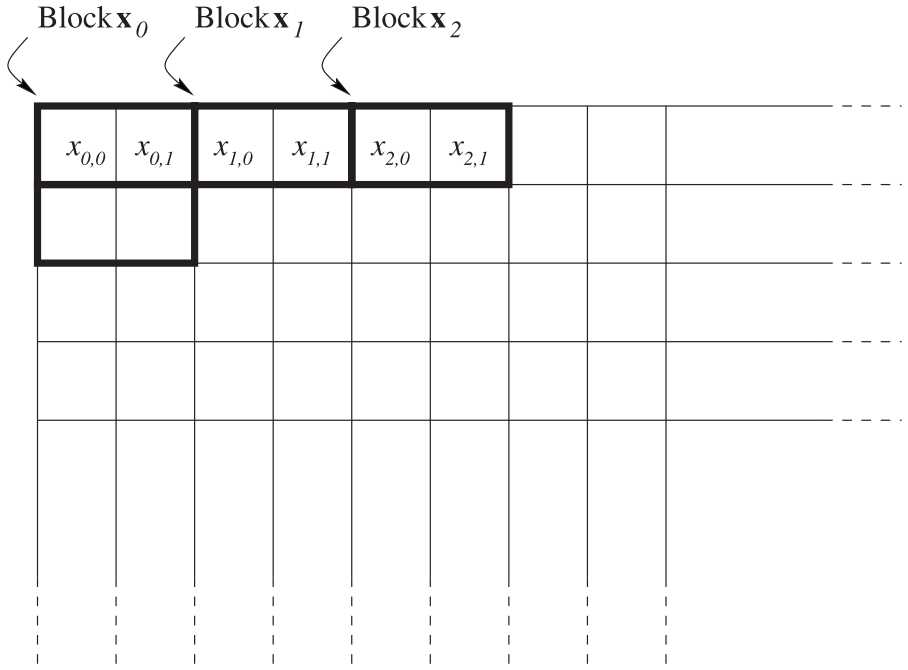

FIGURE 1.3
Image blocking with 1×2 pixel nonoverlapping blocks.

Further, the total energy under the transformation is preserved

$$\begin{aligned} \|\mathbf{y}\|^2 &= \mathbf{y}^T\mathbf{y} \\ &= \left([\mathbf{W}]^T\mathbf{x}\right)^T\left([\mathbf{W}]^T\mathbf{x}\right) \\ &= \mathbf{x}^T[\mathbf{W}][\mathbf{W}]^T\mathbf{x} \\ &= \mathbf{x}^T\mathbf{x} \\ &= \|\mathbf{x}\|^2 \ , \end{aligned} \tag{1.8}$$

where $\|\mathbf{x}\|$ is the norm of the vector $\mathbf{x}$ defined as

$$\begin{aligned} \|\mathbf{x}\| &= \sqrt{\mathbf{x}^T\mathbf{x}} \\ &= \sqrt{\sum_{i=1}^{N} x_i^2} \ . \end{aligned} \tag{1.9}$$

For the above example where $N = 2$, by inspection, the matrix $[\mathbf{W}]$ is simply a

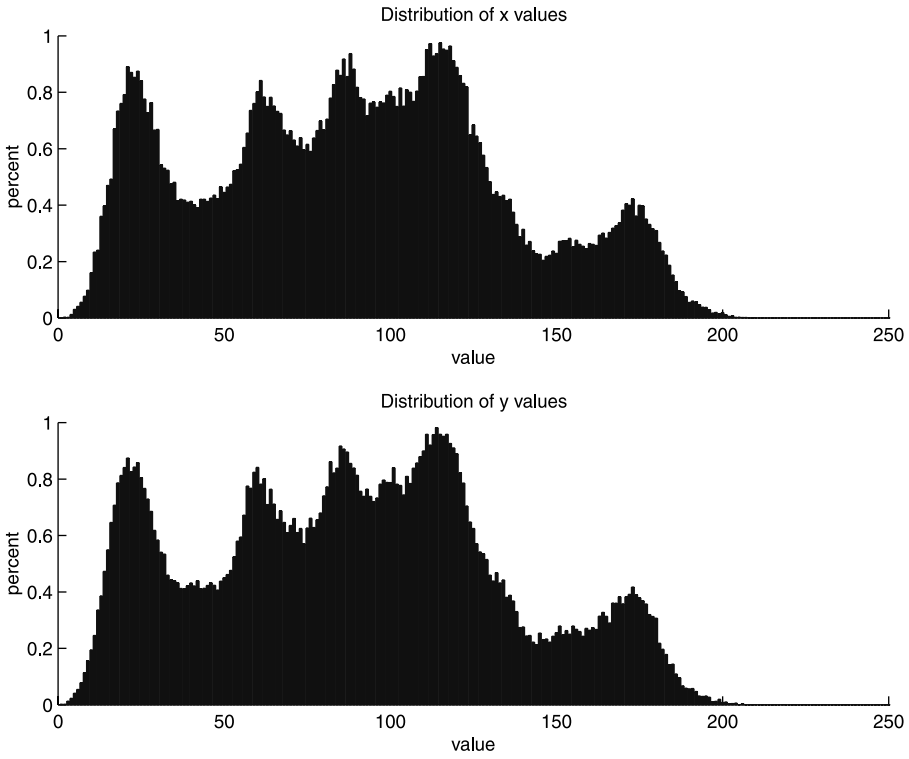

FIGURE 1.4
Distributions for each component.

rotation by 45°

$$[\mathbf{W}] = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix}. \tag{1.10}$$

For an arbitrary covariance matrix, the problem of finding the appropriate transformation is the orthonormal eigenvector problem. Since the covariance matrix is real and symmetric, we can find its real eigenvalues and corresponding eigenvectors. Let $[\mathbf{C}]_y$ be the desired diagonal covariance matrix of the transformed variable $\mathbf{y}$ which will be of the form

$$[\mathbf{C}]_y = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}, \tag{1.11}$$

where the diagonal elements are the variances of the transformed data. The diagonal

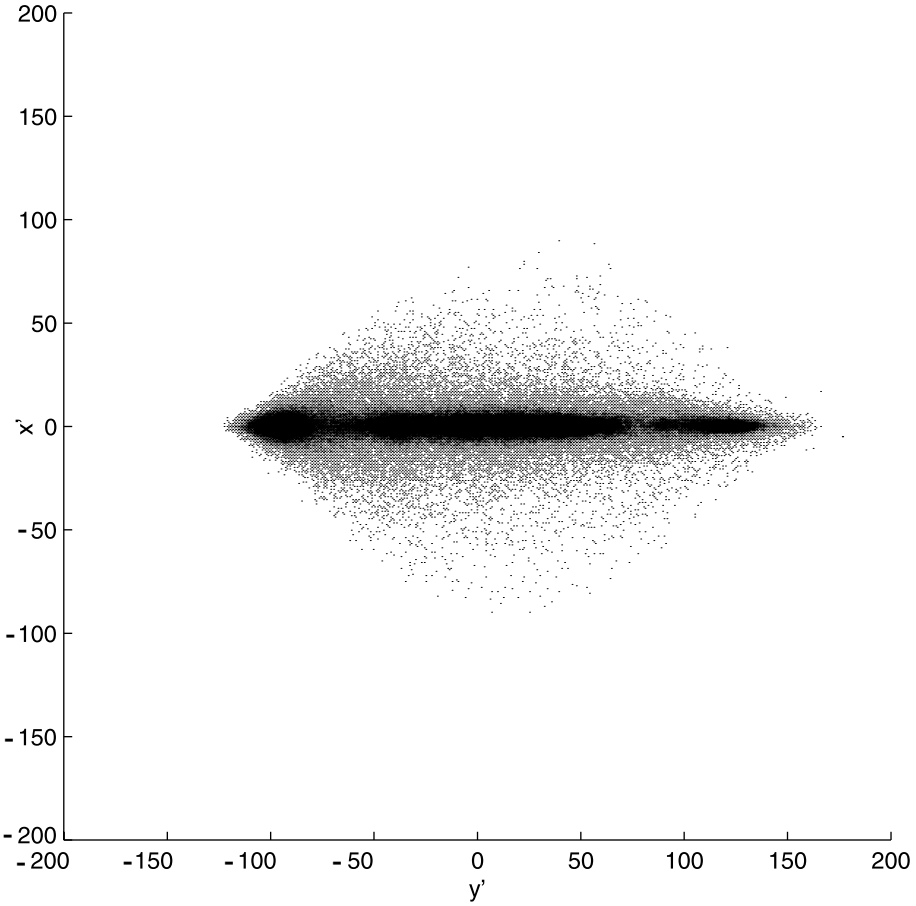

FIGURE 1.5
Scatter plot of pixel value pairs rotated by 45°.

matrix can be calculated from the original covariance matrix, $[\mathbf{C}]_x$, as

$$\begin{aligned}
[\mathbf{C}]_y &= E\left[\mathbf{y}\mathbf{y}^T\right] \\
&= E\left[\left([\mathbf{W}]^T\mathbf{x}\right)\left([\mathbf{W}]^T\mathbf{x}\right)^T\right] \\
&= E\left[[\mathbf{W}]^T\left(\mathbf{x}\mathbf{x}^T\right)[\mathbf{W}]\right] \\
&= [\mathbf{W}]^T[\mathbf{C}]_x[\mathbf{W}]\ ,
\end{aligned} \tag{1.12}$$

or equivalently,

$$[\mathbf{C}]_x[\mathbf{W}] = [\mathbf{W}][\mathbf{C}]_y\ . \tag{1.13}$$

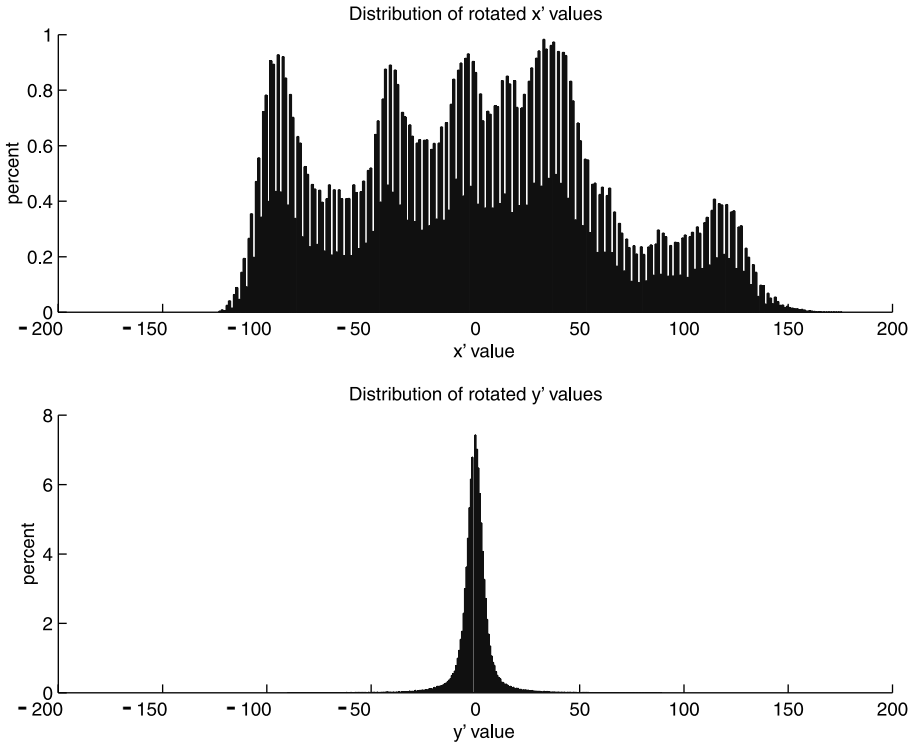

FIGURE 1.6
Distributions for each component of the rotated pixel value pairs.

Since the desired $[\mathbf{C}]_y$ is diagonal, Eq. (1.13) can be rewritten for each column vector, $\mathbf{w}_i$, of $[\mathbf{W}]$ as

$$[\mathbf{C}]_x \mathbf{w}_i = \lambda_i \mathbf{w}_i \ . \tag{1.14}$$

The solutions for λ_i and $\mathbf{w}_i$ with $i = 1, \ldots, N$ in Eq. (1.14) are the N eigenvalue, eigenvector pairs of the matrix $[\mathbf{C}]_x$ of dimension $N \times N$. That is, each column vector of $[\mathbf{W}]$ is an eigenvector of the covariance matrix, $[\mathbf{C}]_x$, of the original data. To ensure that $[\mathbf{W}]$ is orthonormal, Gram-Schmidt orthogonalization may be applied to the eigenvectors as they are obtained.

This transformation defined by the eigenvalues of the covariance matrix is the Karhunen-Loève transform (KLT), named after Karhunen [17] and Loève [19] who developed the continuous version of the transformation for decorrelating signals. Earlier, Hotelling [15] had developed a "method of principal components" for removing the correlation from the discrete elements of a random variable. As a result, the method is also referred to as the Hotelling transform or principal components analysis (PCA).

1.2.1 Calculation of the KLT

Estimation of Covariance

The calculation of the KLT is typically performed by finding the eigenvectors of the covariance matrix, which, of course, requires an estimate of the covariance matrix. If the entire signal is available, as is the case for coding a single image, the covariance matrix can be estimated from n data samples as

$$[\widehat{\mathbf{C}}]_x = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \ , \tag{1.15}$$

where $\mathbf{x}_i$ is a sample data vector. If only portions of the signal are available, care must be taken to ensure that the estimate is representative of the entire signal. In the extreme, if only one data vector is used then only one nonzero eigenvalue exists, and its eigenvector is simply the scaled version of the data vector. For typical images, it is rarely the case that their covariance matrix has any zero eigenvalues. For a data vector of dimension N, a good rule of thumb is that at least $10 \times N$ representative samples from the various regions within an image be used to ensure a good estimate if it is not feasible to use the entire image.

Calculation of Eigenvectors

While it is beyond the scope of this chapter to provide a detailed discussion of the algorithms for extracting the eigenvalues and eigenvectors, we will present a brief overview of the general methods commonly used. The reader is referred to [16, 28] for more detailed explanations. For actual implementations of the methods, many numerical packages such as LAPACK [22] (which is based on EISPACK [21] and LINPACK [23]), MATLAB [20], IDL [31], and Octave [11], and the routines in "cookbooks," such as that by Press et al. [28], provide routines for the solution of eigensystems.

A simple approach is the Jacobi method. It develops a sequence of rotation matrices, $[\mathbf{P}]_i$, that diagonalizes $[\mathbf{C}]$ as

$$[\mathbf{D}] = [\mathbf{V}]^T [\mathbf{C}][\mathbf{V}] \ , \tag{1.16}$$

where $[\mathbf{D}]$ is the desired diagonal matrix and $[\mathbf{V}] = [\mathbf{P}]_1[\mathbf{P}]_2[\mathbf{P}]_3 \cdots$. Each $[\mathbf{P}]_i$ rotates in one plane to remove one of the off-diagonal elements. It is an iterative technique which is terminated when the off-diagonal values are close to zero within some tolerance. Upon termination, the matrix $[\mathbf{D}]$ contains the eigenvalues on the diagonals and the columns of $[\mathbf{V}]$ are the basis vectors of the KLT.

While this technique is quite simple, for larger matrices it can take a large number of calculations for convergence. A more efficient approach for larger, symmetric matrices divides the problem into two stages. The Householder algorithm can be applied to reduce a symmetric matrix into a tridiagonal form in a finite number of steps. Once the matrix is in this simpler form, an iterative method such as QL factorization can be used to generate the eigenvalues and eigenvectors. The advantage

of this approach is that the factorization on the simplified tridiagonal matrix typically requires fewer iterations than the Jacobi method.

Recently, there has been some interest in iterative methods of principal components extraction that do not require the calculation of a covariance matrix [7, 14, 26]. These techniques update the estimate of the eigenvectors for each input training vector. One such method developed by Oja [25] is of the form

$$\hat{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) + \alpha \left[y(t)\mathbf{x}(t) - y^2(t)\hat{\mathbf{w}}(t) \right] , \tag{1.17}$$

where $\mathbf{x}$ is an input vector, $\hat{\mathbf{w}}(t)$ is the current estimate of the basis vector, $y = \mathbf{w}^T\mathbf{x}$ is the coefficient value, and α is a learning-rate parameter. Eq. (1.17) has been shown to converge to the largest principal component [14, 27]. This algorithm can be generalized through deflation to extract any or all of the principal components [7, 33]. Also, adaptive schemes have been based on this method [8]. While these algorithms have some advantages over covariance-based methods, there are still some concerns over stability and convergence [3, 4, 35].

Markov-1 Solution

The calculation of the eigenvectors for an arbitrary covariance matrix can still require a large number of computations. However, there is a special class of matrix that has an analytical solution for its eigenvectors and eigenvalues [29, 30]. If a process were to have a covariance function of the form

$$[\mathbf{C}]_{ij} = \sigma^2 \rho^{|i-j|} , \tag{1.18}$$

where ρ is the correlation coefficient such that $0 < \rho < 1$, such a process is referred to as a first order stationary Markov process or simply Markov-1. The solution for the ith element of the jth basis vector for N-dimensional data is given by

$$w_{ij} = \left[\frac{2}{\left(N + \mu_j\right)} \right]^{1/2} \sin \left\{ r_j \left[(i+1) - \frac{(N+1)}{2} \right] + (j+1)\frac{\pi}{2} \right\} , \tag{1.19}$$

where μ_j is the jth eigenvalue calculated as

$$\mu_j = \left(1 - \rho^2\right) \left[1 - 2\cos\left(r_j\right) + \rho^2 \right] , \tag{1.20}$$

and r_j is the jth real positive root of the transcendental equation

$$\tan\left(Nr\right) = -\frac{\left(1-\rho^2\right)\sin\left(r\right)}{\cos\left(r\right) - 2\rho + \rho^2\cos\left(r\right)} . \tag{1.21}$$

To extend this to two-dimensional data, one can assume a separable transform. The horizontal and vertical correlation coefficients, ρ_H and ρ_V, are estimated from the image to calculate a horizontal basis set, $w_{ij}^{(H)}$, and vertical basis set, $w_{ij}^{(V)}$, respectively.

Then, the i, j element of the kth two-dimensional basis vector, w_{ijk}, is calculated as the product of the two:

$$w_{ijk} = w_{ik}^{(H)} w_{jk}^{(V)} \ . \tag{1.22}$$

As many images exhibit a Markov-1 structure, this solution to the KLT can be quite useful due to its ease of generation.

1.3 Performance of Transforms

On its own, an orthonormal transformation does not effect data compression. The blocks of pixels are simply transformed from one set of values to another and, for reversible transformations, back again on reconstruction. To reduce the number of bits for representing an image, the coefficients are quantized, incurring some irreversible loss, and then encoded for more efficient representation. By decorrelating the data before these steps using the KLT, more data compaction can be achieved.

To examine the effects of this extra efficiency, we can make use of Shannon's information measures [34].

1.3.1 Information Theory

The information conveyed by an observation of some random process is related to its probability of occurrence. If an observation were all but certain to occur, i.e., its probability were close to 1, it would not be very informative. However, if it were quite unexpected, the observation would convey much more information. Shannon formalized this relationship between the probability of an event, $P(x)$, and its information content, $I(x)$, as

$$I(x) = -\log P(x) \ . \tag{1.23}$$

If the logarithm is taken with respect to base 2, the information, $I(x)$, is measured in units of *bits*.

A random variable, $\mathbf{x}$, is a collection of all possible events and their associated probabilities. The average information for a random variable can be calculated as

$$\begin{aligned} H(\mathbf{x}) &= \sum_i P\,(x_i)\, I\,(x_i) \\ &= -\sum_i P\,(x_i) \log P\,(x_i) \ , \end{aligned} \tag{1.24}$$

where the sum is taken through all possible events. The average information is called the entropy of the process.

Entropy is useful in determining theoretical performance measures of compression methods. Shannon showed that entropy gives a lower bound on the average number of bits required to encode the events of a random process without introducing error. In other words, one needs at least as many bits per event, on average, as the entropy to represent a set of observations.

However, these measures are not directly applicable to the coefficients of an arbitrary transformation. They are defined for discrete events whereas the coefficients, since they are floating-point values, must be considered real-valued samples of continuous distributions. Since the probability of any such real-valued sample is zero, the (discrete) entropy is undefined. Instead, we define the *differential entropy* [13] as

$$h(x) = -\int_{-\infty}^{\infty} p(s) \log p(s) ds \ . \tag{1.25}$$

For simple distributions such as the Gaussian, uniform, or Laplacian distributions the differential entropy is of the form

$$h(x) = \frac{1}{2} \log \sigma_x^2 + k \ , \tag{1.26}$$

where σ_x^2 is the variance of the random variable and k is a distribution-dependent constant (e.g., for a Gaussian, $k = \frac{1}{2} \log_2 2\pi e$) [1].

A good transformation, then, should minimize the sum of the differential entropies for the resulting coefficients. Due to the logarithmic term, this is equivalent to minimizing the product of the variances of the coefficients. However, recall that for any orthonormal transformation, the total energy is preserved, so the sum of the coefficient variances is fixed. One measure of the efficiency of the transform is the coding gain [10] defined as the ratio between the algebraic mean of the variances, which is independent of the transform, and the geometric mean of the variances, which is transform dependent:

$$G_W = \frac{\frac{1}{N} \sum_{i=1}^{N} \sigma_{y_i}^2}{\left(\prod_{i=1}^{N} \sigma_{y_i}^2 \right)^{1/N}} \ . \tag{1.27}$$

For the raw signal, before any transformation, all the variances are approximately equal giving a unity coding gain. Any increase in one of the coefficient variances must be matched by an equal decrease in one or more of the other variances for an orthonormal transform. The arithmetic mean is therefore the same, but the geometric mean decreases resulting in a coding gain of greater than one.

For a given energy of the signal, minimizing the product of the variances maximizes the coding gain. Conversely, maximizing the coding gain minimizes the lower bound on the number of bits required to encode the image. So, to minimize the product of the variances given a fixed sum, one should maximize the variance of the first

coefficient. Next, subject to the orthonormality constraint, maximize the variance of the second coefficient, and so on. This procedure is nothing more than extracting the principal components or, equivalently, generating the KLT. Therefore, the KLT, by decorrelating the data, produces a set of coefficients that minimizes the differential entropy of the data.

1.3.2 Quantization

In transform coding, the transform coefficients are quantized to effect the data reduction. While the transformation is reversible, quantization is not, and therefore introduces error. Let $\hat{\mathbf{y}}$ be the set of quantized coefficient values for a block. On reconstruction, the block is calculated as

$$\hat{\mathbf{x}} = [\mathbf{W}]\hat{\mathbf{y}} \; . \tag{1.28}$$

The squared error for the block is calculated as

$$\begin{aligned} \varepsilon^2 &= \left\| \hat{\mathbf{x}} - \mathbf{x} \right\|^2 \\ &= \left(\hat{\mathbf{x}} - \mathbf{x} \right)^T \left(\hat{\mathbf{x}} - \mathbf{x} \right) \\ &= \left([\mathbf{W}]\hat{\mathbf{y}} - [\mathbf{W}]\mathbf{y} \right)^T \left([\mathbf{W}]\hat{\mathbf{y}} - [\mathbf{W}]\mathbf{y} \right) \\ &= \left(\hat{\mathbf{y}} - \mathbf{y} \right)^T [\mathbf{W}]^T [\mathbf{W}] \left(\hat{\mathbf{y}} - \mathbf{y} \right) \\ &= \left(\hat{\mathbf{y}} - \mathbf{y} \right)^T \left(\hat{\mathbf{y}} - \mathbf{y} \right) \\ &= \left\| \hat{\mathbf{y}} - \mathbf{y} \right\|^2 \; . \end{aligned} \tag{1.29}$$

So, the squared error on reconstruction is the same as the squared error of the coefficients for orthonormal transformations.

The quantized coefficients are typically encoded using a lossless method, such as arithmetic coding or Huffman coding. These methods can, at best, reduce the average number of bits to the entropy of the quantized coefficients.

To illustrate the advantage of performing the KLT before quantization, we calculate the total entropy for a number of quantization intervals on both the original data and the transformed data. For this example, a midstep, uniform quantizer is used where the quantized value is calculated as

$$\hat{y} = q \text{ round } (y/q) \; , \tag{1.30}$$

based on the width of the quantization interval, q, where the function round(x) returns the nearest integer to the real value x. The results are shown in Fig. 1.7. For a given squared error due to quantization, the entropy in bits per pixel is less for the transformed data than for the original data.

1.3.3 Truncation Error

Another approach to reducing the data and hence introducing error is the complete removal of a number of the coefficients before quantization. Say only M of the N coefficients were to be retained. The resulting expected squared error is calculated as

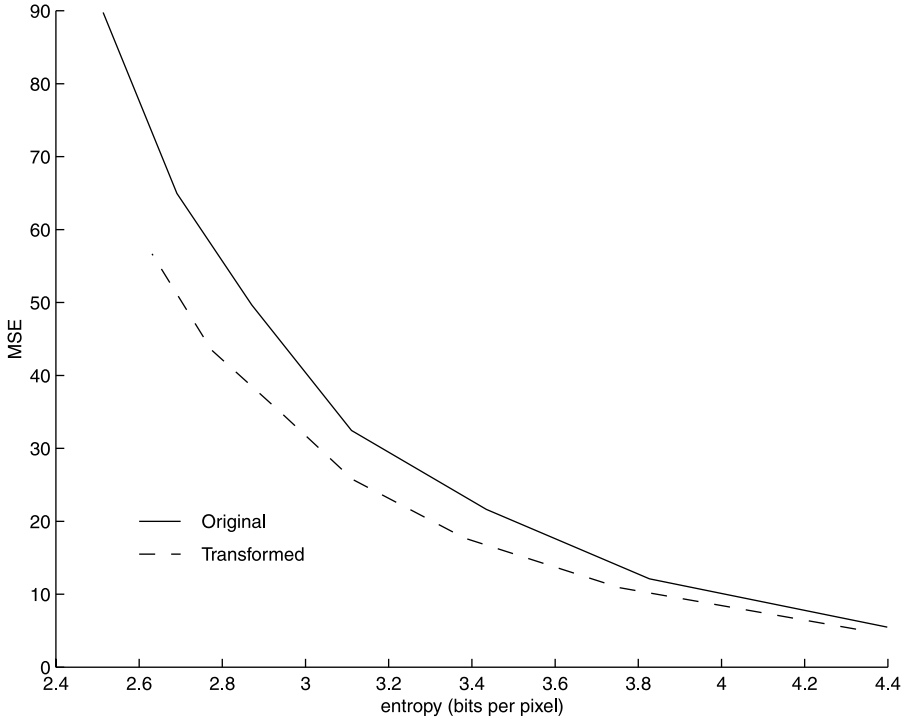

FIGURE 1.7
Plot of mean squared error (MSE) versus entropy in bits per pixel for a number of quantization widths.

$$
\begin{aligned}
E\left[\varepsilon^2\right] &= E\left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2\right] \\
&= \frac{1}{N}E\left[\sum_{i=1}^{M}(y_i-y_i)^2+\sum_{i=M+1}^{N}(y_i-0)^2\right] \qquad (1.31)\\
&= \frac{1}{N}E\left[\sum_{i=M+1}^{N}y_i^2\right] \\
&= \frac{1}{N}\sum_{i=M+1}^{N}\sigma_i^2\ .
\end{aligned}
$$

Recall that for the KLT the variances of the coefficients, σ_i^2, are the eigenvalues, λ_i, of the covariance matrix. To minimize the expected squared error, the M coefficients corresponding to the M largest eigenvalues should be kept.

Notice that the above minimization is valid for any transformation whose M basis vectors span the M-dimensional subspace defined by the M largest principal components (eigenvectors for the M largest eigenvalues). However, only the KLT ensures that the remaining coefficients can be coded with the minimum number of bits since it minimizes the differential entropy of the coefficients. To illustrate this point, let us generate the 64 KLT basis vectors for an 8×8 blocking of the test image and keep only the first four. The variances of the resulting coefficients are shown in the first column of Table 1.1. The MSE due to the removal of the 60 lowest variance coefficients is 96.1. Now, let us generate another set of 4 basis vectors by taking random linear combinations of the first 4 KLT basis vectors. The new set still spans the space defined by the original 4 KLT basis vectors. As a result, the MSE due to truncation and the sum of the remaining variances are identical to those of the KLT bases. However, the product of the variances is much higher, and, as a result, the coding gain is much smaller than for the KLT bases. This means that the representation is less efficient and will require more bits to encode the coefficients for the same degree of distortion.

Table 1.1 Performance Differences Between First Four Basis Vectors of KLT and a Random Combination of Them

	KLT bases	Random span
σ_1^2	113995	20876
σ_2^2	6880	18236
σ_3^2	2727	79310
σ_4^2	1691	6873
$\sum_{i=1}^{4} \sigma_i^2$	125294	125294
$\sum_{i=5}^{64} \sigma_i^2$	6147	6147
Truncation MSE	96.1	96.1
$\prod_{i=1}^{4} \sigma_i^2$	3.6×10^{15}	207.5×10^{15}
Coding gain	4.04	1.47

1.3.4 Block Size

The question remains of what size to use for the image blocks. The larger the block, the greater the decorrelation, hence the greater the coding gain. However, the number

of arithmetic operations for the forward and inverse transformations increases linearly with the number of pixels in the block. Furthermore, the size of the covariance matrix is the square of the number of pixels. Not only does the calculation of the eigenvectors require more resources, but the number of samples to get a reasonable estimate of the covariance matrix increases significantly. As well, if the set of KLT basis vectors is to be kept with the image for reconstruction, the size of the basis set is also of concern. Therefore, there is a trade-off between computational requirements and the degree of decorrelation in determining the block size.

Fig. 1.8 shows the coding gain as a function of block size for the test image. It clearly shows that the use of larger block sizes results in larger coding gains. For example, increasing the block size from 4×4 to 8×8 increases the gain from 27 to 39. However, the number of floating point operations per pixel increases by a factor of four from 32 to 128.

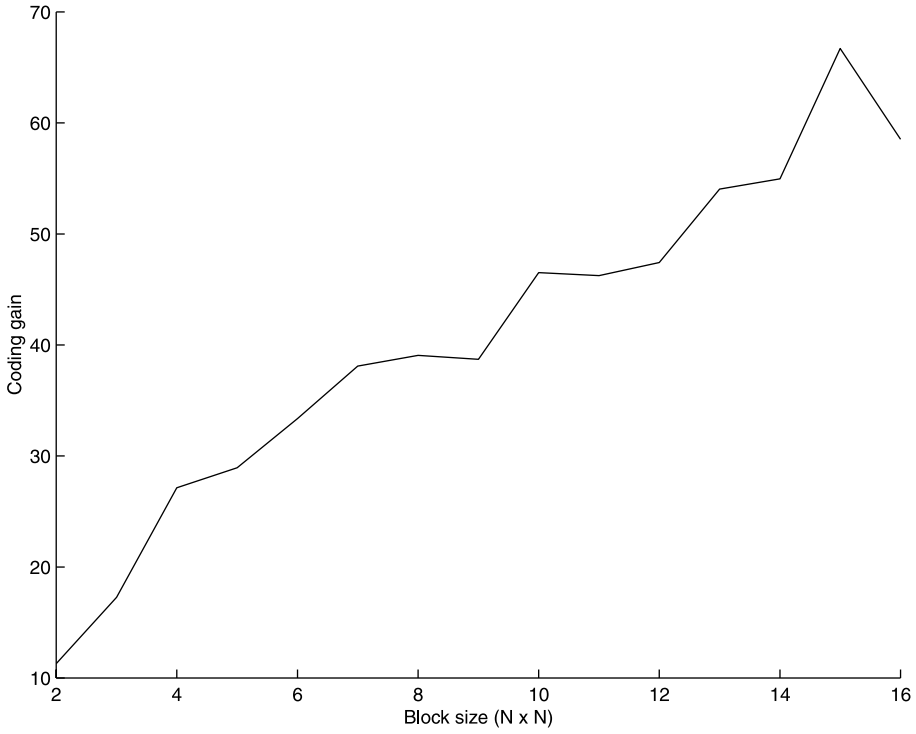

FIGURE 1.8
Coding gain as a function of block size for test image.

Of course, using a block the same size as the image results in a perfect coding gain since the entire image can be represented by a single component. Unfortunately, this representation is so image specific that the transform basis itself must also be included with the compressed image to enable reconstruction. Since the basis vector *is* the image, one is no further ahead. However, such full-frame transform coding may be appropriate for sequences or collections of similar images.

Interblock Correlation

The KLT produces decorrelated coefficients within the image blocks. There is no assurance, however, that the coefficients from block-to-block are also decorrelated. In fact, for most images there is a significant correlation between the first coefficients for adjacent blocks. For example, Fig. 1.9 shows the scatter plot of adjacent pairs of the first coefficient for the 8×8 KLT of the test image. Note the strong correlation between the adjacent values. In contrast, Fig. 1.10 shows little if any correlation between adjacent second coefficients.

A simple method of reducing such correlation is to encode only the difference between adjacent coefficients after initially encoding the first. This method is known as differential pulse code modulation (DPCM). The use of DPCM on the first coefficients significantly increases the overall coding efficiency by reducing the variance of the coefficient. For example, performing DPCM on the first coefficient of the above 8×8 KLT coefficients reduces the variance from 113995 to 51676. The resulting scatter plot of the adjacent pairs of differences is shown in Fig. 1.11. The use of DPCM has removed the correlation between adjacent values of the first coefficient.

1.4 Examples

1.4.1 Calculation of KLT

To calculate the KLT of an image, the covariance matrix is first estimated. The estimate is calculated from the set of sequential nonoverlapping blocks for the image. For the following examples, blocks of 8×8 pixels are used. For the "Lena" image, this results in 4096 blocks. The eigenvalues and the corresponding eigenvectors are extracted from the covariance matrix. Because the matrix is symmetric, the eigenvalues and eigenvectors can be calculated using the tridiagonalization and QL factorization approach.

The resulting 64 basis vectors are shown in Fig. 1.12 as two-dimensional basis images or blocks. The bases are in order from the largest variance at the top left to the lowest at the bottom right. Dark pixels represent negative values and light pixels represent positive values. The first basis is almost flat due to the similarity of pixel values within most blocks. As was the case for the two-dimensional scatter plot of Fig. 1.2, the 64-dimensional scatter plot would show a strong concentration of points along the diagonal line $x_1 = x_2 = \cdots = x_{64}$. As this is true for most images, the

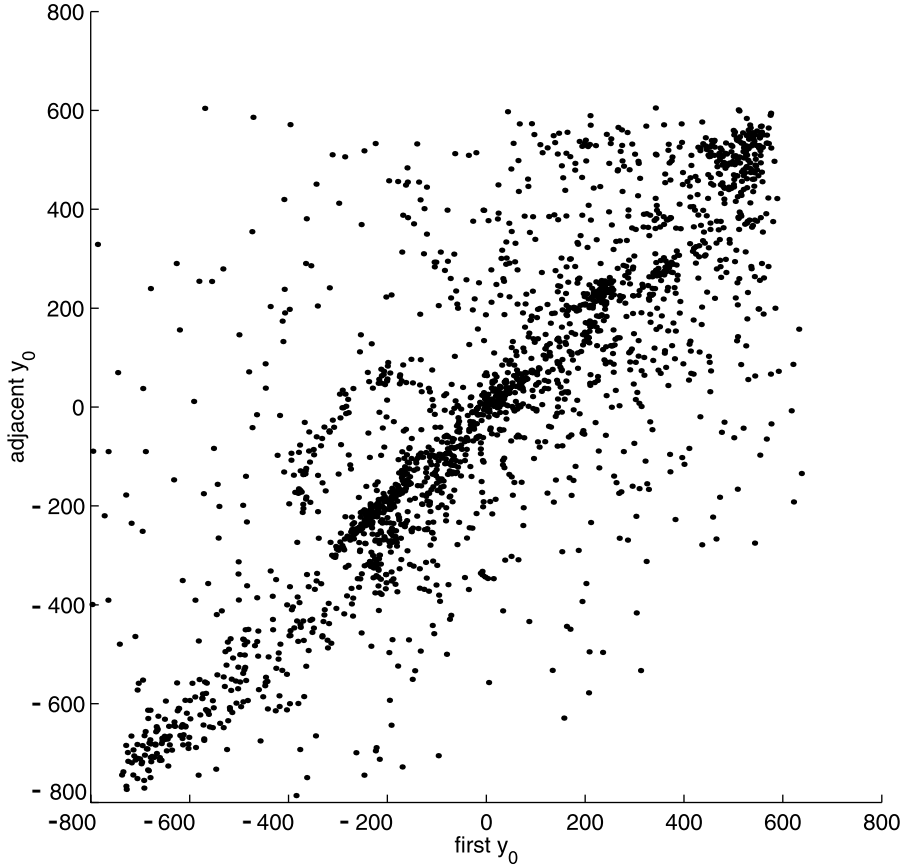

FIGURE 1.9
Scatter plot of adjacent pairs of the first coefficient.

first component of the KLT tends to be constant or d.c. As the variance increases, the degree of variation, or frequency, increases. This relationship generally agrees with the form of the KLT solution for a Markov-1 process as shown in Eq. (1.19) where the frequency increases as the basis index increases. Again, as most images have an approximate Markov-1 structure, the form of the KLT bases are similar.

1.4.2 Quantization and Encoding

Once the coefficients are calculated, they are quantized and then losslessly encoded. There are numerous such methods, but a discussion and comparison of them would be beyond the scope of this chapter. For illustrative purposes, we will use an encoding scheme similar to that adopted by the JPEG standard [36]. The coefficients are quantized by a midstep uniform quantizer as defined in Eq. (1.30). For simplicity, the

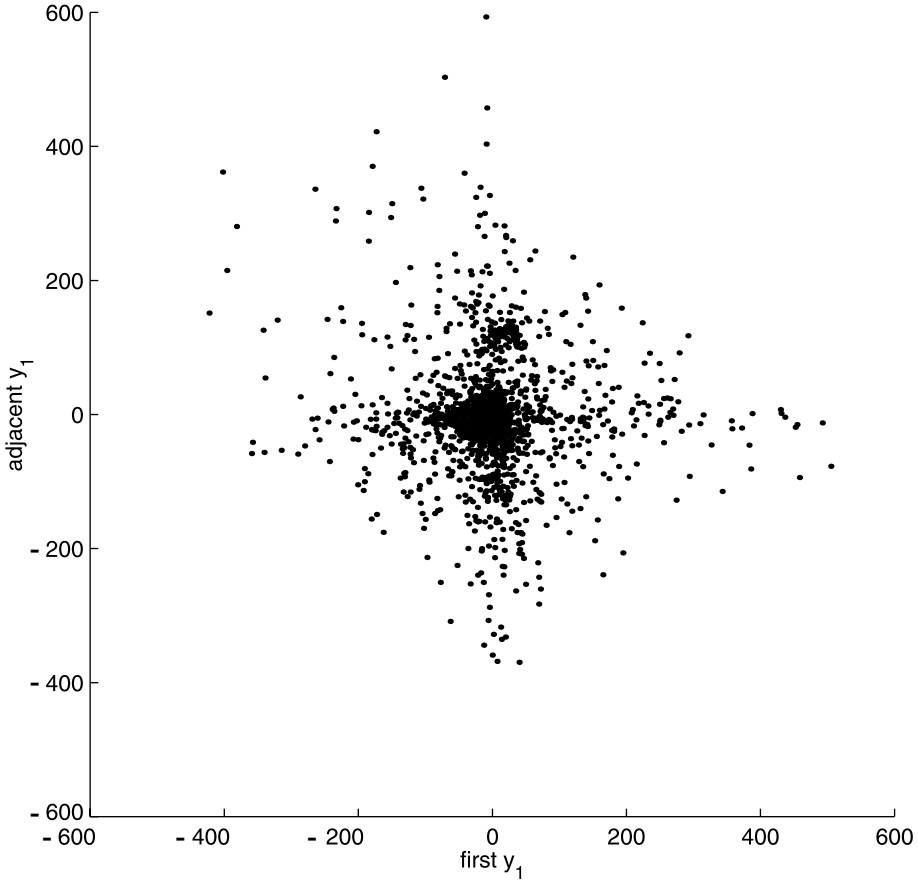

FIGURE 1.10
Scatter plot of adjacent pairs of the second coefficient.

same quantization step size, q, is used for all coefficients, unlike the JPEG standard that varies the degree of quantization for each coefficient according to the visibility of error as judged by human observers. Each quantized coefficient is encoded first by a Huffman encoded value for the number of bits required by the coefficient followed by the minimum number of bits for the coefficient value itself. Zero-valued coefficients from adjacent blocks are run-length encoded for further compaction.

The results for various degrees of quantization are shown in Table 1.2. As the coarseness of quantization increases, the size of the file decreases resulting in greater compression. The equivalent average number of bits per pixel is also shown. For comparison to show the efficiency of the coefficient encoding, the entropy of the quantized coefficient values is also shown. The actual bit rate and the entropy are very similar. At high compression the actual bit rate is slightly lower than the entropy because of the run-length encoding of zero values.

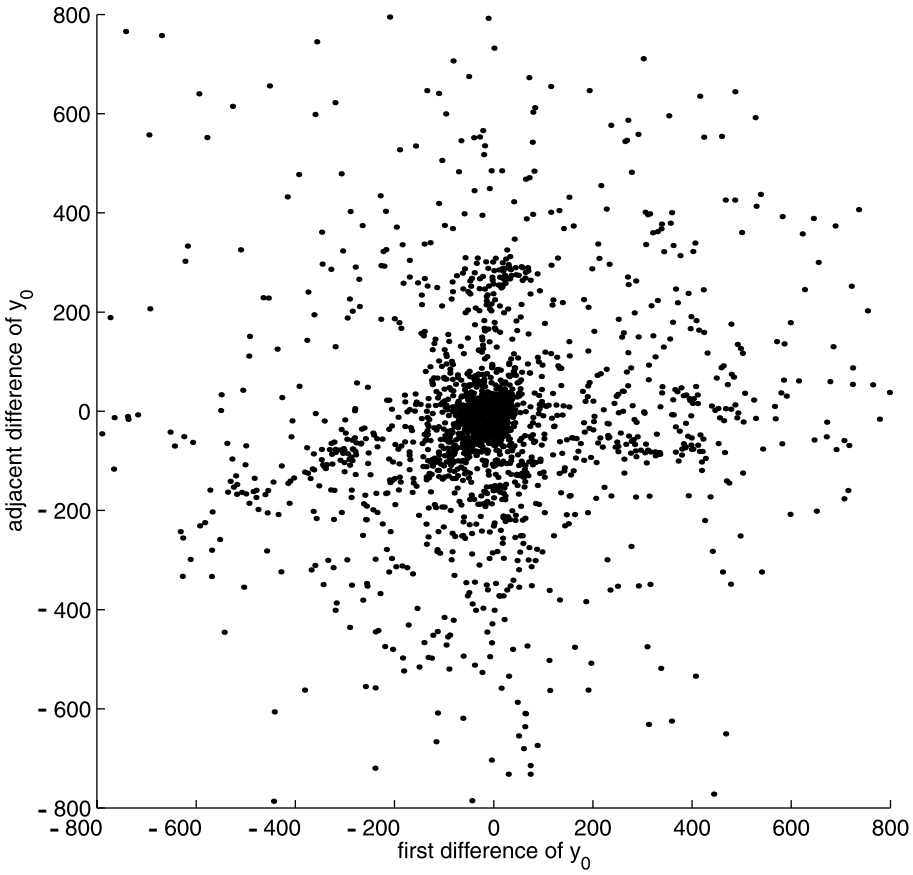

FIGURE 1.11
Scatter plot of adjacent pairs of differences of the first coefficient.

As the bit-rate decreases, distortion increases. Table 1.2 shows the distortion in two equivalent common measures [6]. The mean squared error (MSE) is defined as

$$\text{MSE} = E\left[\left(x - \hat{x}\right)^2\right], \tag{1.32}$$

where x is the original pixel value and $\hat{x}$ is the reconstructed value. The peak signal-to-noise ratio (PSNR) is a logarithmic measure of distortion given in decibels (dB) and is defined as

$$\text{PSNR} = 10\log_{10}\frac{(255)^2}{E\left[\left(x - \hat{x}\right)^2\right]}, \tag{1.33}$$

where 255 is the peak value of an 8-bit image. The larger the PSNR value, the better the accuracy of reconstruction. The plot of the distortion as PSNR versus the bit

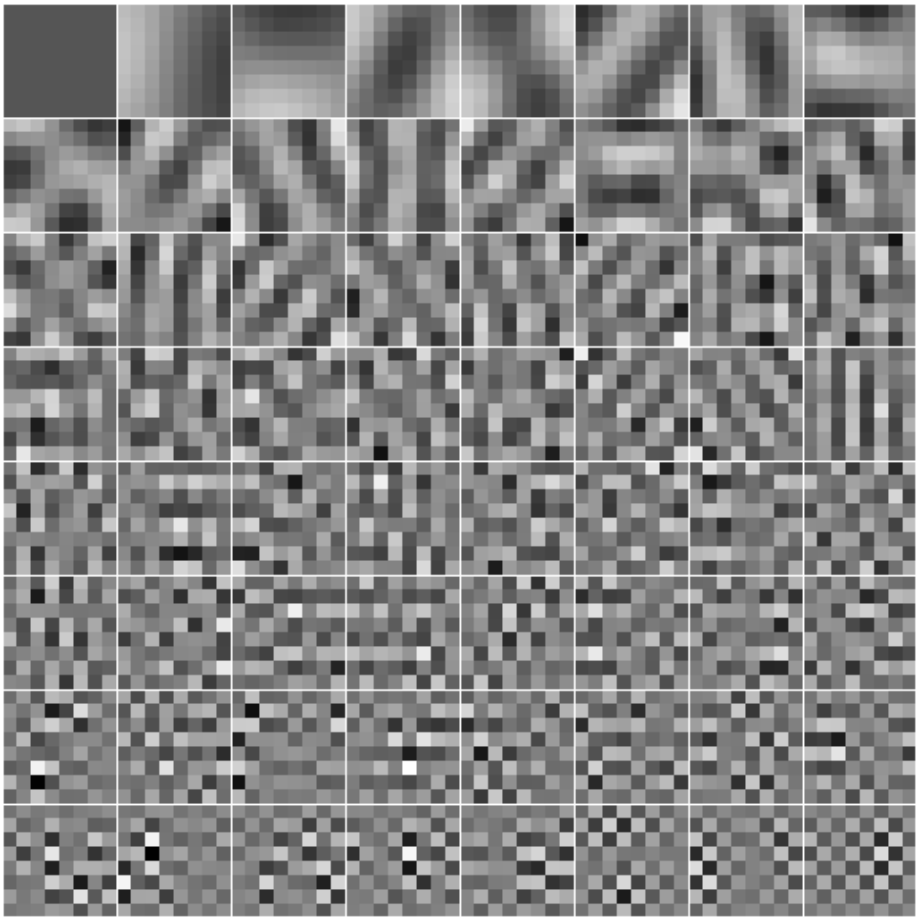

FIGURE 1.12
KLT basis images for "Lena" image.

rate is shown in Fig. 1.13. From rate-distortion theory, for a stationary memoryless Gaussian source, the bit rate, R, as a function of the squared error distortion, ε^2, is given by [1]

$$R(\varepsilon) = \begin{cases} \frac{1}{2}\log_2\left(\sigma^2/\varepsilon^2\right) & 0 \le \varepsilon^2 < \sigma^2\,, \\ 0 & \sigma^2 \le \varepsilon^2\,. \end{cases} \tag{1.34}$$

For high bit rates, the rate-distortion curve follows the logarithmic relationship between the squared error and the bit rate. As the quantization interval increases, the distortion overtakes the variance for more coefficients. As a result, the curve begins to drop sharply as the distortion increases without a corresponding further reduction in bit rate. In the limit as the quantization interval increases, the bit rate becomes zero

Table 1.2 Compression of "Lena" Image Using KLT

Quantizer Width	File Size (bytes)	Bits/pixel	Entropy (bits)	MSE	PSNR (dB)
2	139948	4.27	4.08	0.42	51.95
4	109141	3.33	3.11	1.42	46.62
8	78820	2.41	2.18	5.19	40.98
16	42245	1.29	1.28	15.01	36.37
24	27196	0.83	0.90	23.78	34.37
36	18375	0.56	0.64	36.27	32.54
48	13893	0.42	0.50	48.45	31.28
64	10548	0.32	0.39	64.70	30.02
92	7547	0.23	0.28	93.68	28.41
128	5492	0.17	0.21	130.19	26.98
192	3797	0.12	0.15	199.21	25.14
256	2831	0.09	0.11	273.42	23.76
512	1457	0.04	0.06	638.18	20.08

and the squared error is then simply the variance.

Fig. 1.14 shows the reconstructed image after a compression of 10:1 (0.8 bits per pixel). Overall, very little distortion is visible. Areas of constant brightness, edges, lines, and textured regions are all reproduced quite faithfully. Even on closer examination, little distortion is evident, as shown by comparing Figs. 1.15(a) and (b). At 10:1 compression, some minor distortion is seen as spurious texture in the background. As well, the lone feather piece in the center-left region is somewhat distorted. As the compression ratio increases, though, the distortion becomes more apparent, as shown by Figs. 1.15(c) and (d) for ratios of 20:1 and 40:1, respectively. The texture of the hat is lost in areas at 20:1, while artifacts in the background region are more pronounced. The edges of the hat, however, are still rather crisp and the textured region of the feathers on the brim does not seem as distorted as the hat texture. Because the set of bases is image specific, certain features, such as these, may be well represented and be somewhat resistant to distortion at moderate compression ratios. By 40:1, though, the image is quite distorted. This type of distortion is sometimes referred to as "block effect distortion" because the block boundaries used in block transform coding are visible.

1.4.3 Generalization

In theory, the transform basis set for the KLT is specific to a particular image. However, in practice the statistics of images at the block-size level of detail tend to be similar. As a result, the KLT computed from one set of image data performs quite well on another set. For example, the above results were based on the KLT computed from the covariance matrix of the set of sequential, nonoverlapping blocks from the image. These blocks are the exact data that are used to encode the image. If the covariance

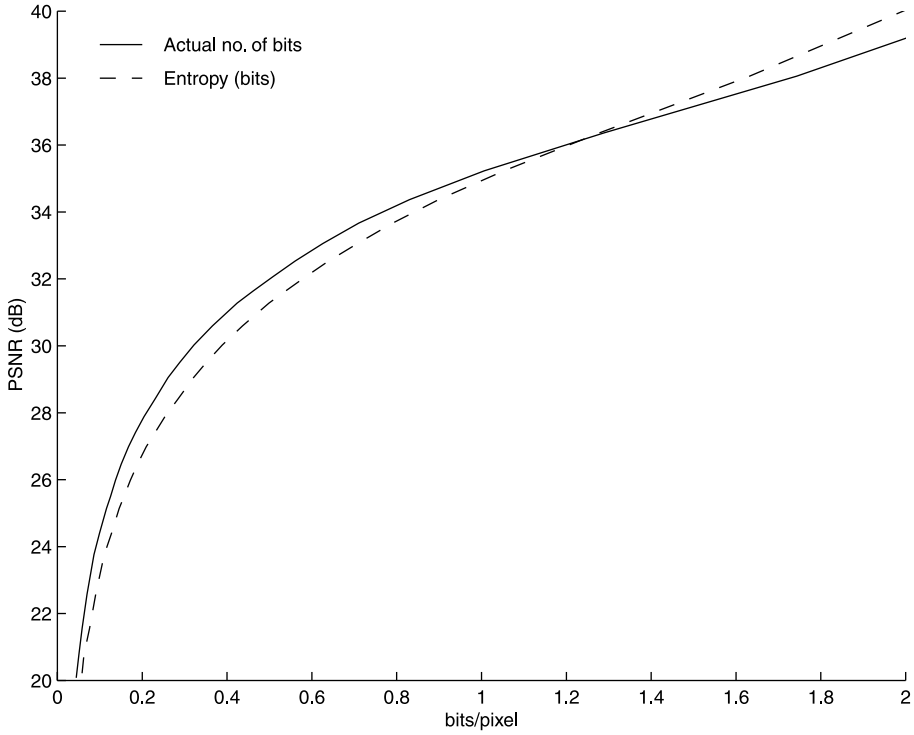

FIGURE 1.13
Plot of distortion (PSNR) versus bit rate showing both the entropy and actual coding rates.

matrix were to be calculated from randomly chosen blocks from arbitrary locations on the image, the data for generating the KLT would be different from the data used in encoding the image. Fig. 1.16 shows the results for both the KLT generated from the sequential set of blocks and a set of 4096 randomly chosen blocks. While the transform generated from the same data to be coded performs better, the improvement is not significant.

What happens if the KLT is generated based on an image completely different from the one being encoded? A second test image, "Goldhill," is shown in Fig. 1.17. This image was encoded using the KLT generated from the image and the KLT originally generated from the "Lena" image. The rate-distortion curves are shown for both cases in Fig. 1.18. As expected, using the same data for generating the transform as for encoding results in better performance than using different data to generate the transform. However, as the figure shows, this increase is only minor. In this case, the transformation based on the "Lena" image generalizes well to the other image.

FIGURE 1.14
Image after compression of 10:1, MSE = 24.8, PSNR = 34.2 dB. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

1.4.4 Markov-1 Solution

To compare the usefulness of the Markov-1 solution to the KLT, we first look at the autocorrelation of the image. As shown in Table 1.3, the autocorrelation does appear to follow the Markov-1 model of $\mathrm{E}[x_i x_j] = \mathrm{E}[x^2]\rho^{|i-j|}$ with $\rho_H = 0.9543$ for horizontally neighboring pixels. A similar relationship also holds for vertically neighboring pixels with $\rho_V = 0.9768$. For simplicity we will assume a separable, isotropic distribution and choose $\rho = 0.9543$ for both directions. The resulting KLT bases are shown in Fig. 1.19. Note the strong sinusoidal nature of the basis images. The rate-distortion results for using this set of KLT bases are shown in Fig. 1.20 along with the original results for the KLT generated from the image itself. Since the two

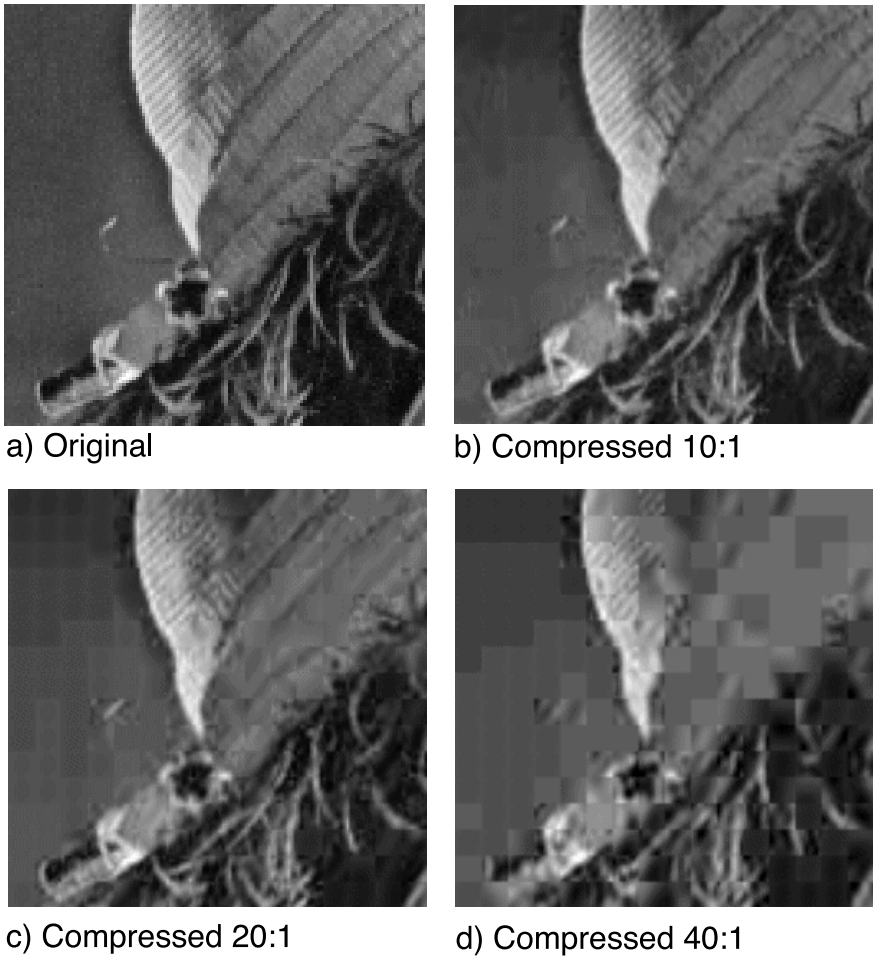

FIGURE 1.15
Details of image before and after 10:1, 20:1, and 40:1 compression. (a) Original, (b) Compressed 10:1, (c) Compressed 20:1, (d) Compressed 40:1. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

curves are almost identical, the savings in computational resources from having a closed form solution for the Markov-1 case incurs little if any cost in performance.

1.4.5 Medical Imaging

One of the most demanding application areas for the use of image compression is the compression of medical images. The implications of introducing any sort of distortion in this class of images are grave. There are numerous legal and regulatory issues which consequently are of concern [37]. As a result, there is an argument for

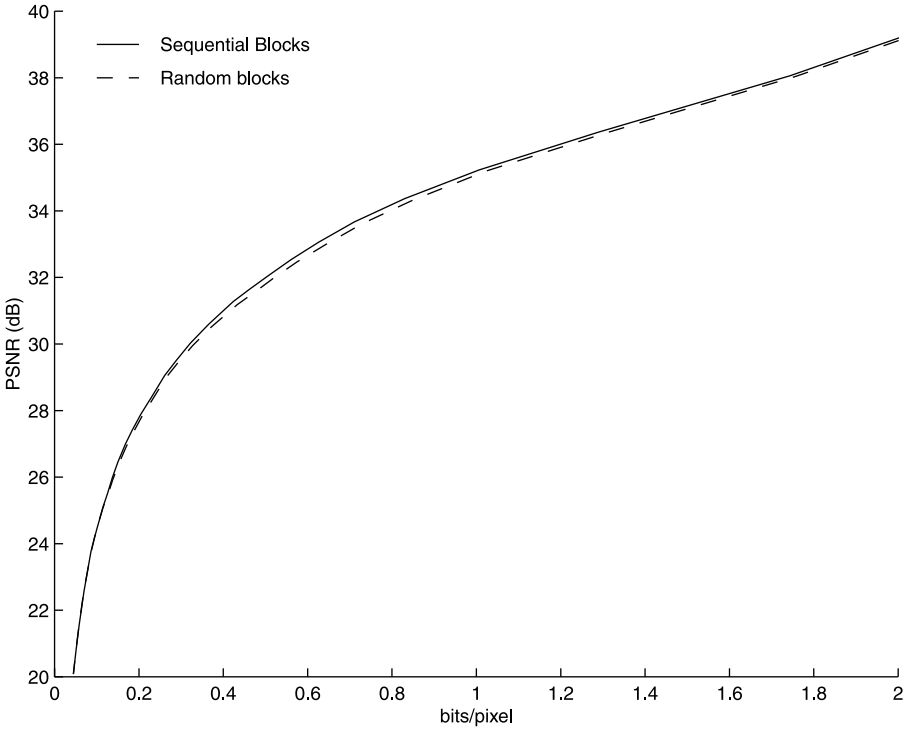

FIGURE 1.16
Plot of distortion versus bit rate for KLT calculated from both randomly chosen blocks and sequential blocks.

the use of lossless compression in this field; however, such an approach is of limited usefulness due to the theoretical limits on the maximum allowable compression.

The question, of course, is how much compression can be achieved? For lossy image compression methods, this is the same as asking how much distortion can be introduced in the reconstructed image. To answer this question, the end-use of the images must properly be defined. For the following example, as originally presented in Dony et al. [9], the application is for educational use. Currently, radiology residents acquire their diagnostic skills through examining actual clinical images of normal patients as well as those with various pathologies. With the growth in digital imaging, it is now possible to store such a library of images digitally in a computer database. The residents would be free to call up any of the images and examine them at their convenience. The evaluation criteria for this environment are quite different from, say, a diagnostic environment. In the educational environment, the diagnosis or pathology is given beforehand. It is sufficient that an image show clearly the pathology in question or the characteristics of a normal image. So, it is the overall quality of the image and the visibility of the pathology as judged by an experienced radiologist which must be measured.

FIGURE 1.17
Second test image, "Goldhill."

Nine digital chest radiographs (X-rays) obtained for clinical reasons were selected for evaluation as being representative of both normal anatomy and pathology. A sample image is shown in Fig. 1.21. Each of the nine images was compressed using an adaptive variation of the KLT at 10:1, 20:1, 30:1, and 40:1, and the five versions of each image were presented simultaneously to each of seven radiologists, in random order and without the evaluator knowing the degree of compression. The radiologists were asked to rank image quality and visibility of pathology in the context of their suitability for educational use. Possible ratings varied from excellent, good, and fair — acceptable — and poor or bad — unacceptable. A mean opinion score (MOS) was calculated by assigning a numeric value to each rating, e.g., excellent scored 5 points and bad 1 point [24].

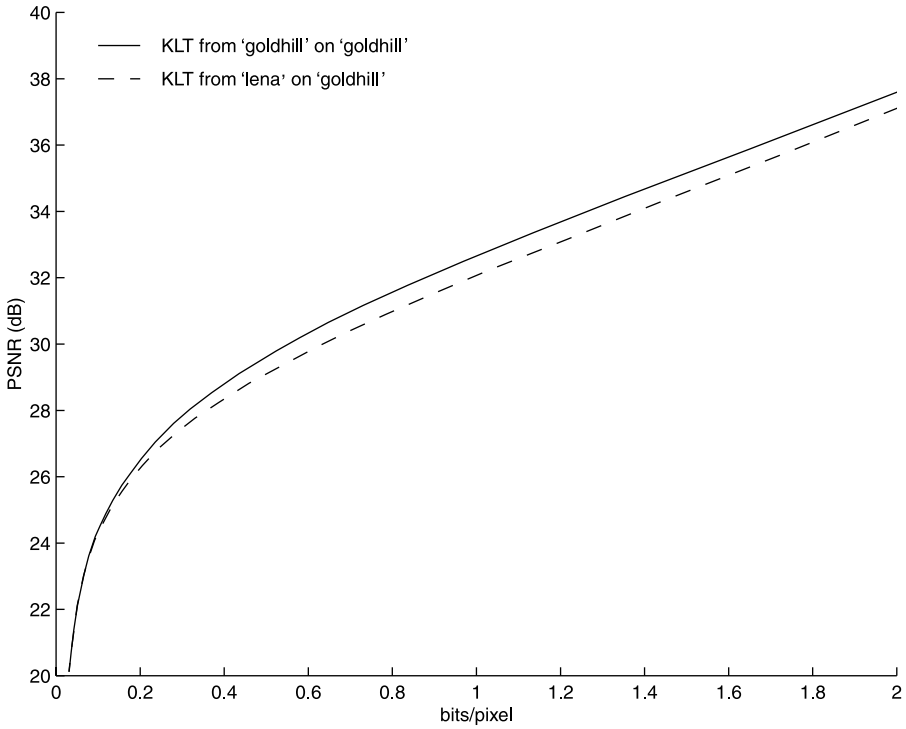

FIGURE 1.18
Distortion versus bit rate for "Goldhill" image using KLT from both "Goldhill" image and "Lena" image.

The results of evaluation are summarized in Fig. 1.22 which shows the plot of the mean opinion score for both scoring criteria. The figure shows that the MOS at the various degrees of compression remains quite close to that of the original. For image quality, the MOS for the original is 4.28 and drops only to 4.01 at 40:1. The MOS for the pathology visibility is 4.33 for the original and 4.10 for the 40:1 compression ratio. Therefore the use of a compression method based on the KLT results in usable images at even relatively high compression.

1.4.6 Color Images

Another application of the decorrelation abilities of the KLT is the compression of color images. Color images can be represented by three color components per pixel. Typically these are the three primary colors, red, green, and blue (RGB), corresponding to the responses of the three color receptors in the retina of the human eye. Similarly, in most color vision systems, three color filters of red, green, and blue are used to produce, respectively, the three color components per pixel. From the original RGB data, there are numerous transformations that can represent color values

Table 1.3 Correlation Between First 8 Neighboring Pixels on the Rows

	$E[x_i x_j]$	$E[x_i x_j]/E[x_{i-1} x_j]$
$\|i-j\|=0$	2657	-
$\|i-j\|=1$	2589	0.9744
$\|i-j\|=2$	2472	0.9546
$\|i-j\|=3$	2338	0.9460
$\|i-j\|=4$	2223	0.9510
$\|i-j\|=5$	2111	0.9492
$\|i-j\|=6$	2010	0.9524
$\|i-j\|=7$	1914	0.9523

in different coordinate spaces [18]. Some, for example HSI, express the components in a form that follows more closely the human perceptions of color qualities such as hue, saturation, and intensity. Others, for example YIQ, attempt to decorrelate the chromatic and intensity information. For the following example, we will explore the use of the decorrelation property of the KLT on the raw RGB data.

A simple approach to compression would be to treat each of the three RGB components as separate images. However, this method does not exploit the correlation between the three color values at each pixel. An alternative is to include all three component pixel values within a block. For example, an 8×8 block will contain 192 individual values. The KLT can then decorrelate the component values allowing improved coding.

To show the difference in coding performance between combining and not combining the three component values, the image shown in Fig. 1.23 is used as a test image. The image is 512×768 pixels in size and each pixel has 3 RGB values of 8 bits each for a total of 24 bits per pixel. For the separate encoding, three transforms were calculated and applied, one for each component. The resulting rate-distortion relationship is shown as the dashed curve in Fig. 1.24. The bit rate combines the file sizes of all three components and the distortion is the mean across the components. For the combined method, the image was divided into blocks of 8×8 pixels $\times$ 3 components for a total input dimension of 192. The performance of the KLT generated from this data is shown by the solid curve of Fig. 1.24. The figure shows that the difference in performance is substantial. For example, at a compression of 12:1 (2 bits per pixel), allowing the transform to decorrelate the RGB components results in a 4 dB increase in fidelity. Again, this example shows that the greater the decorrelation, the better the performance of the transform.

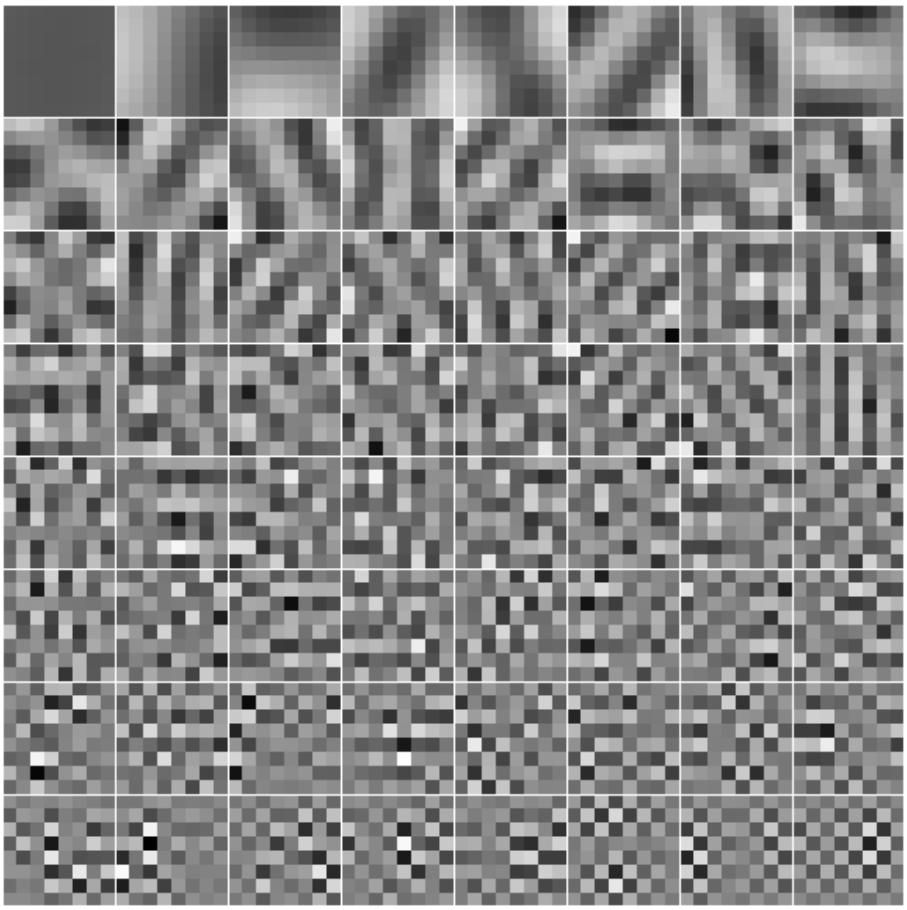

FIGURE 1.19
KLT basis images for Markov-1 model, $\rho = 0.9543$.

1.5 Summary

The Karhunen-Loève transform (KLT) is defined as the linear transformation whose basis vectors are the eigenvectors of the covariance matrix of the data. As it diagonalizes the covariance matrix, it decorrelates the data. The resulting set of coefficients can be encoded with fewer bits for a given distortion than the raw data.

The KLT is the optimal transformation in terms of minimizing the bit rate. The use of eigenvectors as the basis vectors ensures that the variance of the first coefficient is maximized, and, subject to the orthogonality of basis vectors, all subsequent coefficient variances are maximized in order. Maximizing each variance means that

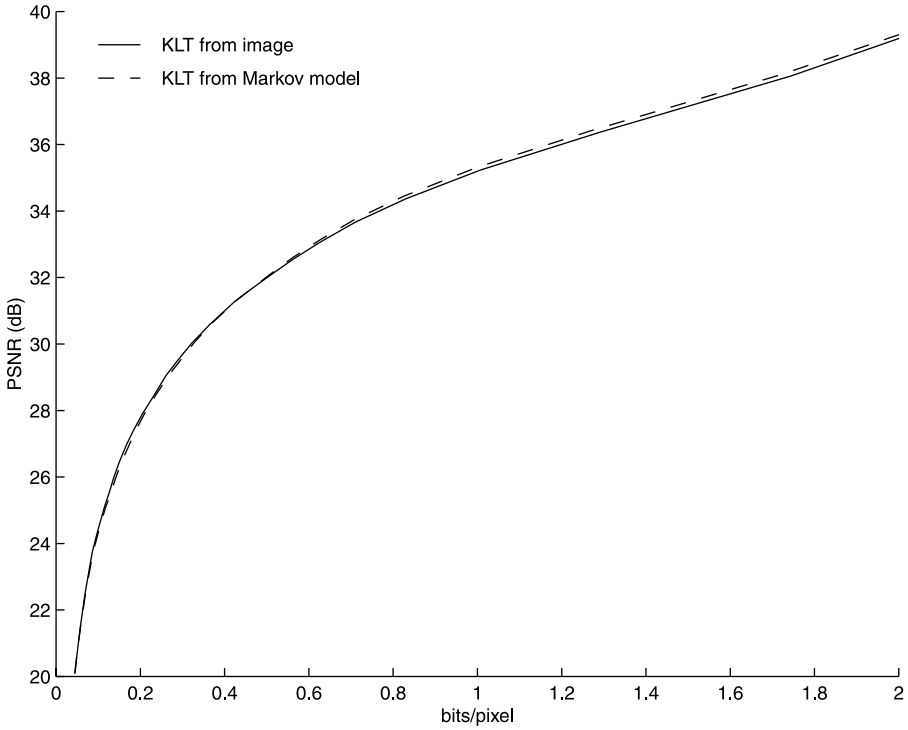

FIGURE 1.20
Plot of distortion (PSNR) versus bit rate for the KLT from the image covariance matrix and the KLT generated from the Markov-1 model.

the product of all the variances is minimized due to the energy preserving nature of any orthonormal transformation. Since the total differential entropy for the blocks increases with the product of the variances, the KLT minimizes the entropy thereby minimizing the bound on the bit rate.

The transform has a number of important performance characteristics for image compression. At moderate compression ratios, very little distortion is visible. As the compression ratio increases, more distortion becomes evident. However, because the transform is based on data from the image, some areas remain faithfully reproduced at even relatively low bit rates. The most prominent feature of the distortion as the compression ratio increases is the blocking effects of using finite sized blocks. While the KLT is calculated from the covariance matrix of an image and the covariances of different images are rarely identical, the transform based on one image can still perform well on a different image since the second order statistics of many images are rather similar. Even the use of the quite general Markov-1 model for the covariance results in performance almost as effective as the strictly image-specific transformation. As well, the decorrelating property of the transform can be used successfully on pixel

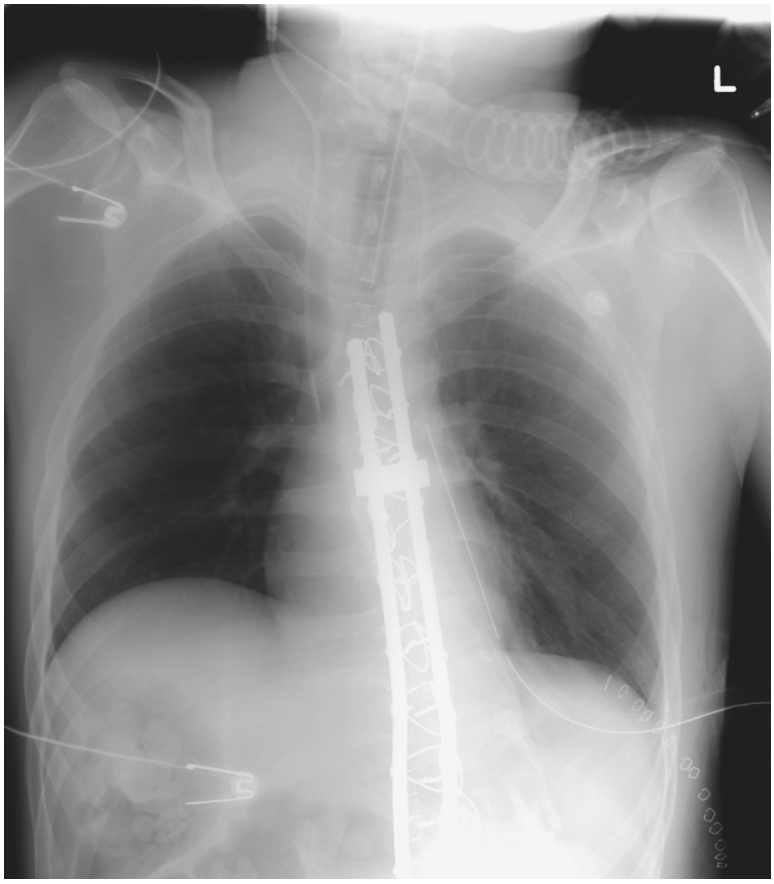

FIGURE 1.21
Sample chest radiograph for medical image compression evaluation.

data with more than one component, such as the three RGB components in color images.

While the KLT has the theoretically optimal decorrelation property, it has seldom been used in practice. While the transform can generalize well, the basis vectors must accompany an image or set of images for reconstruction if the Markov-1 model is not used. There are also the additional computational requirements of estimating the covariance and solving the eigensystem to extract the principal components. Further, the computation of the forward and inverse transform is considered "slow," requiring an order of $O(N^2)$ operations per block of N pixels or $O(N \times p)$ for an image of p pixels. Finally, while the transform may be optimal from an information-theoretic basis, the distortion criterion may not correspond well with our visual perception of distortion. For example, the block effect distortion is quite visible at high compression

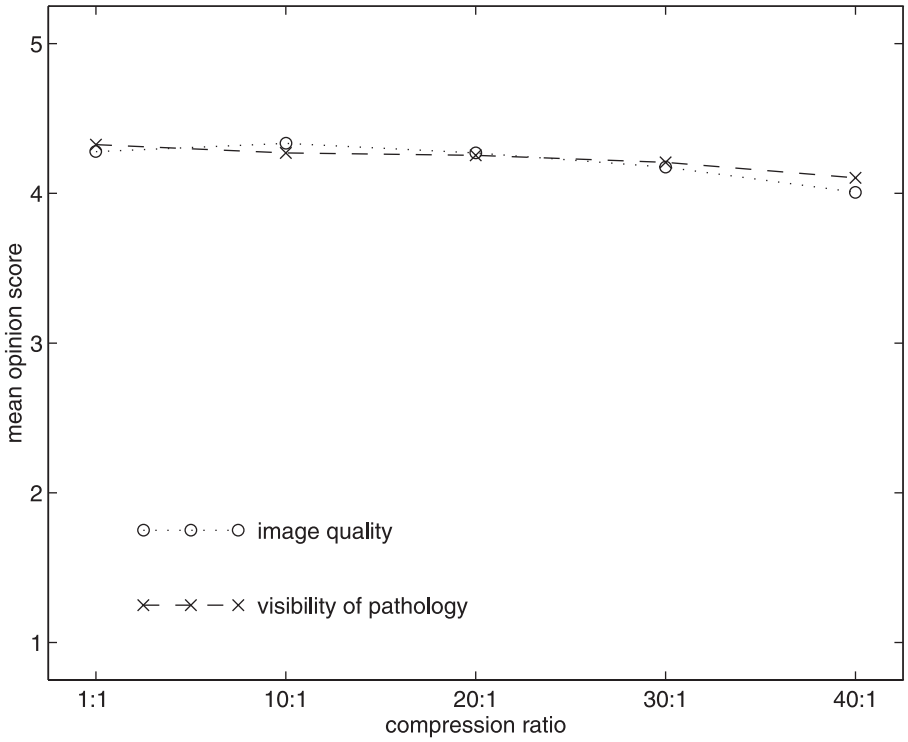

FIGURE 1.22
Mean opinion score across all images and evaluators.

FIGURE 1.23
Color test image, "Monarch."

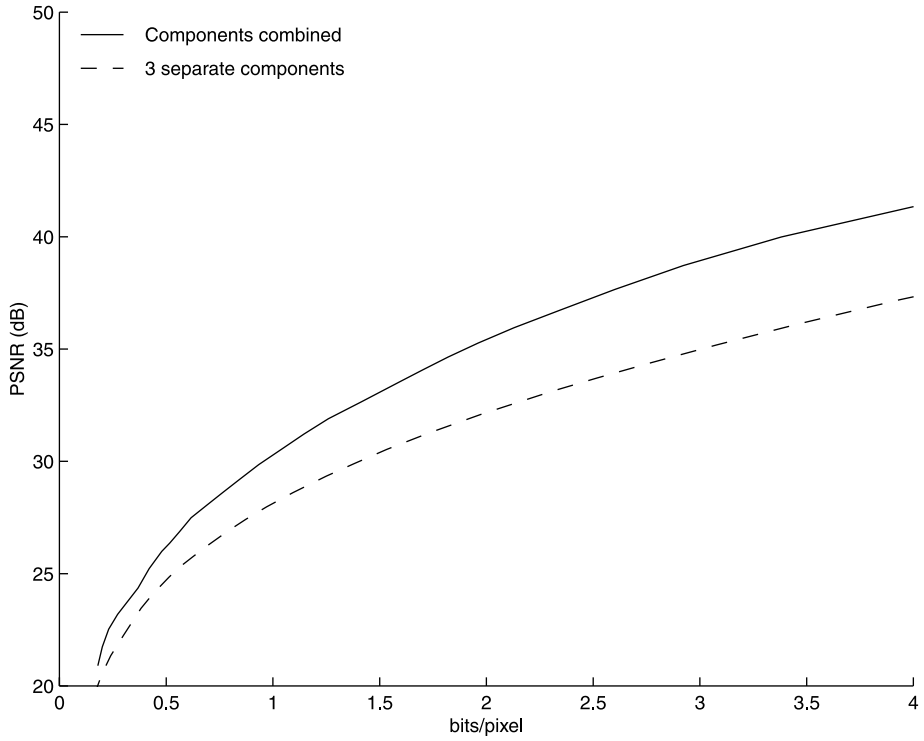

FIGURE 1.24
Distortion versus bit rate for "Monarch" image for encoding the RGB components separately and together.

ratios, yet it is not accounted for in the distortion criteria. A full frame KLT is theoretically possible, but it is only practical for sets of quite small images.

References

[1] Berger, T., *Rate Distortion Theory,* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[2] Castleman, K.R., *Digital Image Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1996.

[3] Chatterjee, C., Roychowdhury, V.P., and Chong, E.K.P., On relative convergence properties of principal component analysis algorithms, *IEEE Trans. Neural Networks,* 9(2):319–329, 1998.

[4] Chen, T., Hua, Y., and Yan, W.-Y., Global convergence of Oja's subspace algorithm for principal component extraction, *IEEE Trans. Neural Networks,* 9(1):58–67, 1998.

[5] Clarke, R.J., *Transform Coding of Images,* Academic Press, San Diego, CA, 1985.

[6] Clarke, R.J., *Digital Compression of Still Images and Video,* Academic Press, San Diego, CA, 1995.

[7] Diamantaras, K.I. and Kung, S.Y., *Principal Component Neural Networks: Theory and Applications,* John Wiley & Sons, New York, 1996.

[8] Dony, R.D. and Haykin, S., Optimally adaptive transform coding, *IEEE Trans. Image Processing,* 4(10):1358–1370, 1995.

[9] Dony, R.D., Haykin, S., Coblentz, C., and Nahmias, C., Compression of digital chest radiographs using a mixture of principal components neural network: an evaluation of performance, *RadioGraphics,* 16, 1996.

[10] Gersho, A. and Gray, R.M., *Vector Quantization and Signal Compression,* Kluwer Academic Publishers, Norwell, MA, 1992.

[11] GNU Octave, `http://www.che.wisc.edu/octave`.

[12] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing,* Addison-Wesley, Reading, MA, 1993.

[13] Gray, R.M., *Source Coding Theory,* Kluwer Academic Publishers, Norwell, MA, 1990.

[14] Haykin, S., *Neural Networks: A Comprehensive Foundation,* Macmillan, New York, 1994.

[15] Hotelling, H., Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.,* 24:417–447, 498–520, 1933.

[16] Jolliffe, I., *Principal Component Analysis,* Springer-Verlag, New York, 1986.

[17] Karhunen, K., Über lineare methoden in der wahrscheinlich-keitsrechnung. *Ann. Acad. Sci. Fennicea,* Ser. A137, 1947. (Translated by Selin, I. in "On Linear Methods in Probability Theory," Doc. T-131, The RAND Corp., Santa Monica, CA, 1960.)

[18] Levkowitz, H., *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications,* Kluwer Academic Publishers, Norwell, MA, 1997.

[19] Loève, M., Fonctions Aléatoires de second order, In Lévy, P., Ed., *Processus Stochastiques et Movement Brownien,* Hermann, Paris, 1948.

[20] MathWorks, `http://www.mathworks.com`.

[21] Netlib Repository, `http://www.netlib.org/eispack`.

[22] Netlib Repository, `http://www.netlib.org/lapack`.

[23] Netlib Repository, `http://www.netlib.org/linpack`.

[24] Netravali, A.N. and Haskell, B.G., *Digital Pictures: Representation and Compression,* Plenum Press, New York, 1988.

[25] Oja, E., A simplified neuron model as a principal component analyzer, *J. Math. Biology,* 15:267–273, 1982.

[26] Oja, E., Neural networks, principal components, and subspaces, *Int. J. Neural Systems,* 1(1):61–68, 1989.

[27] Oja, E. and Karhunen, J., On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix, *J. Math. Analysis and Applications,* 106:69–84, 1985.

[28] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes in C: The Art of Scientific Computing,* Cambridge University Press, Cambridge, UK, 1988.

[29] Rao, K.R. and Yip, P., *Discrete Cosine Transform: Algorithms, Advantages, Applications,* Academic Press, New York, 1990.

[30] Ray, W. and Driver, R.M., Further decomposition of the Karhunen-Loève series representation of a stationary random process, *IEEE Trans. Information Theory,* IT-16:663–668, 1970.

[31] Research Systems, `http://www.rsinc.com`.

[32] Rosenfeld, A. and Kak, A.C., *Digital Picture Processing,* Vol. I & II, 2nd ed., Academic Press, San Diego, CA, 1982.

[33] Sanger, T.D., Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks,* 2:459–473, 1989.

[34] Shannon, C.E., A mathematical theory of communication, *The Bell System Technical J.,* 27(3):379–423, 623–656, 1948.

[35] Solo, V. and Kong, X., Performance analysis of adaptive eigenanalysis algorithms, *IEEE Trans. Signal Processing,* 46(3):636–645, 1998.

[36] Wallace, G.K., The JPEG still image compression standard, *Communications of the ACM,* 34(4):30–44, 1991.

[37] Wong, S., Zaremba, L., Gooden, D., and Huang, H.K., Radiologic image compression — A review, *Proc. IEEE,* 83(2):194–219, 1995.