Chapter 1

Video Compression

Most recent research on video processing has been on digital video, i.e. video which is discrete in time, space, and amplitude. By digitizing a video signal, it is possible to process and store the information on computers and computer peripherals. In the first part of this thesis, we consider one aspect of video processing, known as video compression. The purpose of video compression is to reduce the number of bits required to encode a digitized video sequence. Nearly all video compression schemes are lossy, meaning they result in some loss of quality upon reconstructing the coded bitstream. As expected, there is a direct correlation between the reconstructed video quality and the coded bit rate. Thus we must choose a specific target bit rate or quality to encode a video sequence.

The primary disadvantage of encoding at a fixed target bit rate or quality is that we limit the decoder to use the same parameters. A decoder that does not possess the computational power to process the bitstream at the coded bit rate will be unable to reconstruct any video. On the other hand, a decoder that exceeds the requirements will be wasting computational capacity. One method of catering to different decoders is known as simulcasting. Simulcasting is the process of encoding, storing, and broadcasting the same video at a variety of different bit rates or qualities. However, this wastes resources at every level, since the information in the lower-quality bitstreams are typically subsets of the information in the highest-quality stream.

Another option is to use scalable video compression algorithms which are becoming increasingly more important in a large number of applications. By scalable compression algorithms, we mean ones that are capable of producing *one* compressed full rate video bit stream whose various subsets correspond to different resolutions/rates of the same video sequence. Potential applications of these algorithms include interconnecting heterogeneous networks with various bandwidth characteristics, simulcast applications, multimedia applications, video storage and servers, and video on-demand services. As an example, in heterogeneous networks of the future in which high-speed fiber optics networks with gigabits per second capability are connected to lower bandwidth networks such as ISDN and POTS, it is highly desirable to "filter" a compressed video bit stream originating from a source to various destinations with different bandwidth connection capabilities. Another example is in multimedia systems in which it is highly desirable to provide each user with a video bit rate proportional to the window size s/he chooses to open. In this scenario, use of scalable compression algorithms has several advantages. First, the user's decoder does not necessarily need to decode a full resolution bit stream before choosing the desired bit rate/resolution; second, by providing a subset of the full resolution video bit stream, the server can serve a larger number of clients; finally, scalable compression of video sequences is more storage efficient in that it eliminates the need for storing multiple copies of the video at different resolutions in the server.

Another use of scalability is during periods of resource overallocation in which the video can undergo graceful degradation, i.e. a partial loss of the bitstream results in only partial loss in quality. A bitstream with many levels of quality at different bit rates is considered to be highly scalable, or scalable with a fine degree of granularity. On the other hand, a bitstream is considered nonscalable if any loss of bits causes a complete loss of video at the receiver.

The first part of this thesis addresses video compression, with an emphasis on scalability. In this chapter, we begin by reviewing two common methods of video compression: MPEG in Section 1.1 and subband video coding in Section 1.2. Since the latter results in a much more scalable bitstream, we present our own work on subband coding in Chapters 2 through 5. In Chapter 2 we examine the effects of using subband filters that respond to object movements. We expand the idea by designing subband systems to respond to a larger range of motion in Chapter 3. Chapter 4 describes a more efficient approach to incorporating motion estimation within a subband framework and also introduces a multi-rate quantization strategy that results in scalable video coding. Finally, in Chapter 5 we refine and optimize our multi-rate quantization strategy to achieve a tradeoff between reconstruction quality and speed.

1.1 MPEG

The Motion Pictures Experts Group (MPEG), a subcommittee of the International Standards Organization, is responsible for the following standards [10]. MPEG-1 is the coding standard for progressively-scanned video, and the performance is optimized for Source Input Format (SIF) resolution images, 352 columns by 240 rows, at 30 frames per second for a bit rate of 1.2 to 1.5 megabits per second (Mb/s). The MPEG-1 syntax allows sampling dimensions as high as $4095 \times 4095 \times 60$ frames per second, although this is not used in practice.

MPEG-2 is generally used for higher resolution images and frame rate. The main difference between MPEG-1 and MPEG-2 is that the latter is capable of coding both progressive and interlaced video signals. In addition, minor changes were made in the prediction and quantization algorithms to improve performance; for more details, see [10]. Finally, MPEG-2 specifies a syntax for four modes of scalability. However, we show in Section 1.1.1 that most forms of scalability are inherently incompatible with motion compensated prediction-based coding schemes such as MPEG-1 and MPEG-2.

As a historical note, MPEG-3 was to be designed as a standard for HDTV and optimized for a bit rate of 20 to 40 Mb/s. However, it was found that MPEG-1 and MPEG-2 performed well for this application, so research efforts into MPEG-3 were abandoned. MPEG-4 is currently under development, with a target resolution of up to 176 columns and 144 rows at 10 frames per seconds. The target bit rate is from 4,800 to 64,000 bits per second.

1.1.1 MPEG-1 Encoder

We show an operational block diagram of an MPEG-1 encoder in Figure 1.1. MPEG-1 classifies frames into three types according to the method in which they will be coded: intraframe (I), predicted (P), and bi-directional predicted (B). The pattern of classification is often periodic and set a priori; however, advanced coders may change the pattern adaptively according to the content of the video sequence. The periodic non-adaptive pattern in the *Star Wars* sequence [11] is: IBBPBBPBBPBBPBB. In this case, the first frame is an I-frame, and so are all the ones $N \times 12$ frames afterwards, where N is an integer. As shown in the diagram, the I-frames pass directly to being partitioned into 8×8 blocks, whereas the P-frames are predicted from the previous I or P-frame, and the

B-frames are predicted from both the previous and the future I or P-frame. Thus B-frames cannot be encoded until after the following I or P-frame.

The frames are then partitioned into 8×8 blocks and then subject to a separable two-dimensional (2-D) Discrete Cosine Transform (DCT). The DC coefficients of each I-frame block are subtracted from the DC coefficient of the previous 8×8 block, a process known as one-dimensional (1-D) Differential Pulse Code Modulation (DPCM) [15]. Note the DPCM is not shown in the diagram for simplicity. Other coefficients pass directly to the quantizer. The output of the quantizer is then run-length coded and Huffman coded to produce the output bitstream, while the I and P-frames are also fed back to a frame buffer for the P and B-frame predictions.

The predictions are done using the Block Matching Algorithm (BMA) for motion compensation. For P-frames, each 8×8 block is matched against a search area in the previous I or P-frame, and the resulting displacement is known as a motion vector and Huffman coded to become a part of the output bitstream. For B-frames, each 8×8 block is searched against the past and future frames for the best matching blocks; the encoder may then choose between using either one, the average of both, or none at all. For both P and B-frames, the motion vectors are also Huffman coded with a precomputed Huffman table.

The final bitstream, consisting of the coded motion vectors and DCT coefficients, is sent to the receiver and also used for rate control. A typical rate control mechanism is the leaky bucket [23], which monitors a buffer. As the buffer approaches the maximum capacity, the quantization intervals are increased, resulting in a decrease in the overall bit rate. In this manner, the rate control can maintain a constant bit rate (CBR) stream to be sent to the receiver by lowering the video quality during complex or high-motion scenes that would otherwise result in an increased bit rate.

Because MPEG-1 is a predictive coder, it has difficulty generating a highly scalable bitstream as follows. To be scalable, a coder must have a low-quality layer that can be decoded independently of the high-quality layer. To be highly scalable, this condition must hold true for many layers – there must be many sets of layers that can be independently decoded. In a predictive coder, the encoder assumes that it accurately tracks the state of the decoder in the frame memory, as given in Figure 1.1. However, if there are many decoders each extracting a different subset of the full coded bitstream, then it is impossible for the encoder to know the state of any given decoder, since the encoder does not know how the quality of video that is being reconstructed at each decoder.

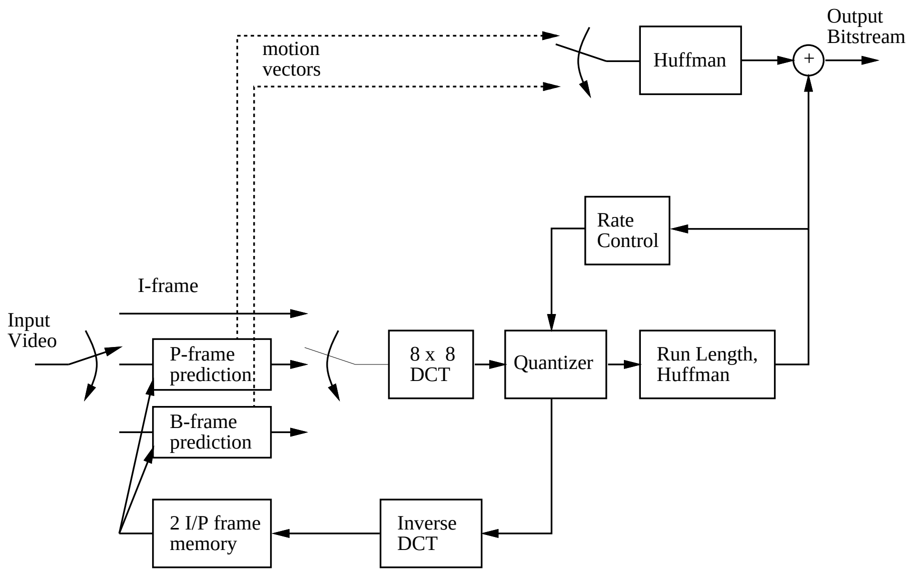

Figure 1.1: MPEG encoder block diagram

There are three strategies that a predictive encoder can use to overcome this limitation and implement scalability. First, the encoder can cater to the lowest bit rate video stream by using that stream in the predictive feedback loop. This is the strategy outlined in the MPEG-2 syntax. However, for a highly scalable bitstream with a large difference between the lowest and highest rates, this method becomes inefficient for the higher rates. This results from the large energy in the prediction residual. Second, the encoder can cater to the highest bit rate stream by using the full stream in the predictive feedback loop. That strategy, however, results in prediction drift at the lower quality decoders, since the encoder is not modeling their states accurately. Finally, the encoder can track the state of each quality of decoder separately to generate a bitstream catering to all of them. The implementation cost of this strategy becomes prohibitive for more than a few rates; thus this strategy cannot be applied to a highly scalable bitstream.

1.2 Subband Coding

Among existing video coding techniques, subband coding has received a great deal of attention in recent years [32]. The idea behind subband coding is to divide the input signal into different frequency components. It is well-known that most of the energy in real video sequences resides in the lower spatiotemporal frequencies; thus three-dimensional (3-D) video subband coding generally contains lowpass and highpass filters in the temporal and spatial dimensions. The resulting energy separation can be shown to whiten the input spectrum in each band and provide a coding gain after quantization [15].

We show an operational block diagram of a 3-D subband encoder using 1-D filters in Figure 1.2. In the diagram, we show one level of filtering for each dimension $t, y, and x$. Highpass filters are denoted by H, lowpass filters are denoted by L, and the subscripts indicate the dimension of filtering. Each filtering operation is followed by a downsampling by a factor of two in the same dimension, and the entire process of filtering and downsampling is commonly known as the analysis stage of subband filtering. After the filtering and downsampling, the signal values are quantized and Huffman coded, together represented by the boxes marked "Q," to generate the output bitstream. To reconstruct the signal, the output subband signals are passed through a synthesis stage consisting of an architecture of upsamplers and subband filters that is symmetric with the analysis stage. According to the Nyquist sampling theorem [22], the subband filters must be perfect bandpass filters to prevent aliasing error resulting from the downsampling. It has been shown [28, 30], however, that there exist Perfect Reconstruction (PR) filters such that aliasing errors from multiple bands cancel out so that the reconstructed signal matches the original signal exactly. In practice, Near-Perfect Reconstruction (NPR) filters are often used, as the aliasing error is much smaller than the quantization error in many implementations.

After the subband decomposition, the $L_T L_Y L_X$ frequency band contains the majority of the video signal energy. 1-D DPCM [15] is used to reduce this energy before quantization, as shown in Figure 1.2. In real implementations [16, 25], this uneven distribution of energy is exploited by adding extra levels of subband filtering and downsampling to the $L_T L_Y L_X$ frequency band. The number of levels in each dimension is chosen by considering the delay, complexity, and cost of implementation.

Subband coding has many advantages over MPEG-1. First, subband coding does not rely on a block structure; thus it exploits the correlation between blocks and results in fewer visually distracting blocky artifacts. Second, it is easier to implement scalability in a subband coder, since it is not a predictive coding scheme. As explained in Section 1.1, predictive coding is inherently incompatible with scalability. Third, subband coding exploits the human visual characteristics in an explicit manner, as each frequency band can be quantized according to human perceptual sensitivity to that band. Finally, subband coding

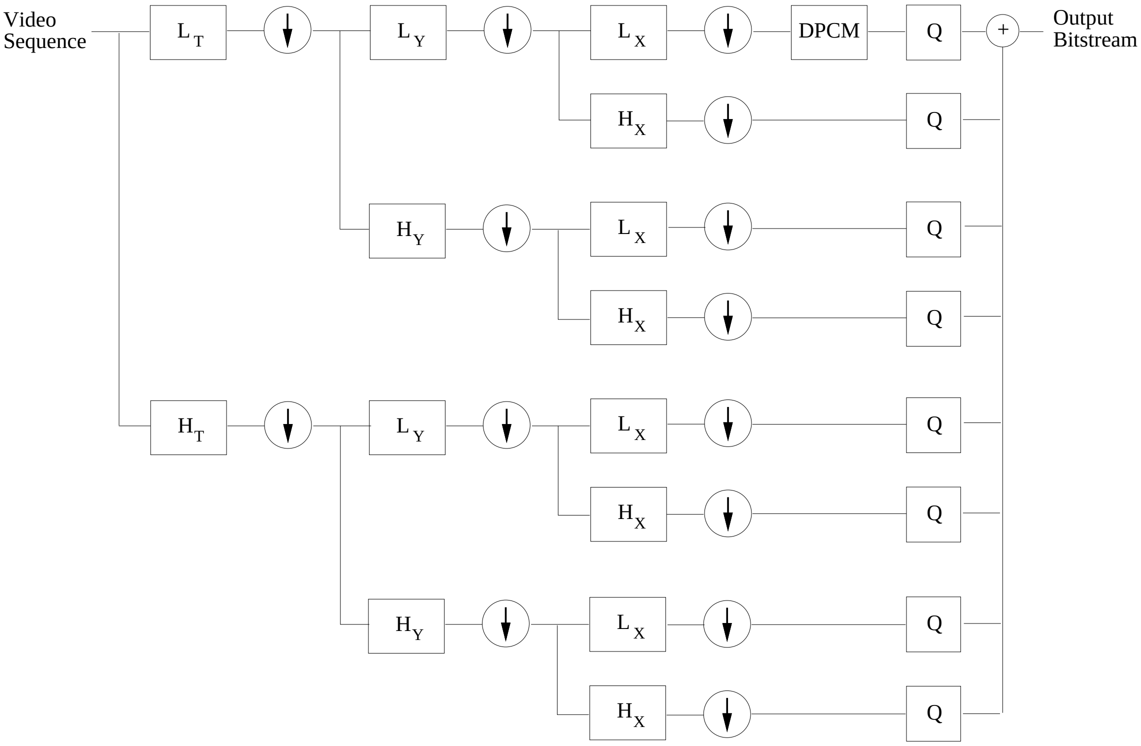

Figure 1.2: 3-D Subband encoder block diagram

is easy to implement in a parallel architecture to distribute the computation across multiple processors and increase speed.

The main disadvantage to subband coding for video is that motion compensation is difficult to implement. As a result, it is difficult to obtain the compression efficiency of a motion compensation predictive coding strategy such as MPEG-1. This motivates our work in Chapters 2 through 4, to try to exploit object motion within the context of subband filtering.