

# Video Compression Based on DCT-DWT Technique

Miss. S. S. Wadd

Dr. JJ.Magdum College of Engineering, Jaysingpur, India

Dr. Mrs. S. B. Patil

Dr. JJ.Magdum College of Engineering, Jaysingpur, India

**Abstract** - The increasing demand to incorporate video data into telecommunications services, the corporate environment, the entertainment industry, and even at home has made digital video technology a necessity. To reduce bandwidth requirement in transferring files is a major challenge today. This review paper introduces advancement in video compression technology with perceiving quality and content of video. Block matching algorithms used for motion estimation in video compression

**Index terms:** Block matching, motion estimation, video compression

## 1. INTRODUCTION

With the advent of multimedia age and spread of internet video storage on CD/DVD and streaming video has been gaining a lot of popularity. The ISO Moving Picture Experts Group (MPEG) video coding standards pertain towards compressed video storage on physical media like CD/DVD, whereas the International telecommunications Union (ITU) addresses real-time point-to-point or multi-point communications over a network.[6]

Video compression algorithms manipulate video signals to dramatically reduce the storage and bandwidth required while maximizing perceived video quality. The video compression algorithms are "lossy." That is, the original uncompressed image can't be perfectly reconstructed from the compressed data, so some information from the original image is lost. Lossy compression algorithms attempt to ensure that the differences between the original uncompressed image and the reconstructed image are not perceptible to the human eye.

## 2. Basic Video Compression System

The basic video compression system comprises of the video encoder at the transmitter side, which encodes the video to be transmitted in terms of bits and the video decoder at the receiver side, which reconstructs the video in its original form from the bit sequence received. The sub systems of encoder and decoder are discussed below with the detailed technology aspects from theory point of view.[8]

Basic Block diagram of video encoder shown in Figure 1.

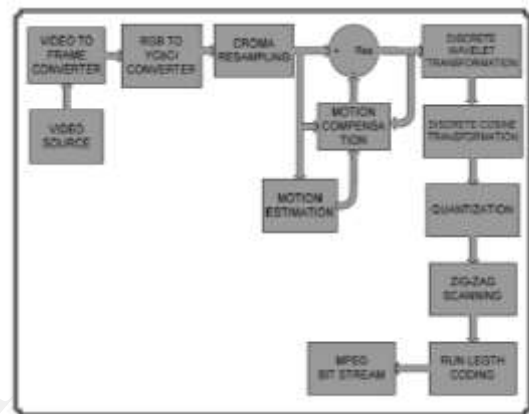


Figure 1. Block diagram of Video Encoder

### 2.1 Video encoder is designed as follows:

Video data may be represented as a series of still image frames. Group of frames are selected and store the image in YCbCr color space.

There are three kinds of frames in group of picture (GOP). GOP: I frame, P frame and B-frame.

#### 2.1.1 Intra Coded (I frame):

I frames are independent frames. The encoding of I frame is similar to still image sequence. I frames are coded as Intra coded frames. [6]

#### 2.1.2 Intra Frame Coding Techniques:

Process the data in blocks of 8x8 samples. The first step in the compression process involves switching from the RGB color space to the YCrCb color space. YCrCb describes a color space where the three components of color are luminance, red chrominance, and blue chrominance. This switch is made because the human eye is less sensitive to chrominance than it is to luminance.[3] Chrominance data can then be sampled at a quarter the rate of luminance data insignificant.

The next step in compression involves reducing spatial redundancy. This is done using essentially the same methods as JPEG. The image is divided into 16 x 16 pixel macroblocks. Each macroblock contains 16 x 16 luminance pixels, and 8 x 8 red/blue chrominance pixels. The luminance block is then split into 4 x 8 x 8 blocks. Now we have 6 x 8 x 8 blocks on which a DCT is performed. Energy tends to be concentrated into a few significant DCT

coefficients. Other coefficients are close to zero / insignificant.

The DCT coefficients are quantized by, Dividing each DCT coefficient by an integer; discard remainder, this result loss of precision. Typically, a few non-zero coefficients are left. [9]

Scan quantized coefficients in a zig-zag order, non-zero coefficients tend to be grouped together. Encode each (run, level) pair using a variable-length code

2.1.3 Non-Intra Frame Coding Techniques:

The next step in compression is intended to reduce temporal redundancy. The first step in this process is to divide a series of frames into a group of pictures (GOP) and then to classify each frame as I, P, or B. The usual method is to break a video into GOPs of 15 frames. The first frame is always an I frame. In a 15 frame GOP, it is common to have two B frames after the I frame, followed by a P frame, followed by two B frames, etc. [10]

The classification of a frame as I, P, or B determines the manner in which temporal redundancies are encoded. An I frame is encoded "from scratch", just as described above.

Moving images contain significant temporal redundancy; Successive frames are very similar as shown in figure 2.



Figure 2. Successive Frames in a Video

Temporal processing is to exploit this redundancy using a technique known as block-based motion compensated prediction, using motion estimation. [3]

Detail explanation of P, B frames given as below:

Predictive Frame (P-frame):

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 also be predicted from other P frames, although only in a forward time manner.

Figure 3. shows dependencies in I, P and B frames.

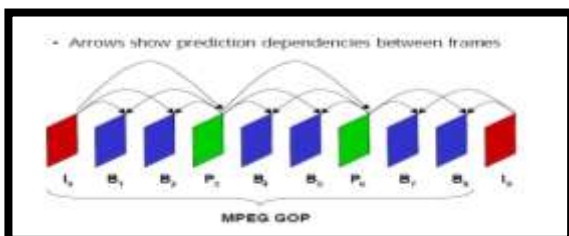


Figure 3. I, P and B frames

Bidirectional predictive frame (B frame):

These frames are commonly referred to as bi-directional interpolated prediction frames, or B frames. Bi-directional motion predicted from previous or future frame. The reference can be either I or P frame. Generally referred to as Inter-frame. [6]

2.2 Block Matching Techniques:

Block matching techniques match blocks from the current frame with blocks from a reference frame. The displacement in block location from the current frame to the location in the reference frame is the motion vector. [7] Block matching techniques can be divided into three main components as shown in Figure 4 block determination, search method, and matching criteria.

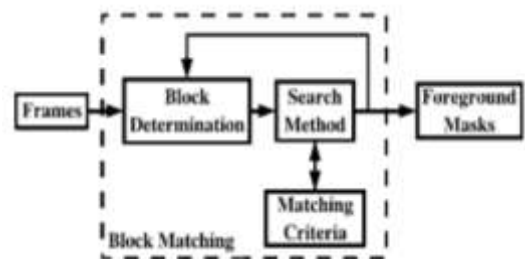


Figure 4: Block Matching Flowchart

The first component, block determination, specifies the position and size of blocks in the current frame, the start location of the search in the reference frame, and the scale of the blocks.

The search method is the second component, specifying where to look for candidate blocks in the reference frame. A fully exhaustive search consists of searching every possible candidate block in the [4]

reference frame. This search is computationally expensive and other search methods have been proposed to reduce the number of candidate blocks and/or reduce the processing for all candidate blocks.

The third component is the matching criteria. The matching criteria is a similarity metric to determine the best match among the candidate blocks.

2.3 Motion Estimation and Motion Compensation:

The temporal prediction technique used in MPEG video is based on motion estimation. The basic premise of motion estimation is that in most cases, consecutive video frames will be similar except for changes induced by objects moving within the frames. In the trivial case of zero motion between frames (and no other differences caused by noise, etc.), it is easy for the encoder to efficiently predict the current frame as a duplicate of the prediction frame. When this is done, the only information necessary to transmit to the decoder becomes the syntactic overhead necessary to reconstruct the picture from the original reference frame. When there is motion in the images, the situation is not as simple. Figure 5 shows an example of a frame with 2 stick figures and a tree. [5]

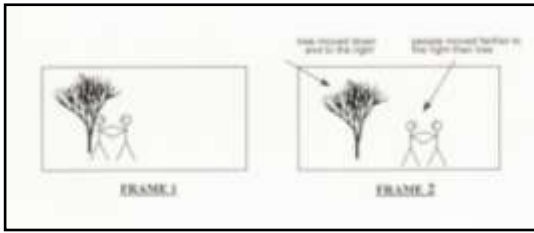


Figure 5 Frames shows differences

Motion estimation is not applied directly to chrominance in MPEG video, as it is assumed that the color motion can be adequately represented with the same motion information as the luminance. It is well known that a full, exhaustive search over a wide 2-dimensional area yields the best matching results in most cases, but this performance comes at an extreme computational cost to the encoder. [3]

Figure 6 shows an example of a particular macroblock from Frame 2 of Figure 7, relative to various macroblocks of Frame 1. As can be seen, the top frame has a bad match with the macroblock to be coded. The middle frame has a fair match, as there is some commonality between the 2 macroblocks. The bottom frame has the best match, with only a slight error between the 2 macroblocks.[7] Because a relatively good match has been found, the encoder assigns motion vectors to the macroblock, which indicate how far horizontally and vertically the macroblock must be moved so that a match is made. As such, each forward and backward predicted macroblock may contain 2 motion vectors, so true bidirectionally predicted macroblocks will utilize 4 motion vectors. [4]

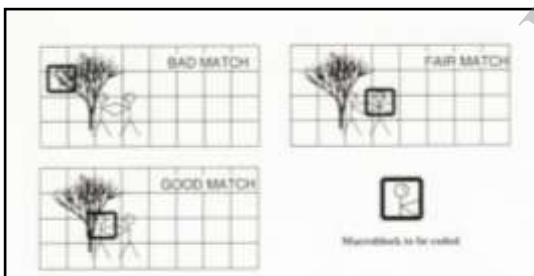


Figure 6 Macroblock Encoded

In this figure, the predicted frame is subtracted from the desired frame, leaving a (hopefully) less complicated residual error frame that can then be encoded much more efficiently than before motion estimation. It can be seen that the more accurate the motion is estimated and matched, the more likely it will be that the residual error will approach zero, and the coding efficiency will be highest.[3]

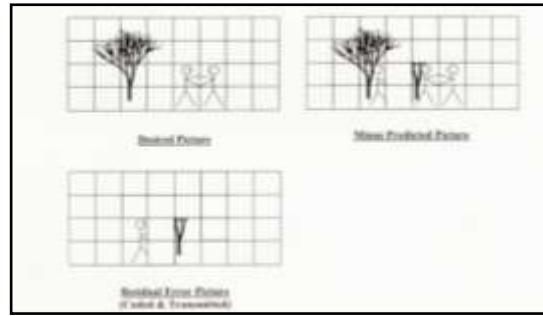


Figure 7 Residual Error Image

In this manner, high quality video is maintained at a slight cost to coding efficiency. After a predicted frame is subtracted from its reference and the residual error frame is generated, this information is spatially coded as in I frames, by coding 8x8 blocks with the DCT, DCT coefficient quantization, run-length/amplitude coding, and bitstream buffering with rate control feedback.

2.4 Decoding System:

Decoding system is the exact reverse process of encoding.[8] There are four steps for getting the original image not exact but identical to original from compressed image.

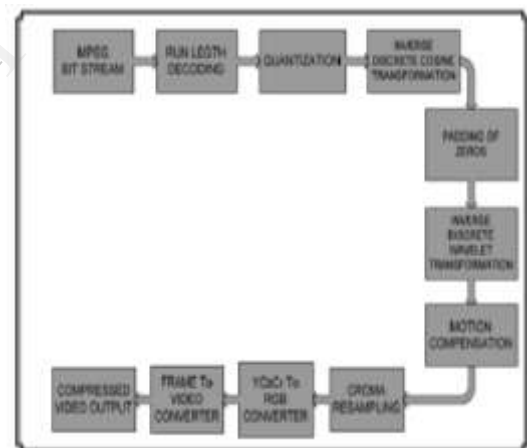


Figure 8 Block diagram of video decoder

Design Steps for Decoder:

- Step1. Load compressed image from disk
- Step2. Image is broken into N\*N blocks of pixels.
- Step3. Each block is de-quantized by applying reverse process of quantization.
- Step4. Now apply inverse DCT on each block. And combine these blocks into an image which is identical to the original image. [10].

### 3 MEASURING PARAMETER:

Following parameters are subjective measure of degradation of video after the compression. [1]

#### 3.1 PSNR (peak signal-to-noise ratio) :

PSNR is measured for degradation of video after the compression. PSNR is ratio between the maximum possible power of a signal and the power of corrupting noise.

$$\text{PSNR} = 10 * \log \frac{255^2}{\text{MSE}}$$

#### 3.2 SSIM (Structural Similarity):

The **structural similarity** (SSIM) index is a method for measuring the similarity between two images.

The SSIM metric is calculated on various windows of an image. The measure between two windows  $\mathbf{X}$  and  $\mathbf{Y}$  of common size  $N \times N$  is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

With

- $\mu_x$  the average of  $\mathbf{x}$ ;
- $\mu_y$  the average of  $\mathbf{y}$ ;
- $\sigma_x^2$  the variance of  $\mathbf{x}$ ;
- $\sigma_y^2$  the variance of  $\mathbf{y}$ ;
- $\sigma_{xy}$  the covariance of  $\mathbf{x}$  and  $\mathbf{y}$ ;
- $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$  two variables to stabilize the division with weak denominator;

#### 3.3 Compression Ratio :

The compression ratio between two videos: the original one and the compressed one coming out of video encoder, which is measured by equation as below:

$$\text{CR} = 100 * \frac{\text{Compressed\_data\_rate}}{\text{Uncompressed\_data\_rate}}$$

### 4 REFERENCES :

1. Weinberger, M.J., et al.: 'The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS', IEEE Trans. Image Process., 2000, 9, pp. 1309–1324
2. CALIC, ftp://ftp.csd.uwo.ca/pub/from\_wu/
3. Memon, N.D., and Sayood, K.: 'Lossless compression of video sequences', IEEE Trans. Commun., 1996, 44, pp. 1340–1345
4. Martins, B., and Forchhammer, S.: 'Lossless compression of motion compensation'. Proc. IEEE DCC'98, Los Alamitos, CA, 1998, p. 560
5. Brunello, D., Calvagno, G., Mian, G.A., and Rinaldo, R.: 'Lossless compression of video using temporal information', IEEE Trans. Image Process., 2003, 12, pp. 132–139.
6. Dragos Ruiu, 'An Overview of MPEG-2', The 1997 Digital Video Test Symposium.
7. Aroh Barjatya, 'Block matching algorithms for Motion estimation', DIP 6620 Spring 2004 Final Project Paper
8. Barry G. Haskell, Atul Puri, Arun N. Netravali, Digital Video: An Introduction to MPEG-2,
9. Chapman and Hall, 1997.
10. K.R. Rao, P. Yip, Discrete Cosine Transform – Algorithms, Advantages, Applications, Academic Press, Inc., 1990.
11. Majid Rabbani, Paul W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering. Press, 1991.