Lay, Jose A. and Ling Guan “Image Retrieval in Frequency Domain Using DCT Coefficient Histograms”
*Multimedia Image and Video Processing*
Ed. Ling Guan et al.
Boca Raton: CRC Press LLC, 2001
Chapter 14

Image Retrieval in Frequency Domain Using DCT Coefficient Histograms

Jose A. Lay and Ling Guan

14.1 Introduction

As an increasing amount of multimedia data is distributed, used, and stored in the compressed format, an intuitive approach for lowering the computational complexity toward the implementation of an efficient content-based retrieval application is to propose a scheme that is able to perform retrieval directly in the compressed domain. In this chapter, we show how energy histograms of the lower frequency discrete cosine transform coefficients (LF-DCT) can be used as features for the retrieval of JPEG images and the parsing of MPEG streams. We also demonstrate how the feature set can be designed to cope with changes in image representation due to several common transforms by harvesting manipulation techniques known to the DCT domain.

14.1.1 Multimedia Data Compression

Loosely speaking, multimedia data compression includes every aspect involved in using a more economical representation to denote multimedia data. Hence, to appreciate the significance of multimedia data compression, the types and characteristics of a multimedia datum itself need to be examined.

Multimedia is a generic term used in computing environments. In the digital information processing domain, it refers to a data class assembled by numerous independent data types, such as text, graphics, audio, still images, moving pictures, and other composite data types. A composite data type is a derivative form created when instances of two or more independent data types are combined to form a new medium. Multimedia data are used in a broad range of applications. Sample applications include telemedicine, video telephony, defense object tracking, and Web TV.

Multimedia data can be categorized in numerous ways. With respect to the authoring process, multimedia data can be classified as synthesized and captured media. The classification can also be based on the requirement of presentation where multimedia data are grouped into discrete and continuous media.

Multimedia data have several characteristics. They are generally massive in size, which requires considerable processing power and large storage supports. Storage requirements for
several uncompressed multimedia data elements are listed in Table 14.1. Where continuous media are used, multimedia data will also be subjected to presentation constraints such as media synchronization and the requirement for continuity in operation [1]. Consequently, the bandwidth requirement for a synchronized media unit will be the aggregate bandwidth along with the synchronization overhead. For instance, a musical video clip in common intermediate format (CIF) may consist of a series of synchronized video, text, and audio data. This media may need to be continuously played for 3 min, and the aggregate bandwidth requirement excluding the overhead will be a steady throughput of 74.4 Mbps over the 3-min time span. Thus, a system with a storage space of not less than 1.67 GB and a data bus of sustainable 9.3 MB/s will be required to present this media. Although the storage requirement may not seem to be too ambitious, the data rate is certainly beyond the capability of the fastest CD-ROM drive embedded in the current PCs.

<table>
<thead>
<tr>
<th>Multimedia Data Type</th>
<th>Sampling Details</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo audio (20–20 KHz)</td>
<td>44,000 samples/s × 2 channels × 16 bit/sample</td>
<td>1.41 Mbps</td>
</tr>
<tr>
<td>VGA image (640 × 480)</td>
<td>640 × 480 pixels × 24 bit/pixel</td>
<td>7.37 Mbit/image</td>
</tr>
<tr>
<td>Digitized video (NTSC)</td>
<td>720 × 576 pixels/frame × 24 bit/pixel × 30 frames/s</td>
<td>298.59 Mbps</td>
</tr>
<tr>
<td>Web TV video (CIF)</td>
<td>352 × 288 pixels/frame × 24 bit/pixel × 30 frames/s</td>
<td>72.99 Mbps</td>
</tr>
</tbody>
</table>

Furthermore, as the Internet facilitates the multimedia data from a workstation onto the widely networked environment, the network bandwidth will also need to match the throughput requirement. The transmission rates for several network standards are listed in Table 14.2. It is clear that the uncompressed multimedia data are scarcely supported by local area networks (LANs), let alone a connection established through the public switched telephone network (PSTN).

<table>
<thead>
<tr>
<th>Network Technology</th>
<th>Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public switched telephone network (PSTN)</td>
<td>0.3–56 Kbps</td>
</tr>
<tr>
<td>Integrated services digital network (ISDN)</td>
<td>64–144 Kbps</td>
</tr>
<tr>
<td>Telecommunication T-1</td>
<td>1.5 Mbps</td>
</tr>
<tr>
<td>Telecommunication T-3</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Local area network (Ethernet)</td>
<td>10 Mbps/100 Mbps</td>
</tr>
</tbody>
</table>

To overcome the bulky storage and transmission bandwidth problems, a compressed form of multimedia data was introduced. Extensive compression may significantly reduce the bandwidth and storage need; however, compression may also degrade the quality of multimedia data, and often the loss is irreversible. Therefore, multimedia data compression may be viewed as a trade-off of the efficiency and quality problem [2, 3]. The data rates of several compressed multimedia data are listed in Table 14.3.

For the last decade, three very successful compression standards on multimedia data elements have been JPEG, MPEG-1, and MPEG-2. JPEG has facilitated the vast distribution of images
Table 14.3  Sample Data Rates of Compressed Data

<table>
<thead>
<tr>
<th>Compressed Data</th>
<th>Sampling Details</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel audio for MPEG</td>
<td>$64/128/192$ Kbps/channel $\times$ $N$ channel</td>
<td>$N \times 64/128/192$ Kbps</td>
</tr>
<tr>
<td>Color JPEG image (640 x 480)</td>
<td>$640 \times 480$ pixels $\times$ 3 color components $\times (0.25–2)$ bit/component sample</td>
<td>0.23–1.84 Mbit/image</td>
</tr>
<tr>
<td>MPEG-1 video</td>
<td>360 $\times$ 345 pixels/frame $\times$ 30 frames/s</td>
<td>1.5 Mbps or higher</td>
</tr>
<tr>
<td>H.261 video (CIF)</td>
<td>352 $\times$ 288 pixels/frame $\times$ (15–30) frames/s</td>
<td>56 Kbps–2 Mbps</td>
</tr>
</tbody>
</table>

on the Internet. MPEG-1 has made possible the storage of a movie title onto a couple of video compact disc (VCD) media. MPEG-2 then extended this achievement to the forms of DVD and HDTV. Their success stories have been prominent ones [4]. Not only have they created many new opportunities in business, but they have also changed the computing experience by shortening the path in multimedia content authoring and usage. Digital images are now directly attainable from digital cameras in the JPEG format. Likewise, MPEG-1 and -2 streams can straightforwardly obtained from digital videocameras. In short, many more people have been enabled to create and use multimedia contents.

14.1.2 Multimedia Data Retrieval

The various aspects concerned with the enabling of multimedia data accessing are generally termed multimedia data retrieval (MDR). As increasing amounts of multimedia data or their elements become available in the digital format, information technology is expected to provide maximum usability of these data. However, the established text-based indexing schemes have not been feasible to capture the rich content of multimedia data, as subjective annotation may lead to undetectable similarities in the retrieval process. Consequently, content-based retrieval (CBR) was proposed. In addition to textual descriptors, multimedia data are described using their content information; color, texture, shape, motion vector, pitch, tone, etc., are used as features to allow searching to be based on rich content queries. The use of textual descriptors will still be desirable, because they are needed in identifying information that cannot be automatically extracted from multimedia contents, such as name of the author, date of production, etc.

Three basic modules of a CBR system are feature extraction, feature description, and proximity evaluation. Feature extraction deals with how the specific traits of content information can be identified and extracted from the content-rich data. Feature description specifies how those features can be described and organized for efficient retrieval processing. Lastly, proximity evaluation provides the specification in which similarities among contents can be measured based on their features.

The advancement of CBR studies has been remarkable. The current challenge has been the network-wide implementation of these techniques. In past years, many studies on CBR have been conducted, notably in the image and video retrieval domain. Numerous features and their associated description and proximity evaluation schemes have been introduced. A handful of proprietary CBR systems have also been developed. Content-based image and video retrieval is now considered a developed field. However, searching content information across the Internet has not been viable, because no unified feature description scheme has been commonly adopted. For this reason, an effort to standardize the description of content information was initiated. The work was named Multimedia Content Description Interface.
but is better known as MPEG-7. Surveys on CBR systems and their research issues are given in [5]–[7].

MPEG-7 aims to extend the capabilities of the current CBR systems by normalizing a standard set of descriptors that can be used to describe multimedia contents. MPEG-7 also intends to standardize ways to define other descriptors and their description schemes. MPEG-7 will also standardize a language to specify description schemes [8]. Information on MPEG-7 is available through the MPEG Web site [9].

Figure 14.1 depicts the scope of MPEG-7 [8]. Because the focal interest of MPEG-7 is on the interfacing of descriptors, the feature extraction process and how the features are used in searching on a database will remain an open area for industry competition, since their normalization is not required to allow interoperability.

![Diagram block of the proposed MPEG-7 scope.](image)

**FIGURE 14.1**
Diagram block of the proposed MPEG-7 scope.

### 14.1.3 About This Chapter

The chapter has been written to be self-contained. Background information is included to provide newcomers with the underlying concepts. Readers familiar with those concepts may skim through or skip them. Subjects are discussed within a general perspective. While the primary aim of this chapter concentrates on compressed domain image and video retrieval using the energy histogram features, efforts have been made to present the discussion under the broader theme of MDR, which allows relation with MPEG-7 to be easily established.

More studies on retrieval and processing models are needed to optimize the applicability of MPEG-7. Because only the interfacing of features is covered in MPEG-7, many more research opportunities and needs are left open for feature extraction and search engine studies. Furthermore, while feature extraction has been an active research subject in CBR studies, works on supporting retrieval models have been scarce. Therefore, it is sensible that many more studies on retrieval and processing models are needed before an MPEG-7 optimum search engine can be materialized. A brief discussion on retrieval and processing models as well as the perceived MPEG-7 search engine is consecutively presented in Sections 14.3.1, 14.3.2, and 14.3.3.

Meanwhile, as multimedia data are distributed, used, and stored in the compressed format, the compressed domain technique is highly relevant. The compressed domain technique deals with data directly (or through partial decompression) in their compressed domain, hence avoiding (or reducing) the decompression overhead found in many uncompressed domain schemes.

The DCT domain features are at the center of compressed domain techniques. A noteworthy observation on the many successful multimedia standards is that most of them are based on transform coding using the DCT. Logically, exploitations of DCT domain features are essential in supporting the realization of an efficient current CBR application. The underlying concepts of DCT, the DCT coefficients in JPEG and MPEG data, and the energy histogram are presented in Section 14.2, while an overview of the DCT domain features reported in recent studies is given in Section 14.3.5.
The rest of Section 14.3 will be used to present various aspects related to the proposed retrieval scheme. An MPEG-7 optimum search engine construct is presented in Section 14.3.3. The DCT domain manipulation techniques are covered in Section 14.3.4, while the energy histograms of the LF-DCT coefficient features are presented in Section 14.3.5. Several proximity evaluation schemes are discussed in Section 14.3.6, and the experimental results for the retrieval of JPEG images and the parsing of MPEG streams are provided in Section 14.3.7. Conclusions are given in Section 14.4.

14.2 The DCT Coefficient Domain

Before we describe how the DCT coefficients can be used as potent features for retrieving a JPEG image and/or parsing an MPEG stream, we shall first enunciate the basic concept underlying the DCT domain. We will embark by giving an alternative explanation of the DCT using the matrix notation, then go on to show how the DCT coefficients reside in the JPEG and MPEG data, and finally articulate the notion of the energy histograms of the DCT coefficients.

14.2.1 A Matrix Description of the DCT

DCT was first introduced in 1974 [10]. It is now the major building block of many very popular image and video compression standards. Together with the vast development of semiconductors, DCT-powered compression standards have delivered a magnificent computing environment, an internetworked world rich with multimedia contents.

The 8 × 8 block forward and inverse 2D DCTs used in JPEG and MPEG are given by Forward 2D DCT:

\[ f(u, v) = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} s(i, j) \cos \left( \frac{2i + 1}{16} u \pi \right) \cos \left( \frac{2j + 1}{16} v \pi \right) \]

Inverse 2D DCT:

\[ s(i, j) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v f(u, v) \cos \left( \frac{2i + 1}{16} u \pi \right) \cos \left( \frac{2j + 1}{16} v \pi \right) \]

where \( C_\tau = 1/\sqrt{2} \) for \( \tau = 0 \) and \( C_\tau = 1 \) for \( \tau \neq 0 \). The \( f(u, v) \) are the so-called DCT coefficients and \( s(i, j) \) are the values of the \( i, j \) input samples.

Since the 2D DCT is attainable by concatenating two 1D DCTs, we will use the latter to convey the purpose of this section. Thus, we can reveal the concept without dealing with too many indices in the equation. The formal definition for an 8-element 1D DCT is given by Forward 1D DCT:

\[ f(u) = \frac{1}{2} C_u \sum_{i=0}^{7} s(i) \cos \left( \frac{2i + 1}{16} u \pi \right) \]

where \( C_u = 1/\sqrt{2} \) for \( u = 0 \) and 1 otherwise, \( f(u) \) are the 8-element 1D DCT coefficients, and \( s(i) \) are the 8 input elements.
Thinking in vector terms, we can rewrite the transform using a matrix notation by arranging \( f(u) \) and \( s(i) \) and substituting values for \((2i + u)\):

\[
\begin{bmatrix}
  f_0 \\
  f_1 \\
  f_2 \\
  f_3 \\
  f_4 \\
  f_5 \\
  f_6 \\
  f_7
\end{bmatrix} = \frac{1}{2} \mathbf{C} \mathbf{u} \cos \begin{bmatrix}
  \pi/16 & \pi/16 & 5\pi/16 & 7\pi/16 & 9\pi/16 & 11\pi/16 & 13\pi/16 & 15\pi/16 \\
  2\pi/16 & 6\pi/16 & 10\pi/16 & 14\pi/16 & 18\pi/16 & 22\pi/16 & 26\pi/16 & 30\pi/16 \\
  3\pi/16 & 9\pi/16 & 15\pi/16 & 21\pi/16 & 27\pi/16 & 33\pi/16 & 39\pi/16 & 45\pi/16 \\
  4\pi/16 & 12\pi/16 & 20\pi/16 & 28\pi/16 & 36\pi/16 & 44\pi/16 & 52\pi/16 & 60\pi/16 \\
  5\pi/16 & 15\pi/16 & 25\pi/16 & 35\pi/16 & 45\pi/16 & 55\pi/16 & 65\pi/16 & 75\pi/16 \\
  6\pi/16 & 18\pi/16 & 30\pi/16 & 42\pi/16 & 54\pi/16 & 66\pi/16 & 78\pi/16 & 90\pi/16 \\
  7\pi/16 & 21\pi/16 & 35\pi/16 & 49\pi/16 & 63\pi/16 & 77\pi/16 & 91\pi/16 & 105\pi/16
\end{bmatrix} \begin{bmatrix}
  s_0 \\
  s_1 \\
  s_2 \\
  s_3 \\
  s_4 \\
  s_5 \\
  s_6 \\
  s_7
\end{bmatrix}.
\]

Let us denote the \( f(u) \) vector with \( \mathbf{f} \), the cosine function matrix as \( \mathbf{K} \), and the \( s(i) \) vector with \( \mathbf{s} \). We have:

\[
\mathbf{f} = \frac{1}{2} \mathbf{C} \mathbf{u} \mathbf{s}.
\]

Note that we have chosen to write \( \mathbf{f} \) and \( \mathbf{s} \) as column vectors in the equation. Intuitively, the matrix notation shows that a DCT coefficient \( f(u) \) is simply a magnitude obtained by multiplying a signal vector \( \mathbf{s} \) with several scaled discrete cosine values distanced at certain multiples of \( \pi/16 \) frequency. Therefore, calculating the DCT coefficients of a particular signal is essentially carrying out the frequency decomposition \([12]\) or, in a broader sense, the content decomposition of that signal.

Each row in the cosine function matrix represents the basis function of a specific decomposition frequency set. To help visualize this concept, we will reconstruct the cosine matrix by explicitly calculating their cosine values. Furthermore, since \( \pi/16 \) is a factor common to all elements, the trigonometric rules allow the new matrix to be rewritten using only references to the values of the first quadrant components. By doing so, we hope to communicate the idea without getting involved with long decimal elements in the matrix. We denote the first quadrant components of the \( \mathbf{K} \) matrix as:

<table>
<thead>
<tr>
<th>\cos 0</th>
<th>\cos \pi/16</th>
<th>\cos 2\pi/16</th>
<th>\cos 3\pi/16</th>
<th>\cos 4\pi/16</th>
<th>\cos 5\pi/16</th>
<th>\cos 6\pi/16</th>
<th>\cos 7\pi/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_0 )</td>
<td>( a_1 )</td>
<td>( a_2 )</td>
<td>( a_3 )</td>
<td>( a_4 )</td>
<td>( a_5 )</td>
<td>( a_6 )</td>
<td>( a_7 )</td>
</tr>
</tbody>
</table>

The new matrix may be rewritten as:

\[
\begin{bmatrix}
  f_0 \\
  f_1 \\
  f_2 \\
  f_3 \\
  f_4 \\
  f_5 \\
  f_6 \\
  f_7
\end{bmatrix} = \frac{1}{2} \mathbf{C} \mathbf{u} \begin{bmatrix}
  a_0 & a_0 & a_0 & a_0 & a_0 & a_0 & a_0 & a_0 \\
  a_1 & a_3 & a_5 & a_7 & -a_7 & a_5 & -a_3 & a_1 \\
  a_2 & a_6 & -a_6 & -a_2 & -a_2 & a_6 & a_2 & a_2 \\
  a_3 & -a_7 & -a_1 & -a_5 & a_5 & a_1 & a_7 & -a_3 \\
  a_4 & -a_4 & -a_4 & a_4 & a_4 & -a_4 & -a_4 & a_4 \\
  a_5 & -a_1 & a_7 & -a_3 & a_3 & -a_7 & a_1 & -a_5 \\
  a_6 & -a_2 & a_2 & -a_6 & -a_6 & a_2 & -a_2 & a_6 \\
  a_7 & -a_5 & a_3 & -a_1 & a_1 & -a_3 & a_5 & -a_7
\end{bmatrix} \begin{bmatrix}
  s_0 \\
  s_1 \\
  s_2 \\
  s_3 \\
  s_4 \\
  s_5 \\
  s_6 \\
  s_7
\end{bmatrix}.
\]

Note that the occurrence of sign changes increases as we move downward along the matrix rows. Row 0 has no sign changes, since \( a_0 = \cos 0 = 1 \) for every element in that row. However, row 1 has one sign change, row 2 has two sign changes, and so on. The sign changes within a basis function basically indicate the zero-crossings of the cosine waveform. Thus, as
FIGURE 14.2
Eight DCT cosine basis function waveforms.

the occurrence of the sign change intensifies, the frequency of the waveform increases. The eight basis function waveforms associated with matrix $K$ are shown in Figure 14.2.

Since the first cosine basis function ($K_0$) has no alternating behavior, the DCT coefficient associated with this basis function is usually dubbed as the DC coefficient, referring to the abbreviation used in electrical engineering for the direct current. Consequently, the other DCT coefficients are called AC coefficients.
Now that we view the DCT coefficients as the frequency domain apparatus of a time or spatial signal, we shall expect to reconstruct the original signal from its DCT coefficients, that is, to find the inverse DCT (IDCT). To do this, we will continue to use the matrix representation built in the previous section.

In a matrix sense, finding the IDCT can be just a matter of solving the inverse for the transforming operator. The matrix equivalent for the IDCT equation may be easily written as

\[ s = \left( \frac{1}{2} CuK \right)^{-1} f \]

which is basically a problem of finding an inverse for the scaled matrix. Note that we have neglected the $1/2Cu$ scaling constants in the discussion so far. We do so because it really does not inhibit us from passing on the ideas of frequency or content decomposition. However, as we move on with formulating the IDCT equation, the importance of these scaling constants will become relevant. In fact, a major reason for introducing the scaling constants is largely based on the requirement for having an orthogonal transforming matrix. So one can achieve the IDCT by merely transposing the matrix $1/2CuK$.

To demonstrate the role of the scaling constants, we will need to explore some characteristics of the basis functions. Examining the matrix $K$, we can see that each of the basis functions (matrix-row) is orthogonal to the others. This means a dot product of any two rows in the matrix will yield zero. However, none of the basis functions is orthonormal. In other words, the basis functions are not of unit vectors. Nevertheless, the results are almost as good.

\[
V = KK^T = \begin{bmatrix}
8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4
\end{bmatrix}
\]

Note that each of the diagonal elements in $V$ equals the dot product of a basis function with itself. $V_{0,0}$ is directly attainable in the form of the sum of squares of the first basis function ($K0$), whereas the others are computed by application of trigonometric identities.

Equally important are the zero-value elements in $V$, which indicate that orthogonality does exist among the basis function vectors. However, having orthogonal row vectors alone is not sufficient for $K$ to become an orthogonal matrix. An orthogonal matrix also requires all of its row vectors (column vectors) to be of unit length, so the product of the matrix with its transpose ($KK^T$) can produce an identity matrix ($I$).

Since $K$ is a square matrix and the orthogonal property exists among its basis function vectors, the only task left is to turn them into unit vectors, which can be realized simply by scaling each of the basis functions with its length:

\[ K_{iu} = \frac{Ki}{\|Ki\|} \quad (i = 0, 1, \ldots, 7) \quad \text{and} \quad \|K_{iu}\| = 1. \]

Simple arithmetic yields $\|k0\| = \sqrt{8} = 2\sqrt{2}$ and $\|k1\| \ldots \|k7\| = \sqrt{4} = 2$. Therefore, to make $K$ orthogonal, we shall scale the first basis function ($K0$) by a factor of $1/ (2\sqrt{2})$, while the others need only to be divided by 2. Separating the $1/2$ factor from the elements, $K_c$ can
be devised as:

\[
K_c = \frac{1}{2} \begin{bmatrix}
c_0 & c_0 & c_0 & c_0 & c_0 & c_0 & c_0 \\
a_1 & a_3 & a_5 & a_7 & -a_7 & -a_5 & -a_3 & -a_1 \\
a_2 & a_6 & -a_6 & -a_2 & -a_6 & a_6 & a_2 \\
a_3 & -a_7 & -a_1 & -a_5 & a_5 & a_1 & a_7 & -a_3 \\
a_4 & -a_4 & -a_4 & a_4 & -a_4 & -a_4 & a_4 \\
a_5 & -a_1 & a_7 & -a_3 & a_3 & a_7 & -a_1 & -a_5 \\
a_6 & -a_2 & a_2 & -a_6 & a_6 & a_2 & a_2 & a_6 \\
a_7 & -a_5 & a_3 & -a_1 & a_1 & -a_3 & a_5 & -a_7
\end{bmatrix},
\]

with \( c_0 = a_0/\sqrt{2} = 1/\sqrt{2} = a_4 \). \( K_c \) is an orthogonal matrix, where \( K_c^{-1} = K_c^T \), so \( K_c K_c^T = I \). It is quite interesting to see that we have just turned the highly sophisticated DCT into the down-to-earth linear equation:

Forward DCT: \( f = K_c s \)
Inverse DCT: \( s = K_c^{-1} f \Rightarrow s = K_c^T f \)

Treating the DCT as linear equations not only allows us to eagerly derive the IDCT term, but also clearly presents some of its most prominent properties by using solely the linear algebra concept.

From the linear algebra point of view, the DCT is an orthogonal linear transformation. A linear transformation means the DCT can preserve the vector addition and scalar multiplication of a vector space. Thus, given any two vectors \( \langle p \quad q \rangle \) and a scalar \( \alpha \), the following relations are true:

\[
f(p + q) = f(p) + f(q),
\]
\[
f(\alpha p) = \alpha f(p).
\]

Linearity is useful when dealing with frequency domain image manipulation. Several simple techniques of the DCT frequency domain image manipulation are discussed in Section 14.3.3. Meanwhile, the orthogonal term implies that the lengths of the vectors will be preserved subsequent to a DCT transformation. For an input vector \( s = [a, b, c, d, e, f, g] \), and the transformed vector \( f \), we have

\[
\|s\| = \|f\| = \sqrt{a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2}
\]

as \( \|f\|^2 = \|K_c s\|^2 = (K_c s)^T (K_c s) = s^T (K_c^T K_c) s = s^T s = \|s\|^2 \).

This characteristic is often referred to as the energy conservation property of the DCT.

Another important property of being an orthogonal transform is that the product of two or more orthogonal matrices will also be orthogonal. This property then enables a higher dimensional DCT and its inverse to be performed using less complicated lower dimensional DCT operations. An example for performing the \( 8 \times 8 \) block 2D DCT and its inverse through the 8-element DCT is given below:

Forward 2D DCT: \( f_{2D} = K_c s K_c^T \)

Inverse 2D DCT: \( s_{2D} = K_c^T f K_c \)

Now that we have described how the 2D DCT can be accomplished by successive operations of 1D DCT, we shall turn our attention to the most interesting behavior of the DCT known as
the energy packing property, which indeed has brought the DCT coefficients into the heart of several widely adapted compression techniques of the decade.

Even though the total energy of the samples remains unaffected subsequent to a DCT transform, the distribution of the energy will be immensely altered. A typical $8 \times 8$ block transform will have most of the energy relocated to its upper-left region, with the DC coefficient $(f_{00})$ representing the scaled average of the block and the other AC coefficients denoting the intensity of edges corresponding to the frequency of the coefficients. Figure 14.3 depicts the energy relocation that occurred in the transform of a typical $8 \times 8$ image data block.

\[
\begin{pmatrix}
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

FIGURE 14.3
Energy packing property.

Now that we have presented the fundamental idea of the DCT in the matrix dialect, we would like to end this section by taking just another step to rewrite the IDCT in its formal vein:

Inverse 1D DCT:

\[
s = K^{-1} f = K^T f
\]

\[
s(i) = \sum_{u=0}^{7} \frac{1}{2} C_u f(u) \cos \left(\frac{(2i + 1)u\pi}{16}\right)
\]

where

\[
\begin{align*}
C_u &= \frac{1}{\sqrt{2}} \quad \text{for} \quad u = 0 \\
C_u &= 1 \quad \text{for} \quad u > 0
\end{align*}
\]

14.2.2 The DCT Coefficients in JPEG and MPEG Media

Video can be viewed as a sequence of images updated at a certain rate. This notion is also valid in the compressed data domain. For instance, a sequence of JPEG images can be used to constitute a motion JPEG (M-JPEG) video stream. In fact, many popular video compression standards including MPEG-1, MPEG-2, H.261, and H.263 are built upon the DCT transform coding techniques developed and used in JPEG. Therefore, the topic of DCT coefficients in MPEG media will be best described after that of JPEG is presented.

The JPEG standard acknowledges two classes of encoding and decoding processes known as lossy and lossless JPEG. Lossy JPEG is based on the energy packing characteristic of DCT and includes mechanisms where a certain amount of information may be irreversibly lost subsequent to its coding processes. Lossy JPEG is able to achieve substantial compression rates. Its modes of operation are further divided into baseline, extended progressive, and extended hierarchical. Conversely, lossless JPEG is based on predictive algorithms where content information can be fully recovered in a reconstructed image. However, lossless JPEG can attain only a moderate compression rate. Because lossless JPEG is based on non-DCT-based algorithms, only lossy JPEG is of interest in this chapter.

Figure 14.4 shows a block diagram of the lossy JPEG codec structure. In the encoding process the spatial image data are grouped into a series of $8 \times 8$ blocks. Each of these blocks is then fed into a forward 2D DCT to produce the 64 DCT coefficients. The blocks
are processed in a sequence from left to right and from top to bottom. The DCT coefficients are then scalarly quantized using a quantization factor set in a quantization table [12]:

\[ f_q(u, v) = \text{round} \left( \frac{f(u, v)}{Q(u, v)} \right) \]

where \( f(u, v) \), \( f_q(u, v) \), and \( Q(u, v) \) are the DCT coefficients being quantized, their quantized values, and the quantization factors provided in the quantization table, respectively.

FIGURE 14.4
A JPEG codec structure.

The quantization step is the lossy part of lossy JPEG. Quantization is primarily employed to prune the higher frequency coefficients by dividing them with larger factors. Thus, variations of quantization tables can be used to tune the desirable compression ratio. However, because a rounding-off operation is involved in every quantization process, quantized coefficients may be subjected to irreversible loss of information. Therefore, quantization tables need to be specifically designed so that the quality degradation is still in the tolerable range. Separate quantization tables are used for the luminance and chrominance components of JPEG data. Two quantization tables provided in the JPEG standard are tabulated in Figure 14.5.

\[
\begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 13 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \\
\end{bmatrix}
\]

Luminance Quantization Table

\[
\begin{bmatrix}
17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\
18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\
24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\
\end{bmatrix}
\]

Chrominance Quantization Table

FIGURE 14.5
Quantization tables.

Upon quantization, the 8 × 8 DCT coefficients within a block are arranged in a zigzag order. Since the DC coefficients tend to be highly correlated among the adjacent data blocks, the difference of two consecutive DC coefficients (rather than an actual DC value) is coded to enhance the compression ratio, whereas the other AC coefficients are run-length coded to remove the “zeros” redundancy. These semicoded coefficients are then further entropy coded using Huffman or arithmetic coding techniques to produce the final JPEG bits. Conversely, on the decoding side, inverse operations of the encoding processes are performed.
In addition to the DCT-based transform coding mechanisms, color subsampling is used in JPEG to further enhance the compression rate. It is understood that human eyes are more sensitive to brightness (luminance) than to color (chrominance). Therefore, certain color information may be arbitrarily reduced from a color image without generating significant quality losses to human perceptions. Consequently, the YUV or YCbCr (rather than RGB) color representation system is adopted in JPEG and MPEG. The luminance (Y) component represents a gray-scale version of the image. The chrominance components (UV) are used to add color to the gray-scale image. One commonly used subsampling ratio is 4:2:2, which means that the luminance of each pixel is sampled while the chrominance of every two pixels is sampled. Several useful JPEG resources are provided in [13]–[15].

Because the updating rate of a video sequence is normally not less than tens of images per second, adjacent images in a video stream may be expected to be in high correlation. Therefore, temporal coding techniques can be used on top of the spatial coding to further enhance the compression performance.

Relying on both, MPEG adopted the intra- and inter-coding schemes for its data. An MPEG stream consists of I (intra), P (predictive), and B (bidirectional) coded frames. An I frame is an intra-coded independent frame. Spatial redundancy on independent frames is removed by the DCT coding where a coded image can be independently decoded. P and B frames are inter-coded reference frames. Temporal redundancy on reference frames is detached by the means of motion estimation. A P frame is coded based on its preceding I or P frame, while a B frame is coded using both of the preceding and the following I and/or P frames. Therefore, decoding a reference frame may depend on one or more related frames. A fine survey of the current and emerging image and video coding standards is presented in [16].

M-JPEG is an extension of JPEG to cope with moving pictures where each frame of a video stream is compressed individually using the JPEG compression technique. The independent compression allows easy random access to be performed on an M-JPEG stream, thus enabling M-JPEG to enjoy much popularity in the nonlinear video editing application.

### 14.2.3 Energy Histograms of the DCT Coefficients

Histogram techniques were originally introduced into the field of image retrieval in the form of color histograms [17]. A color (gray-level) histogram of a digital image is formed by counting the number of times a particular color (intensity) occurs in that image.

$$h[i] = n_i, \quad \begin{cases} h[i] = \text{color histogram of color } i \\ n_i = \text{number of times color } i \text{ occurs in the image} \end{cases}$$

Since color images are normally presented in a multidimensional color space (e.g., RGB or YUV), color histograms can be defined using either a multidimensional vector or several one-dimensional vectors.

The color histogram of an image is a compelling feature. As a global property of color distribution, color histograms are generally invariant to translation and perpendicular rotations. They can also sustain modest alterations of viewing angle, changes in scale, and occlusion [17]. Their versatility may also be extended to include scaling invariance through the means of normalization [18]. However, color histograms are intolerant to the changes of illumination. A small perturbation in the illumination may contribute to considerable differences in histogram data.

Similar to color histograms, an energy histogram of the DCT coefficients is obtained by counting the number of times an energy level appears in the DCT coefficient blocks of a DCT compressed image. Thus, the energy histograms for a particular color component ($h_c$) in an 8
× 8 DCT data block can be written as

\[ h_c[t] = \sum_{u=0}^{7} \sum_{v=0}^{7} \begin{cases} 1 & \text{if } E(f[u,v]) = t \\ 0 & \text{otherwise} \end{cases} \]

where \( E(f[u,v]) \) denotes the energy level of the coefficient at the \((u, v)\) location and \( t \) is the particular energy bin.

As with color histograms, energy histograms are generally tolerant to rotations and modest object translations.

### 14.3 Frequency Domain Image/Video Retrieval Using DCT Coefficients

Frequency domain CBR offers twofold advantages. Computational complexity issues introduced by the discrepancy of spatial domain (uncompressed) feature schemes and frequency domain (compressed) data can be a hindrance in implementing an efficient CBR application, especially on the real-time platform. The frequency domain CBR approach is able to reduce the complexity by processing the compressed data directly or through partial decompression in their frequency domain. Furthermore, direct processing of compressed data allows the retrieval system to operate on rather compact data, which are beneficial in terms of computation resources and network-wide processing.

Nevertheless, many more studies are needed before a full-fledged frequency domain CBR system can be materialized. Because conventional features and processing techniques may not be directly accessible in the compressed domain, exploration of new frequency domain features and processing techniques is becoming mandatory.

Repetition and modified variations of data are common in network environments. Since data sharing is likely in network computing, modified copies of a content are expected across network-based databases. For instance, image databases may contain many images that may differ only in their visual representation. These images are often of basic transformed operations (e.g., mirroring, transposing, or rotating). Therefore, detecting similarities on those transformed images is pertinent to a network-based CBR system.

Later in this section, the energy histograms of LF-DCT coefficients are used as features for retrieval of JPEG images (based on the query by model method) as well as for parsing of MPEG videos. The targeted retrieval scheme is desired to be able to support network- and MPEG-7-based implementation. Therefore, current content-based retrieval and processing
models and their requirements are studied. An MPEG-7 optimum search engine construct is also presented. Image manipulation techniques in the DCT domain are examined with regard to the building of limited transformed variant proof features.

![Figure 14.7](image-url)

**FIGURE 14.7**
Transformed variants are common in network databases.

In video applications, studies have shown that the DC coefficients can be used to detect abrupt scene changes. However, the use of DC coefficients alone does not provide a robust method for parsing of more complex video sequences such as ones with luminance changes and/or dissolving transitions. The energy histogram features are used to enhance the segmentation of DCT-based video. Experimental results for video parsing of MPEG streams along with the retrieval of JPEG images are presented in Section 14.3.7, while a CBR model for content-based video retrieval is briefly described in Section 14.3.1.

### 14.3.1 Content-Based Retrieval Model

The current CBR model is characterized by a separate feature database. To avoid the high computational cost posted by the uncompressed feature techniques, many of the current CBR systems are built on a dual-database model where a pair of independent databases are used to catalogue features and data [18, 19]. **Figure 14.8** shows the dual-database CBR model used in image retrieval applications. The independent feature database is built in addition to the image database itself during the setup phase. Proximity evaluation can be performed by contrasting the extracted features of a query with the records maintained in the feature database. When matches are obtained, the associated image data are returned from the image database. Therefore, the dual-database model is also known as the off-line or indexing model, because off-line feature extraction and pre-indexing processing are required during a database formation.

The dual-database model is advantageous from several perspectives. Since the structure of the dual-database model is comparable to the general indexing system used in text-based databases, this model may enjoy the support of many established techniques and developed tools. Furthermore, because features are pre-extracted during database creation, conventional spatial domain techniques may be used without causing high computational complexities at run time. The dual-database model also fits well with the needs of the video retrieval application, where features from key frames representing the segmented video shots are extracted for indexing use. The content-based video retrieval model is discussed later in this section.

Nevertheless, there are also drawbacks attached to the dual-database model. Because searching in a dual-database CBR system is performed on the pre-extracted feature sets, the query’s features have to conform with the feature scheme used by the feature database. Consequently, choices of features are determined by the in-search feature database. Moreover, universal
FIGURE 14.8
The dual-database content-based image retrieval model.

FIGURE 14.9
The single-database content-based image retrieval model.

searching across the Internet would also be impracticable until the unified description sought by MPEG-7 is widely implemented.

Alternatively, the single-database CBR model [20] can be employed. The single-database CBR model used in image retrieval applications is illustrated in Figure 14.9. In such a model, no preprocessing is required during database construction. Features are extracted on the fly within a retrieval cycle directly from data. Therefore, rapid feature extraction and proximity evaluation are obligatory to the single-database systems. Because feature extraction and proximity evaluation are executed on the fly within a retrieval cycle, the single-database CBR model is also known as the online CBR model.

As with the dual-database model, the single-database model also has upsides and downsides. It is practical for compressed domain-based retrieval (pull application) and filtering (push application), especially when content-based coding such as that of MPEG-4 is used. It also supports ad hoc Internet-wide retrieval implementations because raw compressed data can be
read and processed locally at the searcher machine. This local processing of feature extraction will unlock the restriction of the choices of features imposed by feature databases. However, sending raw compressed data across a network is disadvantageous, because it tends to generate high traffic loads.

Video retrieval is generally more efficient to implement with the dual-database model. Video streams are segmented into a number of independent shots. An independent shot is a sequence of image frames representing a continuous action in time and space. Subsequent to the segmentation, one or more representative frames of each of the segmented sequences are extracted for use as key frames in indexing the video streams. Proximity evaluations can then be performed as that of a dual-database image retrieval system (i.e., by contrasting the query frame with each of the key frames). When matches are obtained, relevant video shots are returned from the video database. The structure of a simplified content-based video database is shown in Figure 14.10.

![Figure 14.10](image)

**FIGURE 14.10**
Structure of a simplified content-based video database.

### 14.3.2 Content-Based Search Processing Model

From the search processing perspective, two fundamental models can be associated with the dual-database and single-database CBR applications. For the dual-database systems, search processing is normally performed on and controlled by the database-in-search. Thus, the associated search processing model is termed the archivist processing model, since proximity evaluation is executed on the archivist environments. The model is also known as the client–server model, because all the processing and know-how is owned and offered by the archivist (server) to the searcher (client). Conversely, on a single-database system, search processing is normally performed on and controlled by the search initiator. Therefore, the associated search processing model is termed the searcher processing model. The archivist processing model and the searcher processing model are illustrated in Figures 14.11a and b, respectively.

The current search processing models are unsatisfactory. The client–server model is undesirable because all the knowledge on how a search is performed is owned and controlled by the archivist server. The searcher processing model is impractical because its operation may involve high network traffic.

Alternatively, a paradigm called the search agent processing model (SAPM) [22] can be employed. The SAPM is a hybrid model built upon the mobile agent technology. Under
the SAPM, an agent (a traveling program) can be sent to perform feature extraction and/or proximity evaluation on remote databases. Figure 14.12 illustrates the sending of a mobile search engine to an SAPM-enabled database host.

14.3.3 Perceiving the MPEG-7 Search Engine

One way to perceive the characteristics of an MPEG-7 optimum search engine is to build on the objectives, scope, requirements, and experimental model of MPEG-7 itself. Since the aims and scope have already been presented in Section 14.1.2, only the remaining issues are covered in this section.

MPEG-7 is intended to be generic. It will support pull and push applications in both real-time and non-real-time platforms. In push applications, the MPEG-7 description can be used to filter information contents such as in automatic selection of programs based on a user profile. Likewise, in pull applications, the description can be used to locate multimedia data stored on distributed databases based on rich queries.

Although MPEG-7 aims to extend the proprietary solution of content-based applications, the descriptions used by MPEG-7 will not be of the content-based features alone. Because MPEG-7 is going to address as broad a range of applications as possible, a large number of description schemes will be specified and further amendment will be accommodated. In general, description of an individual content can be classified into content-based and content identification categories [39]. The content-based description includes descriptors (Ds) and description schemes (DSs) that represent features that are extracted from the content itself.
The content identification description covers the Ds and DSs that represent features that are closely related but cannot be extracted from the content. In addition, there will also be Ds and DSs for collection of contents as well as for the application-specific descriptions. The many DSs included in the current version of the generic audiovisual description scheme (generic AVDS) \[38\] are depicted in Figure 14.13.

The syntactic DS is used to specify the physical structures and signal properties of an image or a multimedia stream. Features such as shots, regions, color, texture, and motion are described under this DS category. The semantic DS is used to specify semantic features that appear in an image or a multimedia stream. Semantic notions such as object and event are described in this DS group. The relations between the syntactic and semantic descriptions are established using the syntactic/semantic link DS. Meta-information relating to media (storage, format, coding, etc.), creation (title, authors, etc.), and usage information (rights, publication, cost of usage, etc.) are, respectively, described in the media info DS, creation info DS, and usage info DS. The summarization DS is used to specify a set of summaries to allow fast browsing of a content. And the model DS is used to provide a way to denote the relation of syntactic and semantic information in which the contents are closely related to interpretation through models.

A wide-ranging choice of description schemes will allow a content to be described in numerous fashions. The same content is likely to be differently described according to the application and/or the user background. A content may also be described in the multiple-level description approach. Therefore, it will remain a challenging task for the MPEG-7 search engine to infer similarity among versions of the description flavors.

The descriptions of a content will be coded and provided as an MPEG-7 file or stream \[40\]. This file may be co-located or separately maintained with respect to the content. Likewise, the MPEG-7 stream may be transmitted as an integrated stream, in the same medium, or through a different mechanism with regard to the associated content. Access to partial descriptions is intended to take place without full decoding of the stream. The MPEG-7 file component is

FIGURE 14.13
MPEG-7’s generic audiovisual description scheme (AVDS).
clearly represented in the experimental model (XM) architecture shown in Figure 14.14. Note that the shaded blocks are the normative components.

FIGURE 14.14
The MPEG-7 experimental model architecture.

The XM is the basis for core experiments in MPEG-7. It is also the reference model for the MPEG-7 standard [41]. Therefore, it is apparent that the indexing or dual-database model will be a better fit, since descriptions may be independently maintained in MPEG-7.

MPEG-7 will also specify mechanisms for the management and protection on intellectual property of its descriptions. It is possible that only requests equipped with proper rights will be allowed access to certain descriptions in an MPEG-7 stream.

A candidate model for the MPEG-7 proper search engine can be based on the meta-search engine [42]–[44]. However, the meta-search engine model is lacking many functions needed in coping with the complexity of the MPEG-7 description. It also lacks efficient mechanisms for controlling the search on a remote search engine. The integration of computational intelligence tools is by no means easy. Therefore, a new search engine type will be needed. Given the distributed characteristic of the MPEG-7 databases, there has been consideration to base the MPEG-7 XM on the COM/DCOM and CORBA technologies. The meta-search engine and the MPEG-7 optimum search tool (MOST) [23] models are shown in Figures 14.15 and 14.16, respectively.

14.3.4 Image Manipulation in the DCT Domain

In Section 14.2.1, it was presented that the DCT is just a linear transform. The linear property allows many manipulations on the DCT compressed data to be performed directly in the DCT domain. In this section, we show how certain transformations of JPEG images can be accomplished by manipulating the DCT coefficients directly in the frequency domain. In addition, a brief description of the algebraic operations will first be presented. Several works on the DCT compressed domain-based manipulation techniques are given in [31]–[33].

Compressed domain image manipulation is relevant, because it avoids the computationally expensive decompression (and recompression) steps required in uncompressed domain processing techniques. Furthermore, in applications where lossy operators are involved, as in
the baseline JPEG, avoiding the recompression step is crucial to spare the image from further degradation due to lossy operations in the recompression process. Thus, direct compressed domain manipulation is lossless in nature. Lossless manipulation is highly appreciated, as it is in accordance with the preservation characteristic of the digital data.

Retracting the linear property in Section 14.2.1, we now show how the algebraic operations of images can be attained directly in the DCT coefficient domain. Let \( p \) and \( q \) be the uncompressed images with \((i, j)\) as the spatial domain indices, while \( P \) and \( Q \) represent the corresponding DCT compressed images with \((u, v)\) as the DCT frequency domain indices, and \( \alpha \) and \( \beta \) as scalars. Several algebraic operations can be written as follows:
Pixel addition:

\[ f(p + q) = f(p) + f(q) \]

\[ p[i, j] + q[i, j] \Rightarrow P[u, v] + Q[u, v] \]

Scalar addition:

\[ p[i, j] + \beta \Rightarrow \begin{cases} P[u, v] + 8\beta & \text{for } [u, v] = (0, 0) \\ P[u, v] & \text{for } [u, v] \neq (0, 0) \end{cases} \]

Scalar multiplication:

\[ f(\alpha p) = \alpha f(p) \]

\[ \alpha p[i, j] \Rightarrow \alpha P[u, v] \]

Note that \( f \) stands for the forward DCT operator. The addition of a constant to the uncompressed image data \( p \) will only affect the DC coefficient in \( P \), since no intensity change is introduced to the image data by the scalar addition. The algebraic operations of addition and multiplication for each of the scalar and pixel functions for the JPEG data are provided in [31]. Additionally, using the DCT coefficients, several transformations such as mirroring or flipping, rotating, transposing, and transversing of a DCT compressed image can be realized directly in the DCT frequency domain. The transformation is realized by rearranging and adjusting the DCT coefficients with several simple linear operators such as permutation, reflection, and transpose matrices.

Let \( Q \) be the JPEG compressed DCT coefficient block, \( D \) be a diagonal matrix of \([1, -1, 1, -1, 1, -1]\), and \( Q_\theta, Q_{HM}, \) and \( Q_{VM} \), respectively, stand for the \( \theta \) angle rotated, horizontal mirror, and vertical mirror of \( Q \). The various rotational and mirroring operations are given in [33] as

Mirroring:

\[ Q_{HM} = QD \]
\[ Q_{VM} = DQ \]

Rotation:

\[ Q_{90} = DQ^T \]
\[ Q_{-90} = Q^T D \]
\[ Q_{180} = DQD \]

Horizontal and vertical mirroring of a JPEG image can be obtained by swapping the mirror pairs of the DCT coefficient blocks and accordingly changing the sign of the odd-number columns or rows within each of the DCT blocks. Likewise, transposition of an image can be accomplished by transposing the DCT blocks followed by numerous internal coefficient transpositions. Furthermore, transverse and various rotations of a DCT compressed image can be achieved through the combination of appropriate mirroring and transpose operations. For instance, a 90° rotation of an image can be performed by transposing and horizontally mirroring the image. The JPEG lossless image rotation and mirroring processing is also described in [34]. A utility for performing several lossless DCT transformations is provided in [13].
14.3.5 The Energy Histogram Features

Before we proceed with the energy histogram features, several DCT domain features common to CBR applications such as color histograms, DCT coefficient differences, and texture features will be presented in this section. An overview of the compressed domain technique is given in [27].

Color histograms are the most commonly used visual feature in CBR applications. Since the DC coefficient of a DCT block is the scale average of the DCT coefficients in that DCT block, counting the histogram of the DC coefficient is a direct approximation of the color histogram technique in the DCT coefficient domain. The DC coefficient histograms are widely used in video parsing for the indexing and retrieval of M-JPEG and MPEG video [28, 29].

Alternatively, the differences of certain DCT coefficients can also be employed. In [30], 15 DCT coefficients from each of the DCT blocks in a video frame are selected to form a feature vector. The differences of the inner product of consecutive DCT coefficient vectors are used to detect the shot boundary.

Texture-based image retrieval based on the DCT coefficients has also been reported. In [24], groups of DCT coefficients are employed to form several texture-oriented feature vectors; then a distance-based similarity evaluation measure is applied to assess the proximity of the DCT compressed images. Several recent works involving the use of DCT coefficients are also reported in [20], [24]–[26].

In this work the energy histogram features are used. Because one of the purposes in this work has been to support real-time capable processing, computational inefficiency should be avoided. Therefore, instead of using the full-block DCT coefficients, we propose to use only a few LF-DCT coefficients in constructing the energy histogram features. Figure 14.17 shows the LF-DCT coefficients used in the proposed feature set.

![LF-DCT Coefficients](image)

**FIGURE 14.17**
LF-DCT coefficients employed in the features.

The reduction is judicious with respect to the quantization tables used in JPEG and MPEG. However, partial employment of DCT coefficients may not have allowed inheritance of the many favorable characteristics of the histogram method such as the consistency of coefficient inclusion in an overall histogram feature. Since it is also our aim to have the proposed retrieval system capable of identifying similarities in changes due to common transformations, the invariant property has to be acquired independently. To achieve this aim, we utilized the lossless transformation properties discussed in Section 14.3.4.

Bringing together all previous discussions, six square-like energy histograms of the LF-DCT coefficient features were selected for the experiment:
The square-like features have been deliberately chosen for their symmetry to the transpose operation, which is essential to the lossless DCT operations discussed in Section 14.3.4. Low-frequency coefficients are intended as they convey a higher energy level in a typical DCT coefficient block. F1 contains a bare DC component, whereas F2B, F3B, and F4B resemble the $2 \times 2$, $3 \times 3$, and $4 \times 4$ upper-left region of a DCT coefficient block. F2A and F3A are obtained by removing the DC coefficient from the F2B and F3B blocks. F2B, F3A, and F3B are illustrated in Figures 14.18a, b, and c, respectively. Note that counting the F1 energy histograms alone resembles the color histogram technique [17] in the DCT coefficient domain. The introduction of features F2A and F3A is meant to explore the contribution made by numerous low-frequency AC components, while the use of F2B, F3B, and F4B is intended for evaluating the block size impact of the combined DC and AC coefficients.

### Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Construction Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>F1F</td>
</tr>
<tr>
<td>F2A</td>
<td>F2F</td>
</tr>
<tr>
<td>F2B</td>
<td>F1F+F2F</td>
</tr>
<tr>
<td>F3A</td>
<td>F2F+F3F</td>
</tr>
<tr>
<td>F3B</td>
<td>F1F+F2F+F3F</td>
</tr>
<tr>
<td>F4B</td>
<td>F1F+F2F+F3F+F4F</td>
</tr>
</tbody>
</table>

![FIGURE 14.18](samples-of-the-square-like-features.png)

**Samples of the square-like features.**

#### 14.3.6 Proximity Evaluation

The use of energy histograms as retrieval features is also an advantageous approach from the perspective of proximity evaluation. In many cases, a computationally inexpensive distance-based similarity measure can be employed. Figure 14.19 illustrates the distance-based similarity measure among pairs of the histogram bins.

![FIGURE 14.19](bin-wise-similarity-measure.png)

**Bin-wise similarity measure.**
Several widely used proximity evaluation schemes such as the Euclidean distance, the city block distance, and the histogram intersection method will be described below. However, the underlying notion of the histogram space will be characterized first.

Because histograms are discretely distributed, each bin of the histogram can be thought of as a one-dimensional feature component (or coordinate) of the \( \mathbb{R}^n \), where \( n \) is the number of bins in the histogram. Furthermore, if we define the \( n \)-dimensional feature space \( \mathbb{R}^n \) as the histogram space \( (H^n) \), then every \( n \)-bin histogram feature can be represented as a point in that histogram space [17]. Consequently, for any pair of the histogram features, \( h_j \) and \( h_k \), the distance between the two histograms can be perceived as the distance of the two representative points in the histogram space. Thus, the distance between \( h_j \) and \( h_k \), \( D(h_j, h_k) \), can be defined to satisfy the following criteria [35]:

1. \( D(h_j, h_j) = 0 \) The distance of a histogram from itself is zero.
2. \( D(h_j, h_k) \geq 0 \) The distance of two histograms is never a negative value.
3. \( D(h_j, h_k) = D(h_k, h_j) \) The distance of two histograms is independent of the order of the measurement (symmetry).
4. \( D(h_j, h_l) \leq D(h_j, h_k) + D(h_k, h_l) \) The distance of two histograms is the shortest path between the two points.

Figure 14.20 illustrates two 2-bin histogram features and their distance, respectively, represented as two points and a straight line on the two-dimensional histogram space.

![Figure 14.20](image)

**FIGURE 14.20**

Histogram features and their distance on the histogram space.

In linear algebra, if \( Q \) and \( M \) are the feature vectors in the \( n \)-dimensional Euclidean space \( \mathbb{R}^n \): \( Q = (q_1, q_2, q_3, \ldots, q_n) \) and \( M = (m_1, m_2, m_3, \ldots, m_n) \), the Euclidean distance \( d_E(Q, M) \) between the \( Q \) and \( M \), written \( d_E(Q, M) \), is defined by:

\[
d_E(Q, M) = \sqrt{(q_1 - m_1)^2 + (q_2 - m_2)^2 + (q_3 - m_3)^2 + \cdots + (q_n - m_n)^2}.
\]

Correspondingly, the Euclidean distance of any pair of the histogram features can be defined in the histogram space \( (H^n) \):

\[
d_E(Q, M) = [(h_Q - h_M)^T (h_Q - h_M)]^{1/2} \quad \text{or} \quad d_E^2(Q, M) = \sum_{i=1}^{n} (h_Q[i] - h_M[i])^2
\]
where $d_E(Q, M)$ is the Euclidean distance between the two images, $h_Q$ and $h_M$ are the histograms of the two images, and $h_Q[t]$ and $h_M[t]$ represent the pairs of the histogram bins.

In general, Euclidean distance provides a very effective proximity measurement in image retrieval application. However, it is also a computationally expensive technique, especially when the floating point data type is involved, as with the DCT coefficient. The $d_E(Q, M)$ requires $n$ floating point multiplications, where $n$ is the number of bins in the histogram feature.

To overcome the high computational problem, the city block distance is usually employed. The city block distance ($d_{CB}$) is defined by

$$d_{CB}(Q, M) = \sum_{t=1}^{n} |h_Q[t] - h_M[t]|$$

where $d_{CB}(Q, M)$ denotes the city block distance between the two histogram features, $h_Q[t]$ and $h_M[t]$ are the pairs of the histogram bins, and $n$ is the number of bins in the histogram.

Figure 14.21 depicts an extreme condition where the two 2D histogram features under measurement are maximally apart. The Euclidean distance and the city block distance for the two feature points on the histogram space are computed below.

![FIGURE 14.21](image)

**FIGURE 14.21**
Maximum distances over two 2D histogram features.

The Euclidean distance is given by

$$d_E^2(Q, M) = \sum_{t=1}^{2} (h_Q[t] - h_M[t])^2 = 1 + 1 = 2$$

$$d_E(Q, M) = \sqrt{2}$$

and the city block distance is

$$d_{CB}(Q, M) = \sum_{t=1}^{2} |h_Q[t] - h_M[t]| = 1 + 1 = 2$$

$$d_{CB}(Q, M) = 2.$$ 

Alternatively, the histogram proximity evaluation scheme used in [17] can be employed. The histogram intersection method is used to locate the common parts of objects by intersecting the two images under similarity evaluation. For two images $Q$ (query) and $M$ (model) composed of $n$-bin histograms, the histogram intersection of the two images is defined by

$$\sum_{t=1}^{n} \min(h_Q[t], h_M[t])$$
where $h_Q[t]$ and $h_M[t]$ denote the particular pair of the histogram bins. Accordingly, the normalized distance of the histogram intersection can be written as

$$
d_h(Q, M) = 1 - \frac{\sum_{t=1}^{n} \min(h_Q[t], h_M[t])}{\min(\Gamma_Q, \Gamma_M)}
$$

where $d_h(Q, M)$ denotes the distance metric of the histogram intersection, $h_Q[t]$ and $h_M[t]$ denote a particular pair of the histogram bins, and

$$\Gamma_x = \sum_{t=1}^{n} h_x[t] \quad \text{with} \quad x = (Q, M).$$

Normalization can be used to add the scale invariance property to the histogram features [18]. Let $h_Q[t]$ be an $n$-bin histogram feature of an image. Then the normalized $n$-bin histogram feature, written as $h^n_Q[t_n]$, is defined as

$$h^n_Q[t_n] = \frac{h_Q[t]}{\sum_{t=1}^{n} h_Q[t]} \quad \text{for} \quad t = 1, 2, 3, \ldots, n.$$

The city block distance is used for the experiments in this work because it provides the lowest computational complexity and easy implementation.

### 14.3.7 Experimental Results

Experimental results for the use of energy histogram features on the retrieval of JPEG images and the parsing of MPEG video are presented in this section.

#### Image Retrieval

The experiment on image retrieval [20] was based on a single database containing nearly 4700 uncategorized, uniformly sized JPEG photographs of a broad range of real-life subjects. The collection is produced and maintained by Media Graphics International [36]. The DCT coefficients were extracted by utilizing the library provided in [13]. Dequantization steps were performed because quantization tables used in JPEG could vary among images. The retrieval system was built on the single-database model.

Some 40 query images were preselected by hand to ensure images appearing similar to the human visual system were properly identified in the database. Thresholding was not considered, because the purpose was to reveal the prerecognized images’ position on the retrieved image list. A sample of two query images and their similar associates are shown in Figure 14.22. Two query images, 36_238.JPG and 49_238.JPG, are used here to illustrate the features contribution on the retrieval performance. The first image group consists of three very similar images taken with slight camera movement, and the second group contains two similar images with slightly different background. The results for the experiment are tabulated in Table 14.4.

We observed that energy histograms based exclusively on the DC coefficient (F1) might only perform well on the retrieval of images with high similarity in colors. The results of F2A and F3A suggested that histograms of low-frequency AC coefficients, which carry the texture and edge information, are contributory to the similarity measure. Thus, the combination of DC and numerous low-level AC coefficients (F2B, F3B, F4B) yielded better results on both
of the lists. On comparison of block size effect, further examination using other queries in our experiment shows that in general F2B and F3B are much preferable to F4B. This may be due to the fact that as the feature block grows larger, heavily quantized coefficients are also taken into consideration; hence erroneous results may be generated.

We also noticed that the retrieval performance for features F2B and F3B is relatively unaffected by translation of small objects in globally uniform images. A retrieval sample is shown in Figure 14.23.

As for retrieval of lossless DCT transformed images, a query was chosen and transformed using the JPEGtran utility provided in [13]. The transformed images were then added into the image database prior to the retrieval test. We observed that all features are able to recognize the transformed images. An image group used in the experiment is shown in Figure 14.24.

On the issue of computational cost, we noticed that the complexity could be significantly reduced through attentive selection of features in the compressed domain. A 2 × 2 block feature (F2B) may reduce the feature complexity by a factor of 1/16 with respect to the overall color histogram method [17].

---

**Table 14.4** Retrieval Hit for Queries

<table>
<thead>
<tr>
<th>Q: 36_238</th>
<th>F1</th>
<th>F2A</th>
<th>F2B</th>
<th>F3A</th>
<th>F3B</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank 1 (best)</td>
<td>37_238</td>
<td>37_238</td>
<td>37_238</td>
<td>37_238</td>
<td>37_238</td>
<td>37_238</td>
</tr>
<tr>
<td>Rank 2</td>
<td>35_238</td>
<td>X</td>
<td>35_238</td>
<td>35_238</td>
<td>35_238</td>
<td>35_238</td>
</tr>
<tr>
<td>Rank 3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rank 4</td>
<td>X</td>
<td>35_238</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q: 49_238</th>
<th>F1</th>
<th>F2A</th>
<th>F2B</th>
<th>F3A</th>
<th>F3B</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank 1 (best)</td>
<td>X</td>
<td>X</td>
<td>48_238</td>
<td>X</td>
<td>48_238</td>
<td>48_238</td>
</tr>
<tr>
<td>Rank 2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>48_238</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rank 3</td>
<td>X</td>
<td>48_238</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rank 4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rank 5</td>
<td>48_238</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Video Parsing

The experiment on video parsing [21] was based on several MPEG streams selected for several testing purposes. Opera House is a simple stream showing scenic views around the Sydney Opera House. It is used to examine the feature performance on parsing of video containing only sharp transitions. Aquarium and Downtown were obtained from the Internet [37]. Aquarium is a relatively simple video sequence, whereas Downtown is a fast-changing video sequence with rapid camera movements and dissolve transitions. Blink was produced in a controlled environment in order to investigate the effects of a flashing and blinking light source. A short description of the video streams is provided in Table 14.5, and several sample frames from Opera House are shown in Figure 14.25.

For relatively simple video sequences such as the Opera House, the use of DC coefficients alone is sufficient to detect abrupt scenery changes. However, for more complex dissolve transitions and other effects, the use of the DC coefficient alone can lead to false detection. In general, features based on F1F and F2F yield better results than the F1F alone. False detection...
The block diagrams for detecting sharp and dissolve transitions are, respectively, shown in Figures 14.22 and 14.26. Since a sharp transition occurs between two frames, the distance of the energy histogram features is usually very large for all YUV components. Therefore, a sharp transition can be detected by using any one of the YUV components. Otherwise, the combined distance of the three component distances can be used.

For a dissolve transition, the frames of the leading shot will fade out and gradually lose their color information, whereas the frames of the second shot will gradually increase in color. This characteristic can be used to assist video parsing. Thus, the features based on U and V components alone can be used to detect dissolve transitions. However, this results in a rather noisy distance graph.

Because the frames are changing gradually, the distances between energy histograms produced from DCT coefficients are small. A median filter is applied to enhance the characteristic. The filtered distances may show large pulses at nontransitional frames; however, this rarely happens at the same time for all three YUV components. Various methods such as averaging were used to combine the filtered distance values of the three components. An effective method found was to take the product of the three component distances. In fact, using the distance graph of the Y component to attenuate the unwanted noise yielded promising results.
The results obtained for Aquarium are shown in Figure 14.28. The graphs indicate a sharp increase at points where sharp transitions occur. The lower-right graph shows the results of combining all the distances after applying a median filter; the crest near frame 600 is caused by ripples on the water surface in the video. The ripples introduce perturbation into UV components. Note that in the normalized total distance plots, the effects of sharp transitions have been removed by ignoring the distance at sharp transitions in the process of combining the three components.

The normalized total distance for Downtown is given in Figure 14.29. Downtown was parsed using the same configuration as in the experiment for Figure 14.23. An interesting fact is that for fast camera movement and the zooming in and zooming out effects, the features display characteristics similar to dissolve transitions. Zooming out at frame 105 and rapid
camera movement at frame 165 in Downtown resulted in large peaks in the graph. This can be seen in Figure 14.7 near frames 110 and 170.

![Graph showing large peaks in the graph at frame 165 in Downtown.](image)

FIGURE 14.29
Normalized distances for Downtown using F2B.

The effects of the change of illumination were investigated using the Blink stream. The results obtained are shown in Figure 14.30. The Blink stream contains many blinks from a light bulb followed by a flash. It is observed that the changes of illumination caused by the light result in great perturbation on the energy histogram of the UV components. Frames containing the blinking effect were correctly parsed using the features, whereas the flash effect was parsed as a false transition. An illumination change of a video shot is illustrated in Figure 14.31.

![Graphs showing component and normalized distances for Blink using F2B.](image)

FIGURE 14.30
Component and normalized distances for Blink using F2B.
14.4 Conclusions

We have presented the use of energy histograms of the LF-DCT coefficient features for the retrieval of JPEG images and the parsing of MPEG videos. We have shown how the features can be used to perform retrieval on medium-size databases as well as to parse relatively complex shot transitions. We have also shown that by introducing the transpose symmetry, a vigorous feature set can be built to accommodate the DCT domain-based lossless transforms.

References


[9] Multimedia Content Description Interface (MPEG-7),
http://drogo.cselt.stet.it/mpeg/.


[13] T. Lane, Independent JPEG Group (IJG) JPEG Library,

[14] A.C. Hung, Stanford University Portable Video Research Group's (PRVG) PRVG-JPEG Software Codec,
ftp://havefun.stanford.edu:/pub/jpeg/JPEGv1.2.1.tar.Z.

[15] Cornell lossless JPEG implementation,


[38] MPEG, ISO/IEC JTC1/SC29/WG11 N2966, MPEG-7 Generic AV Description Scheme (V0.7), Melbourne, Oct. 1999.


