H.261 Codec 18

The H.261 compression-decompression scheme takes its name from the title of the recommendation in which the H.261 codec is specified: Recommendation H.261 published by the International Telegraph and Telephone Consultative Committee (CCITT). This recommendation defines a video encoder that is intended to be used to compress video data that will be sent over Integrated Services Digital Network (ISDN) lines.

The H.261 codec is intended primarily for use in video telephony and videoconferencing applications. Video telephony, in which generally a picture of the speaker's face against a stationary background is transmitted, is possible when only one or two ISDN channels (each capable of carrying 64Kb/s of information) are available. If more channels are available, more complex images can be sent.

The remainder of this chapter is divided into four sections. The first section provides an overview of how an H.261 codec works. The second explains briefly how to create an H.261 CIS. The third discusses CIS attributes that apply specifically to a CIS associated with an XIL H.261 compressor or decompressor (as opposed to the general CIS attributes covered in the section "General CIS Attributes" on page 257). And the fourth section introduces the subject of accelerating the playback of H.261 bitstreams. For further information on this subject, see Chapter 21, "Acceleration in XIL Programs.".

18

Note – This chapter discusses both H.261 compression and decompression. However, the current release of the XIL library includes only an H.261 decompressor. The compressor interface is defined for third parties who want to implement XIL H.261 compressors.

How an H.261 Codec Works

This section presents an overview of how an H.261 codec works. It discusses

- The format of the images that can be used as input to the encoder
- The basic encoding scheme
- Methods of controlling the size of the bitstream produced by the encoder
- How the codec supports multipoint conferencing

Source Images

The images supplied as input to an H.261 compressor must meet both color space and size (width and height) requirements. In terms of color space, the images must be YC_bC_r images that conform to the standard set forth in CCIR Recommendation 601. In terms of size, the images must adhere to either the Common Interchange Format (CIF) or the Quarter-CIF (QCIF) format. Table 18-1 below indicates the widths and heights defined by these formats.

Table 18-1 Sizes of CIF- and QCIF-Format Images

	Width	**Height**
CIF images	352	288
QCIF images	176	144

All H.261 encoders must be able to compress QCIF images. The ability to compress CIF images is optional.

Given an image of the appropriate format, the H.261 compressor subsamples the chrominance values so that there is one C_b value and one C_r value for each two-by-two block of luminance values. It then processes the image in segments called macroblocks. Each macroblock consists of a 16-by-16 block of luminance values and the chrominance values associated with those luminance values. See Figure 18-1 below.

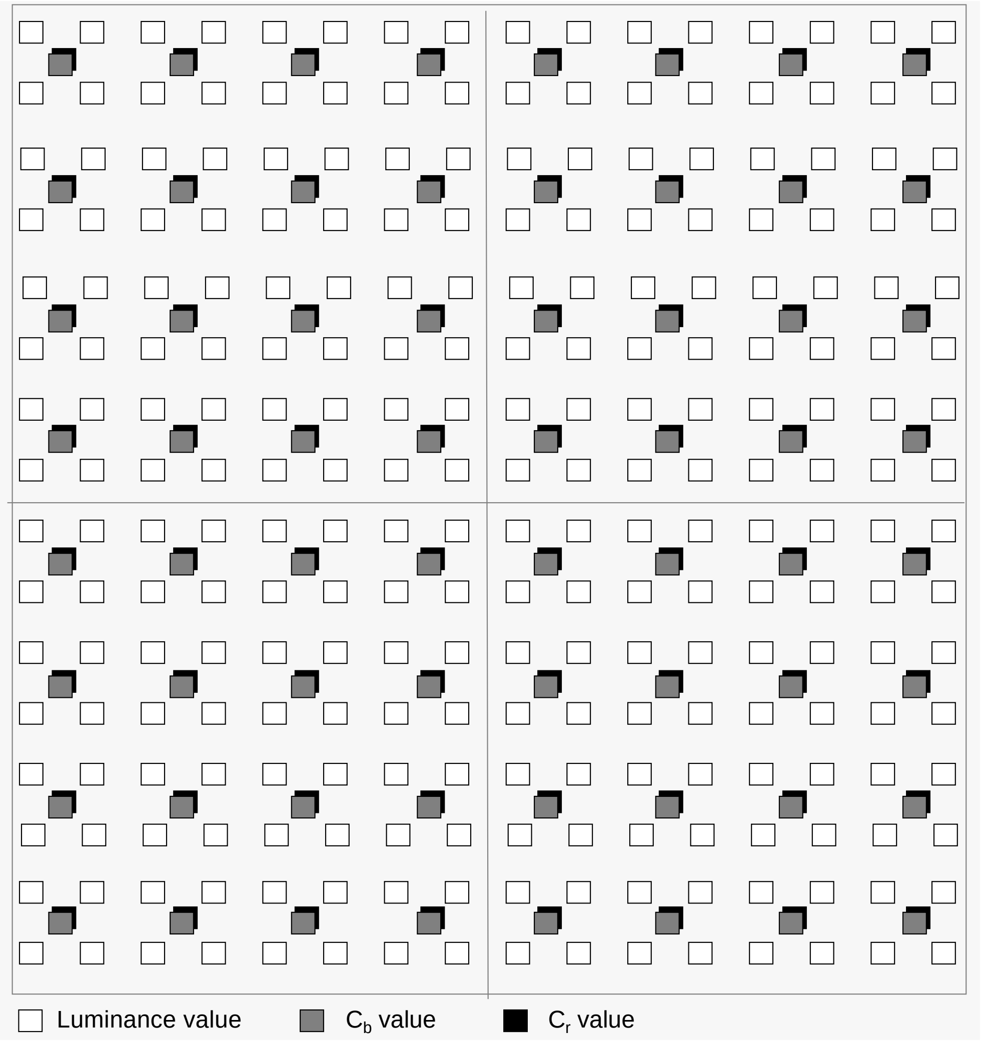

Figure 18-1 Macroblock

As you’ll see shortly, the encoder performs some operations on 8-by-8 blocks of values. Each macroblock contains six blocks: four blocks of luminance values, one block of C_b values, and one block of C_r values.

Basic Encoding Scheme

The flow chart in Figure 18-2 illustrates the procedure the H.261 encoder uses to encode a macroblock.

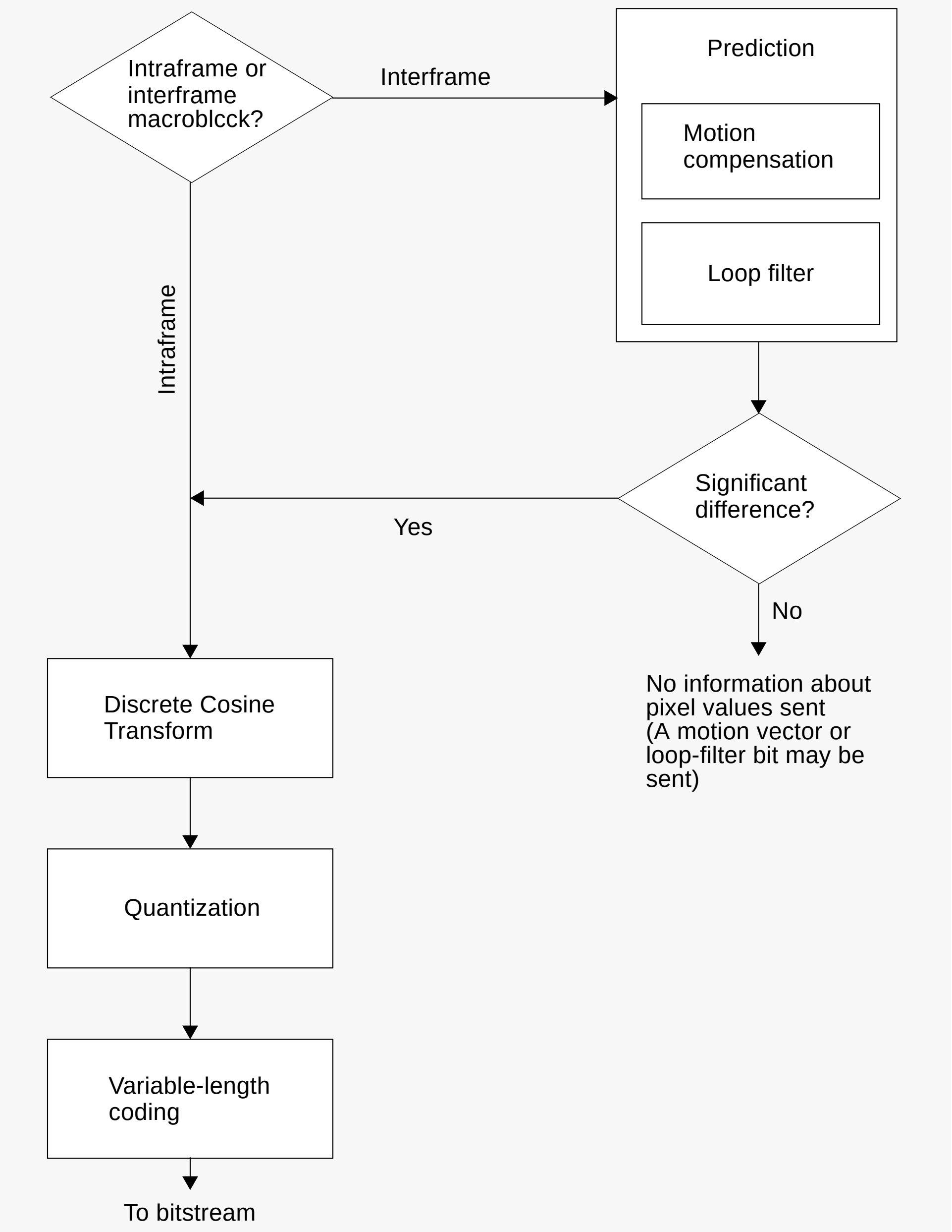

Figure 18-2 Flow Diagram for H.261 Encoding

Intraframe Versus Interframe Encoding

For each macroblock that it encodes, the H.261 encoder can perform intraframe or interframe compression. In intraframe mode, the compressor encodes the actual YC_bC_r values in the macroblock. In interframe mode, the compressor

1. Looks at the YC_bC_r values in the macroblock it is encoding
2. Calculates the difference between the predicted values for the macroblock and the actual values in the macroblock. The predicted values are taken from the most recently compressed image, which is stored in a history buffer.
3. Encodes the difference values if they are significant

In general, the H.261 compressor relies very heavily on interframe encoding because this type of encoding leads to greater rates of compression than intraframe compression. However, Recommendation H.261 requires that the encoder intraframe encode each macroblock at least once every 132 frames. This requirement ensures that if you join a videoconference in progress or your videoconference is disrupted by data transmission problems, all macroblocks will be updated properly within a few seconds.

Prediction

As noted in the last section, in interframe-encoding mode, the encoder calculates the difference between YC_bC_r values in the macroblock it is encoding currently and the values in the corresponding macroblock in the preceding picture. Before performing this calculation, the encoder may perform either a motion-compensation operation or a motion-compensation operation followed by a loop-filter operation. Both of these operations are optional.

When the encoder performs motion compensation, it compares the YC_bC_r values in the current macroblock not only with those in the spatially corresponding macroblock in the preceding picture, but also with the values in macroblocks that neighbor the spatially corresponding macroblock in the preceding picture. See Figure 18-3.

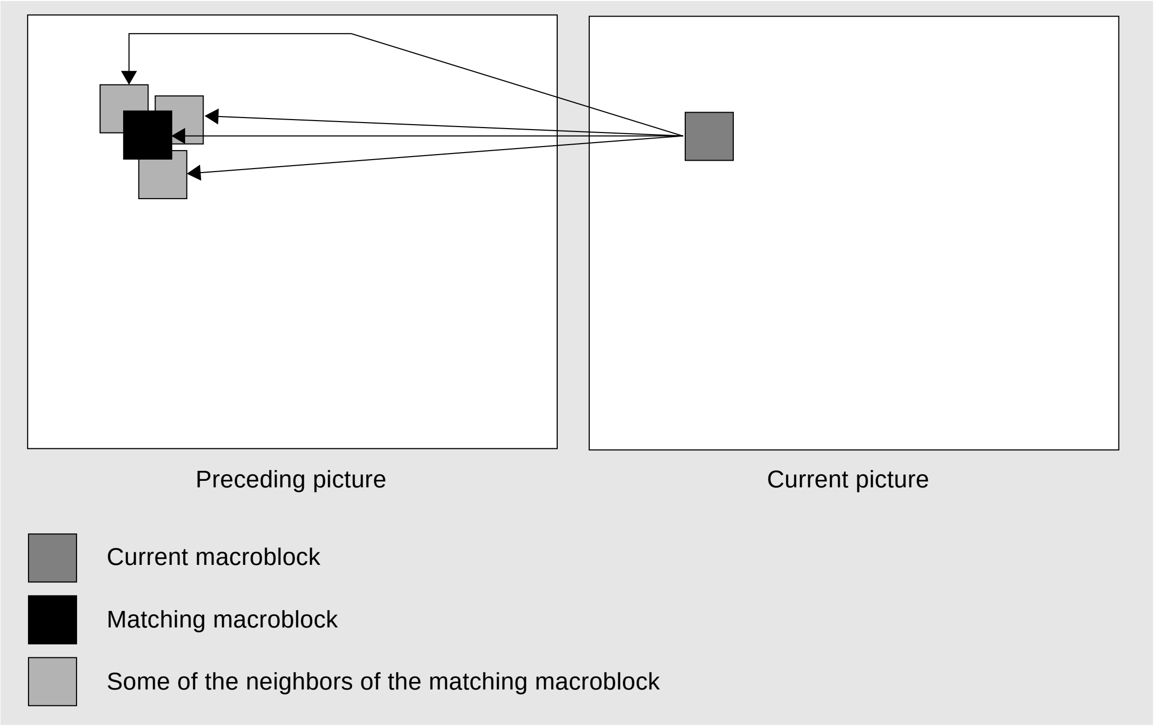

Figure 18-3 Motion Compensation in H.261

The neighboring macroblocks being examined can be offset from the matching macroblock by a maximum of ±15 pixels in both the *x* and *y* directions. The macroblock in the history image that best matches the macroblock in the current image is used in calculating the difference values to encode.

If the macroblock in the history image used to calculate difference values is not the matching macroblock, the encoder must record the number of pixels by which the former is offset from the latter. These *x* and *y* offsets are written to a motion vector, which is later variable-length coded and written to the H.261 bitstream along with any encoded difference values.

The second operation that the encoder may perform before calculating the difference between macroblocks is a filter of the macroblock of interest in the history image. This filtering operation is designed to remove high-frequency information from the macroblock. Generally, this filtering leads to smaller differences between the macroblocks and, thus, to a more compact bitstream.

Encoding YC_bC_r or Difference Values

Whether the encoder is encoding the actual values in a macroblock from the current picture or difference values calculated during the prediction step, it uses the procedure depicted in Figure 18-4.

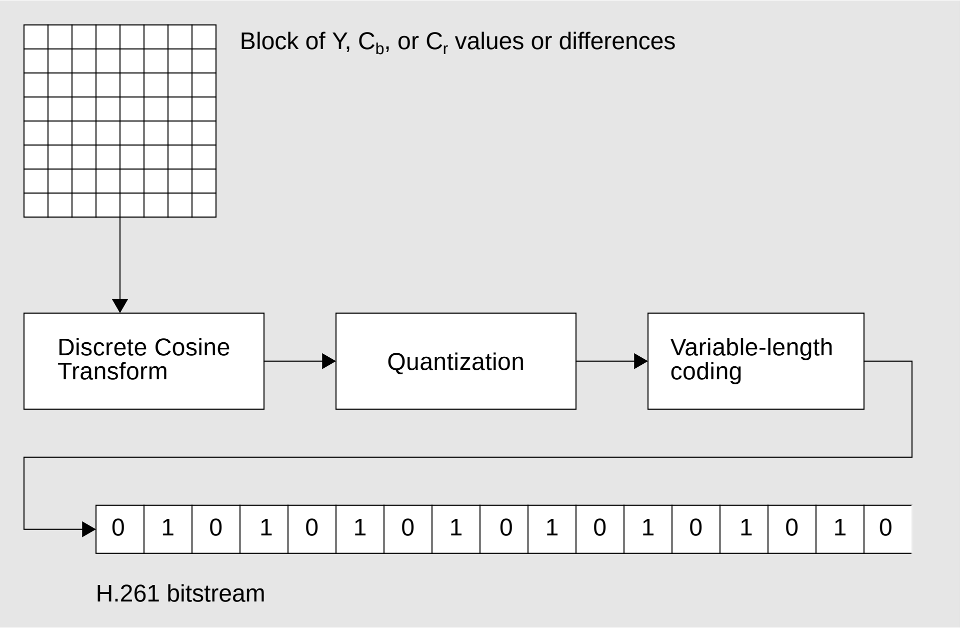

Figure 18-4 Encoding of YC_bC_r or Difference Values in H.261

Each of the six blocks in a macroblock (four blocks of luminance values and two blocks of color values) is encoded separately. The values for each block are

1. Transformed from the spatial to the transform domain using a Discrete Cosine Transform

2. Linearly quantized

3. Encoded with variable length codes or, for less frequently occurring values, with 20-bit codes

This encoding scheme is very similar to the one used in the JPEG still-image compression standard. For a more detailed discussion of the steps involved in this type of encoding, see the section "How the JPEG Baseline Sequential Codec Works" on page 306.

Bit-Rate Control

As mentioned earlier, the H.261 codec is intended primarily for use in videophone and videoconferencing applications. Because these applications need to send data at a constant rate over a network, the encoder must use a constant number of bits to encode, say, a second's worth of video. The encoder can achieve this constant bit rate using any combination of the following techniques.

- Altering the criterion that determines whether a macroblock that is to be interframe encoded actually needs to be encoded. The macroblock needs to be encoded only if its luminance values differ from those in the corresponding macroblock in the preceding picture by a certain amount. By increasing this amount, the encoder decreases the number of macroblocks it must compress.
- Changing the values in the quantizer. To produce a lower bit rate, the encoder can increase the size of the values in the quantizer. This strategy results in quantized coefficients with relatively low values, which can be encoded with relatively short code words.
- Using the loop filter operation described in the section "Prediction" on page 342.

Provisions for Multipoint Conferencing

The CCITT's specification of the H.261 codec includes several features designed to facilitate multipoint conferencing. In a multipoint conference, the receiver may elect to switch between two or more sources of video. These features include:

- A freeze-picture request. This request is an external signal that causes the decoder to stop updating the currently displayed picture. The picture remains frozen until the decoder sees a freeze-picture-release flag in the bitstream or until a timeout period of six seconds or more has elapsed.
- A fast-update request. This request is an external signal that causes the encoder to compress the next picture using intraframe encoding exclusively. The encoder must compress this frame without overflowing its output buffer.
- A freeze-picture release. When an encoder receives a fast-update request, it sets a bit in the header of the next picture it encodes. This bit tells a decoder that has frozen its display to resume displaying pictures in the normal way.

Creating an H.261 CIS

Before you can use the H.261 decompressor to decompress an H.261 bitstream, you must create an H.261 CIS (and write an H.261 bitstream to the CIS). You create this CIS by passing the decompressor name `H261` to the function `xil_cis_create()`. See the code fragment below.

```
XilCis cis;
XilSystemState state;

cis = xil_cis_create(state, “H261”);
```