Video Compression Using MPEG

Sangeeta Mishra
TCET,Mumbai
Kandivli(E)-101

Sudhir Sawarkar
DMCE
Airoli

ABSTRACT
In this paper the mpeg compression technique is being performed. The different encoding and decoding time were calculated .After observing for different no. of frames per iteration it was observed that encoding takes few minutes to encode the data but the decoding procedure is finished of in few seconds. i.e. if encoding is done the decoding is very fast.

General Terms
Communication

Keywords
Video Compression, MPEG , Frames

1. INTRODUCTION
In applications of multimedia technique, the transfers of the video and the audio data **are** very troublesome. Because the multimedia data is **so** large that the QoS of the multimedia transfer is very poor. The technology of the data compression can resolve this problem in the condition of the small bandwidth. So the multimedia data compression has become an important issue recently. Block
Matching (BM) is a very important stage in the video compression, and it provides an effective way to estimate an object's motion from time varying image sequences. In the algorithm, each image frame is divided into non-overlapping blocks, and the best displacement vector between two consecutive frames is searched for each block.
In the past **two** decades, there has been extensive research into motion estimation techniques. Block-based matching has **been** widely adopted by international standards such as the H.261, H.263, MPEG-2 and MPEG-4[1][2] due to its effectiveness and robustness. Therefore, most of the research works have been concentrated on optimizing the
block-based motion estimation technique.
Video images can be regarded as a three-dimensional generalization of still images, where the third dimension is time. Each frame of a video sequence can be compressed by any image compression algorithm. A method where the images are separately coded by JPEG is sometimes referred as *Motion JPEG* (*M-JPEG*). A more sophisticated approach is to take advantage of the *temporal* correlations; i.e. the fact that subsequent images resemble each other very much. This is the case in the latest video compression standard MPEG (*Moving Pictures Expert Group*).

2. MPEG
MPEG standard consists of both video and audio compression. MPEG standard includes also many technical specifications such as image resolution, video and audio synchronization, multiplexing of the data packets, network protocol, and so on. Here we consider only the video compression in the algorithmic level. The MPEG algorithm relies on two basic techniques

- Block based *motion compensation*
- DCT based compression

MPEG itself does not specify the encoder at all, but only the structure of the decoder, and what kind of bit stream the encoder should produce. Temporal prediction techniques with motion compensation are used to exploit the strong temporal correlation of video signals. The motion is estimated by predicting the current frame on the basis of certain previous and/or forward frame. The information sent to the decoder consists of the compressed DCT coefficients of the residual block together with the *motion vector*. There are three types of pictures in MPEG:

- Intra-pictures (*I*)
- Predicted pictures (*P*)
- Bidirectionally predicted pictures (*B*)

An **I-frame** is an 'Intra-coded picture', in effect a fully specified picture, like a conventional static image file. P-frames and B-frames hold only part of the image information, so they need less space to store than an I-frame, and thus improve video compression rates.

A **P-frame** ('Predicted picture') holds only the changes in the image from the previous frame. For example, in a scene where a car moves across a stationary background, only the car's movements need to be encoded. The encoder does not need to store the unchanging background pixels in the P-frame, thus saving space. P-frames are also known as *delta-frames*.

A **B-frame** ('Bi-predictive picture') saves even more space by using differences between the current frame and both the preceding and following frames to specify its content.

Typically, pictures (frames) are segmented into *macroblocks*, and individual prediction types can be selected on a macroblock basis rather than being the same for the entire picture, as follows:

- I-frames can contain only intra macroblocks
- P-frames can contain either intra macroblocks or predicted macroblocks
- B-frames can contain intra, predicted, or bi-predicted macroblocks

Furthermore, in the video codec H.264, the frame can be segmented into sequences of macroblocks called *slices*, and instead of using I, B and P-frame type selections, the encoder can choose the prediction style distinctly on each individual slice. Also in H.264 are found several additional types of frames/slices:

- SI-frames/slices (Switching I); Facilitates switching between coded streams; contains SI-macroblocks (a special type of intra coded macroblock).
- SP-frames/slices (Switching P); Facilitates switching between coded streams; contains P and/or I-macroblocks
- multi-frame motion estimation (up to 16 reference frames, or 32 reference fields)

Multi-frame motion estimation will allow increases in the quality of the video while allowing the same compression ratio. SI- SP-frames (defined for Extended Profile) will allow for increases in the error resistance. When such frames are used along with a smart decoder, it is possible to recover the broadcast streams of damaged DVDs.

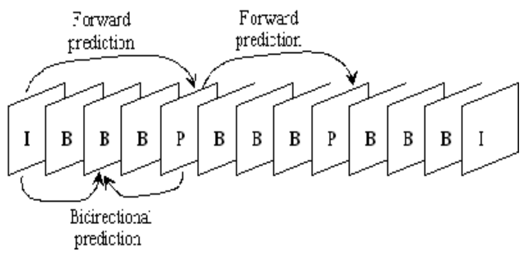

Figure 1: Interframe coding in MPEG.

Figure 1 demonstrates the position of the different types of pictures. Every N^{th} frame in the video sequence is an *I*-picture, and every M^{th} frame a *P*-picture. Here *N*=12 and *M*=4. The rest of the frames are *B*-pictures.

3. Compression of the picture types:

Intra pictures are coded as still images by DCT algorithm similarly than in JPEG. They provide access points for random access, but only with moderate compression. *Predicted pictures* are coded with reference to a past picture. The current frame is predicted on the basis of the previous *I*- or *P*-picture. The residual (difference between the prediction and the original picture) is then compressed by DCT. Bidirectional pictures are similarly coded than the *P*-pictures, but the prediction can be made both to a past and a future frame which can be *I*- or *P*-pictures. Bidirectional pictures are never used as reference.

The pictures are divided into 16×16 macroblocks, each consisting of four 8×8 elementary blocks. The *B*-pictures are not always coded by bidirectional prediction, but four different prediction techniques can be used:

- Bidirectional prediction
- Forward prediction
- Backward prediction
- Intra coding.

The choice of the prediction method is chosen for each macroblock separately. The bidirectional prediction is used whenever possible. However, in the case of sudden camera movements, or a breaking point of the video sequence, the best predictor can sometimes be given by the forward predictor (if the current frame is before the breaking point), or backward prediction (if the current frame is after the breaking point). The one that gives the best match is chosen. If none of the predictors is good enough, the macroblock is coded by intra-coding. Thus, the *B*-pictures can consist of macroblock coded like the *I*-, and *P*-pictures.

The intra-coded blocks are quantized differently from the predicted blocks. This is because intra-coded blocks contain information in all frequencies and are very likely to produce 'blocking effect' if quantized too coarsely. The predicted blocks, on the other hand, contain mostly high frequencies and can be quantized with more coarse quantization tables.

4. Motion estimation:

The prediction block in the reference frame is not necessarily in the same coordinates than the block in the current frame. Because of motion in the image sequence, the most suitable predictor for the current block may exist anywhere in the reference frame. The *motion estimation* specifies where the best prediction (best match) is found, whereas *motion compensation* merely consists of calculating the difference between the reference and the current block.

The motion information consists of one vector for forward predicted and backward predicted macroblocks, and of two vectors for bidirectionally predicted macroblocks. The MPEG standard does not specify how the motion vectors are to be computed, however, *block matching techniques* are widely used. The idea is to find in the reference frame a similar macroblock to the macroblock in the current frame (within a predefined search range). The candidate blocks in the reference frame are compared to the current one. The one minimizing a cost function measuring the mismatch between the blocks, is the one which is chosen as reference block.

Exhaustive search where all the possible motion vectors are considered are known to give good results. Because the full searches with a large search range have such a high computational cost, alternatives such as *telescopic* and *hierarchical searches* have been investigated. In the former one, the result of motion estimation at a previous time is used as a starting point for refinement at the current time, thus allowing relatively narrow searches even for large motion vectors. In hierarchical searches, a lower resolution representation of the image sequence is formed by filtering and subsampling. At a reduced resolution the computational complexity is greatly reduced, and the result of the lower resolution search can be used as a starting point for reduced search range conclusion at full resolution.

5. Algorithm:

1. Read the video clip.
2. Encoding
   - 2.1 give the file pattern ‘IPPPP’
   - 2.2 Convert the color format from RGB to YCbCr
   - 2.3 Start encoding the frame
     - 2.3.1 Define the size of the macroblock as 16×16
     - 2.3.2 Encode each and every macroblock
       - 2.3.2.1 Interframe and intraframe is being done on each and every frame

2.3.2.2 Calculate the motion vector using logarithmic search and the macroblock of size 4:2:0

2.3.2.3 Apply DCT to each 8×8 block

2.3.2.4 Decode this macroblock for refrence by a future P frame.

3. Decoding
   - 3.1 Decode the frame
   - 3.2 Cache the previous frame
   - 3.3 Convert the color format of frame from YCbCr to RGB
   - 3.4 Make sure that the frame is in 8bit range
   - 3.5 Store the frame
4. Calculate the time for encoding and decoding

6. Results And Discussion:

Here after the complete operation the analysis was done and encoding and decoding time for different number of frames was calculated .The entire clip for which the process was done is of 7sec.and 16frames per seconds were considered. Every time 10 frames were considered and corresponding encoding and decoding time were calculated .The same process was repeated for the entire movie and the time was noted down.

No. of Frames per turn	Encoding Time in sec.	Decoding Time in sec.
10	11.364758	2.141091
20	20.796058	4.135522
30	33.761324	6.205527
40	46.433764	8.163229
50	57.180136	10.168335
80	98.585535	16.533691
100	111.419055	21.227948
110	142.064458	22.899313

For entire clip

Encoding time:Elapsed time is 130.170707 seconds.

decoding time:Elapsed time is 22.720953 seconds.

7. REFERENCES

[1] Gary J. Sullivan,Thomas Wiegand "Video Compression—From Concepts to the H.264/AVC Standard" Proceeding of the IEEE, VOL. 93, NO. 1, JAN 2005.

[2] Milind Phadtare*, "*Motion estimation techniques in video processing" Electronic Engineering Times India August 2007 eetindia.com.

[3] Xiaomin Wu, Weizhang Xu, Nanhao Zhu, Zhanxin Yang, "A Fast Motion Estimation Algorithm for H.264"IEEE 2010 International Conference on Signal Acquisition and Processing.

[4] Wei-feng Liu, Ji-li Lu, Zeng-fu Wang, Hua-jun Song , Xue-zheng ,"High Compression Algorithm for Video Stream" IEEE 2008 Conference on Image and Signal Processing.

[5] Hui Gu, Yan-Shanli "The Research Of Block Motion Estimation Algorithm In Video Compression"Proceedings of the Third International Conference on Machine Leg and Cybernetics, Shanghai, 26-29 August 2004.

[6] Marc Alzina, Wojciech Szpankowski,and Ananth Grama2D-Pattern Matching Image and Video Compression: Theory, Algorithms, and Experiments IEEE Transaction On Image Processing, Vol. 11, No. 3, Mar 2002