8.1.6 Image Compression Models

As Fig. 8.5 shows, an image compression system is composed of two distinct functional components: an *encoder* and a *decoder*. The encoder performs compression, and the decoder performs the complementary operation of decompression. Both operations can be performed in software, as is the case in Web browsers and many commercial image editing programs, or in a combination of hardware and firmware, as in commercial DVD players. A *codec* is a device or program that is capable of both encoding and decoding.

ion
d to
x, y) and

Input image $f(x, \dots)$ is fed into the encoder, which creates a compressed representation of the input. This representation is stored for later use, or transmitted for storage and use at a remote location. When the compressed representation is presented to its complementary decoder, a reconstructed output image $\hat{f}(x, \dots)$ is generated. In still-image applications, the encoded input and decoder output are $f(x, y)$ and $\hat{f}(x, y)$, respectively; in video applications, they

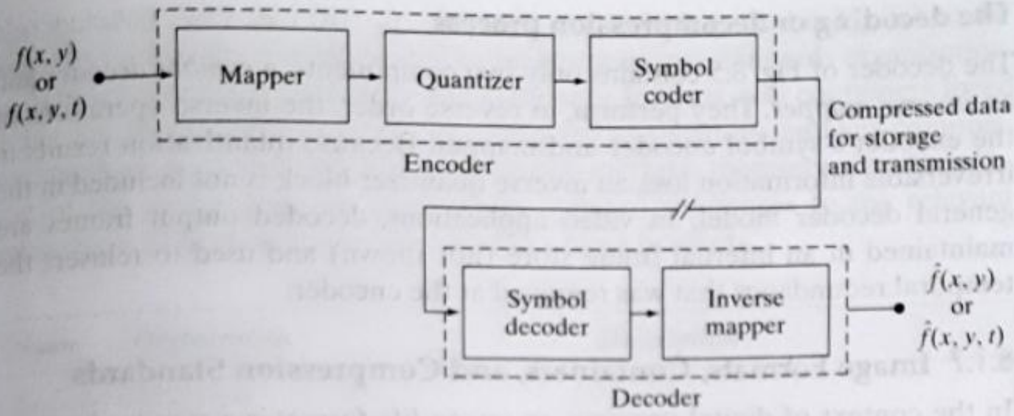

FIGURE 8.5 Functional block diagram of a general image compression system.

are $f(x, y, t)$ and $\hat{f}(x, y, t)$, where discrete parameter t specifies time. In general, $\hat{f}(x, \ldots)$ may or may not be an exact replica of $f(x, \ldots)$. If it is, the compression system is called *error free, lossless,* or *information preserving*. If not, the reconstructed output image is distorted and the compression system is referred to as *lossy*.

The encoding or compression process

The encoder of Fig. 8.5 is designed to remove the redundancies described in Sections 8.1.1–8.1.3 through a series of three independent operations. In the first stage of the encoding process, a *mapper* transforms $f(x, \ldots)$ into a (usually nonvisual) format designed to reduce spatial and temporal redundancy. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image. Run-length coding (see Sections 8.1.2 and 8.2.5) is an example of a mapping that normally yields compression in the first step of the encoding process. The mapping of an image into a set of less correlated transform coefficients (see Section 8.2.8) is an example of the opposite case (the coefficients must be further processed to achieve compression). In video applications, the mapper uses previous (and in some cases future) video frames to facilitate the removal of temporal redundancy.

The *quantizer* in Fig. 8.5 reduces the accuracy of the mapper's output in accordance with a pre-established fidelity criterion. The goal is to keep irrelevant information out of the compressed representation. As noted in Section 8.1.3, this operation is irreversible. It must be omitted when error-free compression is desired. In video applications, the *bit rate* of the encoded output is often measured (in bits/second) and used to adjust the operation of the quantizer so that a predetermined average output rate is maintained. Thus, the visual quality of the output can vary from frame to frame as a function of image content.

In the third and final stage of the encoding process, the *symbol coder* of Fig. 8.5 generates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code. In many cases, a variable-length code is used. The shortest code words are assigned to the most frequently occurring quantizer output values—thus minimizing coding redundancy. This operation is reversible. Upon its completion, the input image has been processed for the removal of each of the three redundancies described in Sections 8.1.1 to 8.1.3.

The decoding or decompression process

The decoder of Fig. 8.5 contains only two components: a *symbol decoder* and an *inverse mapper*. They perform, in reverse order, the inverse operations of the encoder's symbol encoder and mapper. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general decoder model. In video applications, decoded output frames are maintained in an internal frame store (not shown) and used to reinsert the temporal redundancy that was removed at the encoder.