CHAPTER 1

INTRODUCTION

Digital video coding technology has matured over the last decade. Over the past two decades, the world has witnessed tremendous advancements in the area of multimedia based applications like video on demand, video conferencing, medical imaging and multimedia image/video database services. The key obstacle in such applications for scientific and research community has been the efficient representation of vast amount of data used in these applications for storage and transmission purposes while retaining quality. A digitized version of a single color picture at TV resolution contains data of the order of millions of bytes. The reduction in size of image/video data is possible due to the presence of various redundancies like spatial redundancy, coding redundancy, phychovisual redundancy, and temporal redundancy (Gonzalez and Woods (2002)). Commercial products have been developed that are targeted for various applications. With the increased commercialization of multimedia based applications, the need for international video compression standards arose.

To meet this need, motion picture expert group (MPEG) was formed in 1988 as a joint venture of International Organization of Standardization/International Electrotechnical Commission (ISO/IEC) to develop standards for representation of video when used for storage and retrieval on digital storage media. This group has produced various video coding standards like MPEG-1, MPEG-2, MPEG-4 and MPEG-7 depending on applications. MPEG-1, issued in 1992, supports bit rate unto 1.5 Mbits/sec and mainly designed for digital video on CD-ROM with spatial resolution up to 288x360 pixels. In 1994, MPEG-2 was issued to extend bit rate targeted between 2 to 10 Mbits/sec with high definition TV resolution to benefit applications like digital TV distribution, networked database services via ATM, satellite and terrestrial digital broadcasting distribution. Anticipating the rapid convergence of TV, computer and telecommunication, the MPEG officially initiated MPEG-4 standard to standardize algorithms and tools for coding and flexible representation of data to meet the challenges of future multimedia
applications (Chiariglion (1997); Sakora (1997a)). Officially introduced in 1998, it addresses universal accessibility and robustness in error prone environments, improves coding efficiency for low bit rate video applications (of the order of 5-64 Kbits/sec) like mobile or public switched telephone networks (PSTN), supports target rate up to 4Mbits/sec for TV/film applications. MPEG-7, formally known as Multimedia Content Description Interface, has officially been introduced in 2001 which includes standardized tools for detailed and structured representation of multimedia data (Sayood (2000))

1.1 Background of Video Compression

Video sequences consist of intrapixel and interpixel redundancies within and between frames. A good video encoder has ultimate goal of minimizing these redundancies and performance of any video compression technique depends on the amount of redundancy contained in the input data as well as on the actual compression technique used for coding. Depending on the application’s requirements there are two types of video compression - lossless and lossy. The objective of lossless compression is to reduce size of data while preserving the quality of input video. In this case, decoded video quality is exactly same to the original video quality. In lossy compression, aim is to meet the desired compression for storage and transmission at the cost of compromising quality of original video. All MPEG standards use lossy coding technique. The ultimate aim of lossy coding technique is to optimize video quality for a targeted compression (measured in terms of average bits/pixel or sometimes compression ratio) subject to some optimization criterion.

Video sequences have pixel correlation in spatial as well as temporal directions and MPEG coding techniques rely on these correlations. Thus, it is assumed that intensity of a pixel can be predicted from its spatially adjacent pixels (intrapixel redundancy) and temporally adjacent pixels (interpixel redundancy). Intraframe and interframe coding techniques are used to remove these redundancies respectively. Temporal correlation between adjacent frames is almost negligible when there is scene change in video sequences. In such cases, intraframe coding techniques are used to achieve video com-
MPEG compression algorithms use discrete cosine transform (DCT) coding technique on 8x8 image blocks for spatial correlation between pixels within a frame and interframe differential pulse code modulation (DPCM) coding technique for temporal correlation to find motion compensated prediction between frames. In MPEG video encoders, combination of both temporal prediction followed by transform coding is used to achieve high compression (Hybrid DPCM/DCT video coding) (Sakora (1997b)).

1.2 A General Video Encoder

As discussed in previous section, a video encoder uses intraframe and interframe coding techniques to reduce spatial and temporal redundancy respectively. A block diagram of a general video encoder is shown in fig.1.1 which uses discrete cosine transform (DCT), quantization and variable size run length coding for intraframe coding whereas concept of motion estimation is used for interframe coding. These stages are briefly covered in the following sections. Since a video encoder uses the concept of I, P, and B pictures for coding of a video, a brief introduction of these types of frames is given below.

1.2.1 Types of Video Frames

A complete video may consists of various scene changes in its entire span and there will not be correct prediction using previous frame as reference frame when there is such scene change. Therefore, entire video is divided into group of pictures (GOPs) with one group continues until there is scene change which is followed by next GOP in that order. These GOPs consists of I, P and B pictures which are defined as follows.

I Frames
Frames encoded using intraframe coding technique only are called I frames. First frame of a video sequence is an I frame which is coded spatially with no reference to any other frame in sequence. Pictures with scene change are also coded as I frames.

P Frames
Starting with an intra, or I frame, the encoder can forward predict a future frame. This
is commonly referred to as a P frame, and it may be predicted from other P frames, although only in a forward time manner. As an example, consider a group of pictures that lasts for 6 frames. In this case, the frame ordering is given as I,P,P,P,P,I,P,P,P,...... Each P frame in this sequence is predicted from the frame immediately preceding it, whether it is an I frame or a P frame.

**B Frames**

These frames are also called bi-directional interpolated prediction frames as they are predicted using both forward and backward predictions. An example is a group of pictures that lasts for 6 frames and is given as I,B,P,B,P,B,I,B,P,B,P,B,...... As in previous example, I frames are coded spatially only and the P frames are forward predicted based on previous I and P frames. The B frames however, are coded based on a forward prediction from a previous I or P frame, as well as a backward prediction from a succeeding I or P frame. It can be seen that the future frame required for backward prediction of a B frame must be encoded and transmitted first, out of order. There is no limit to the number of consecutive B frames that may be used in a group of pictures, and it is application dependent. However, most broadcast quality applications use 2 consecutive
B frames (I,B,B,P,B,P,...) as the ideal trade-off between compression efficiency and video quality.

B frames have advantage of better coding efficiency and quality of video inputs with movement of hidden areas can better be coded using such frames. Further, since B frames are not used to predict future frames, errors generated will not be propagated in the sequence. Such frames have disadvantages of double memory buffer as two reference frames are required for encoding. Further, there is a drawback of delay in encoding/decoding process as frames are processed out of order.

1.2.2 Intraframe Coding

The term intraframe coding refers to the fact that compression is performed relative to information that is contained only within the current frame and not relative to any other frame in the video sequence. In other words, no temporal processing is performed outside of the current picture or frame. It is very similar to the Joint Photographic Experts Group (JPEG) image encoder for coding still images except at the quantization stage which is uniform for video encoder. The basic processing blocks for intraframe coding are – DCT, quantization, and run length/variable length coder which are summarized below.

**DCT**

Generally, pixels within an image tend to be highly correlated and it is desired to use a transform coding to decorrelate it for image compression. DCT has been shown to be near optimal in literature for this purpose. It concentrates energy and decorrelates pixel intensities within the image, processing 8x8 image block at a time. It confines energy associated with low frequency components of input image towards upper left corner and high frequency components towards lower right corner in its output matrix.

**Quantization**

The aim of quantization is to compress image by representing DCT coefficients with no more precision than necessary to achieve desired image quality, that is, the goal of quantization is to discard the image information which is not visually significant. Quantization is a lossy part of intraframe encoders as it maps DCT coefficients in one
to many way.

**Run length/Variable length Coder**

Since most of the non-zero quantized DCT coefficients are typically concentrated in the upper left-hand corner of the matrix, zigzag scanning of coefficients is used to maximize the probability of achieving long runs of consecutive zeros. Scanning of quantized DCT coefficients in zigzag pattern results in a sequence of numbers. This sequence is then represented as a run-length (representing the number of consecutive zeroes) and an amplitude (coefficient value following a run of zeroes). These values are then looked up in a fixed table of variable length codes constructed by using huffman coding technique to find the corresponding bit stream representation. Variable length code generated by huffman coding is based on greedy approach where most probable occurrence is given a relatively short code, and the least probable occurrence is given a relatively long code.

### 1.2.3 Interframe Coding

Interframe coding uses motion estimation to remove temporal redundancy in video data. In most real video sequences, consecutive frames are very similar to each other (until there is scene change) except movement of some objects within these frames. Since a frame may have multiple objects moving randomly in different directions, motion of these objects can not be accurately predicted if entire frame is processed as a single unit. Therefore, each frame of input video is divided into fixed size blocks, called macroblocks (MB) and motion of objects is predicted at macroblock level. Standard macroblock size is 16x16 as it gives a trade off between computation complexity and quality parameters.

A current frame block (or macroblock) is searched for its best match in the reference frame within a search window. Full search method is the most accurate method of searching as it exhaustively searches all points in the search window but it has very high computation. There are various fast searching algorithms in literature which will be listed out in chapter 2. Location of best match block within the search window in the reference frame with respect to the location of candidate block in the current frame is called *motion vector*. Motion vectors of all blocks in the current frame are
calculated and a predicted frame is generated from the reference frame with the help of these motion vectors. Finally, the difference between the predicted frame and the actual frame, known as residue error, is coded for transmission/storage purposes. Frame store is a memory buffer which is used to store previously coded reference frame. Interframe coding efficiency depends on the amount of zeros in this difference frame.

Since this research work is highly centered around motion estimation in a video encoder, therefore a separate section is dedicated to this topic exclusively.

1.3 Motion Estimation

As discussed earlier, block based motion estimation algorithms (BMEAs) have been used in the past to generate motion estimated frames. Each frame is divided into fixed size block and a current block is searched for its best match in the reference frame in a search window as shown in fig.1.2.

![Motion Estimation Diagram](image)

Figure 1.2: Block based Motion Estimation.

Full search method is optimal as far as matching error is concerned as it searches all candidate points in the search window but it is not used in real encoders for its very high computational cost. Further, mean absolute error (MAE) is used as a similarity measure criterion which is not suitable for inputs with varying contrast and fast motion.
Therefore, motion estimation process itself incorporates two major challenges to the research community—

1. Design of an accurate block matching criterion which reduces matching error and it should be fast enough to meet real time applications.

2. Design of a fast and correct BMEA which reduces searching cost of candidate blocks by selecting limited number of search points in the search window and produces quality results.

It is shown in fig.1.3. Further, these issues have been discussed in detail in chapter 3 and chapter 4 of this thesis.

![Diagram](image)

Figure 1.3: Division of a Motion Estimation Process

### 1.4 Video Decoder

To decode the bit streams generated by a video encoder, processing is done in reverse order as shown in fig.1.4. Bit streams from input buffer are variable length decoded, dequantized by multiplying the decoded coefficients by the corresponding values of the quantization matrix. Finally, inverse DCT is applied to these coefficients to get the residual 8x8 block which are merged to generate the corresponding I frame or residue frame (for P/B frames).

A previously decoded I/P frame which is stored in frame store buffer, is motion
Figure 1.4: Block diagram of a general Video Decoder.

compensated to obtain predicted next frame in sequence which is added to the residue frame to get the desired next frame.

1.5 Organization of thesis

This thesis is organized in six chapters. Chapter 1 covers introduction to video compression. Chapter 2 gives a detailed literature survey about different techniques suggested by various authors. Further, pros and cons of existing techniques have also been discussed which have thoroughly been examined and reviewed.

Chapter 3 focuses on similarity criterions used in fast block based motion estimation algorithms (BMEAs) and a block matching criterion has been proposed which produces better results in terms of various parameters. Proposed technique is very robust for inputs with contrast variations. Further, integral frame based motion estimation has been studied and a multilevel block matching criterion based on this concept has been proposed to minimize the possibility of spurious block matching – a drawback of integral
frame oriented methods.

In chapter 4, a dynamic block motion estimation algorithm has been proposed which takes advantage of spatial and temporal correlations between adjacent MBs and substantial gain has been obtained for the proposed algorithm within same quality parameter. This algorithm has further been modified to minimize search computation.

A detailed experimental analysis has been performed over different video sequences using various performance parameters in chapter 5. These inputs have different resolutions, motions and contrast variations. Results in this chapter are based on various techniques discussed in chapter 3 and chapter 4.

Chapter 6, concludes this research work followed by possible future works.

1.6 Key contributions of proposed thesis

Thesis is centered around motion estimation process in a video encoder. Various key contributions of this thesis are summarized below –

1. Two block matching criterions have been proposed - one using pixel mapping based criterion and other is multilevel block matching criterion using integral frame features. The pixel mapping based criterion further has two versions – first is based on local block features while later is using global frame features.

2. A dynamic pattern search based block motion estimation algorithm has also been proposed which uses spatial and temporal correlation among blocks to find motion vector of a candidate block. Proposed algorithm has again been modified to further reduce its search cost.