MPEG VIDEO COMPRESSION WITH MOTION VECTOR COMPRESSION WITH DPCM

B. Shravan Kumar[1], Dr. V. Usha Shree[2]

[1]*Research Scholar JNTUH Hyderabad, India*

[2]*Professor and Principal, JBREC, R.R Dist, India*

Abstract: The ISO/IEC motion pictures experts group (MPEG) has been for several years investigating solutions for delivering immersive computer generated or captured video content. Standard MPEG video compression algorithms are designed to compress data into small bytes, which is easy to transmit and then decompress. In the proposed method, the Motion vector coding by DPCM achieves decoupling of successive samples. In the successive motion vectors it is possible that they have an equal or similar value with the result that their difference is small or even zero. Using the corresponding VLC codes to encode the differences, the representation of small numbers needs fewer bits and therefore the more the smaller the difference the fewer bits needed.

Keywords: MPEG, Video Compression, Motion vector, Motion Compensation

I. INTRODUCTION

The use of digital video has only spread widely in the in the recent years. However, from the 1960s, analog compression techniques had been tested, without any real success. In the 1980s, the first form of compression of digital video, H.120, appeared, using a compression scheme called DPCM, to be used in CDs. However, this technology was limited, and left something to be desired in terms of quality. Indeed, this codec used pixel-based compression, which does not allow downscaling to below one bit per pixel. This brought the era of so-called "block" codecs. The end of the 1980s thus brought many codecs based on DCT (Discrete Cosine Transform), also used by the standard JPEG [1].

The first codec to correctly use DCT was H.261, finalized in 1989. The beginning 1990s led the MPEG (Motion Picture Experts Group) to standardize the first digital VHS quality system, MPEG1. The rapid development of MPEG1 made it easy to access [2]. However, it suffered from many limitations, especially its non-adaptation to interlaced television, and its limitation data rate. MPEG2 thus appeared to solve these problems, and remains to this day the the most widely used format, from DVD Video to High Definition television (HDTV), by way of satellite video transmissions. This format leaves a wide choice of material and software for decoding, making it easy to use, as well as great adaptability. Its use remains planned in the long term since HDTV television provides for its globalization throughout its territory in 2014. Research and development of cuts video does not stop there, however, and since 1998 the MPEG4 standard has appeared. The MPEG4 uses more complex technology and therefore has much better results, but its use remains at present essentially experimental [3].

A summary of various Compression techniques is shown in table

Table 1: Video Compression standards

Format	Description	1 minute (MB)	Quality evaluation rated from 1 to 5	Benefits	Disadvantages
DV format [4]	Digital Video Multimedia format developed by a consortium of electronics companies in 1996. Very little compression, it is a format particularly suitable for editing and storage.	210	5	- Excellent quality, recommended format for editing. - DV cassettes are a safe medium for storage	File size (approximately 13 GB / hour)
MPEG 1 (PAL VCD: 352 * 288 + Audio 224 Kbits) [5]	MPEG 1 is an audio and video data compression standard in 1988 created by the MPEG group. The bit rate it offers (1.2Mbits) allows you to put a VHS quality video stream on CD (1x)	10	3	- Good quality in native resolution (352 * 288) - format very present in analysis software - standardized format	- Aging format - relatively low native resolution
Xvid (bit rate: 1265 Kbits) [6]	Xvid is an implementation of the MPEG-4 standard. I t is distributed under the GNU General	6.8	4.5	- excellent quality rendering - free format (GNU / GPL)	format little used in analysis software

	Public License (GPL).				
DivX (1 200 Kbits speed) [7]	Format created in 1998 by Jerôme Rota on the basis of Microsoft's MPEG4-v3. It has been distributed by DivX Networks since version 4.	9.6	4.5	- Excellent quality - integrated in certain salon equipment (DVD player, etc.)	- proprietary codec under license - format little used in analysis software
MPEG 4 [8]	MPEG 4 is a set of specifications aimed at encoding multimedia objects. It was designed in 1998 by the MPEG group	9.6	4	- standardized format - fast compression	- quality of rendering can be improved
H264 [9]	H264 is a video compression standard finalized in 2003 by the MPEG and ITU-T groups. It is also known under the names "MPEG-4 part 10" and "MPEG-4 AVC"	9.6	4.5	- Excellent quality - standardized format	- excessively long compression time

II. PROPOSED METHODOLOGY

The proposed methodology is applied on P frame encoding where motion vectors are estimated. The input frame is decomposed into three components namely RGB. The RGB colorspace is converted to YCbCr. Motion estimation is performed to record the difference between the encoding frame and the reference frame stored within the frame buffer. Dividing image into several macroblocks (each macroblock has 6 blocks , 4 for Y, 1 for Cb, 1 for Cr). The consecutive motion vectors are subtracted to reduce number of bits during VLC. The block diagram is shown in figure 1.

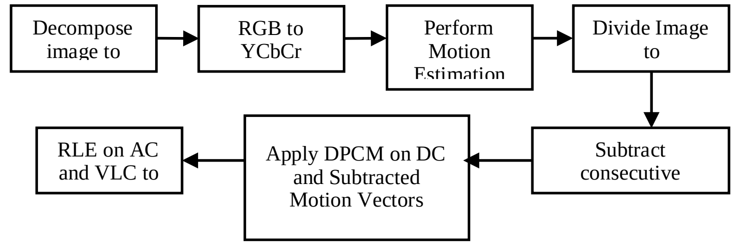


Fig. 1: Block Diagram

2.1 Color Space

Digital images are represented by a function with two variables, the coordinates, which returns the color to the point asked for in the image. The classic representation of a digital image uses a space of color says RGB. This corresponds to a discrete representation of the image where the color in each point is quantified by three values: Red, Green and Blue (RGB). This presentation, however, has several flaws: it uses a large amount of information and does not take into account the fact that the human eye does not perceive colors in RGB but through worms of two types of cells: one perceives luminosity (in black and white), and the others coloring.

MPEG codecs therefore do not compress RGB images, but use a more appropriate color space, so as to better take advantage of the format of human vision. The format adopted is a so-called YUV format. Y represents the luminosity U represents the first value of chrominance (or chroma) V represents the second chrominance (or chroma) value. Three independent components are needed to be able to have a complete color space. The formulas allowing to go from an RGB space to YUV and vice versa are quite simple, and correspond to matrix products (These are base changes, since they are 3-dimensional vector spaces).

$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$

$Cr = V = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$

$Cb = U = - (0.148 * R) - (0.291 * G) + (0.439 * B) + 128$

and

$B = 1.164 * (Y - 16) + 2.018 (U - 128)$

$G = 1.164 (Y - 16) - 0.813 (V - 128) - 0.391 (U - 128)$

$R = 1.164 (Y - 16) + 1.596 (V - 128)$

Using the YUV format makes it possible to take advantage of a characteristic of the human eye:

Its color distinguishing ability is weaker than that of brightness distinguishing. This makes it possible to reduce the amount of information of the chrominance spaces compared to that of luminosity, components U and V are subsampled.

2.2 Discrete cosine transform, Quantization & Compression of the block

DCT transforms a block of one component into a set of frequencies describing the same set (this is a change of isomorphic representation). Once more the final goal and take advantage of the weakness of the human eye which notices much less a data loss distributed as localized.

The formal definition of two-dimensional DCT is as follows:

F (u, v) is the transform, it is the function which gives the value for the frequency pair (u, v).

$$F(u,v) = \frac{2}{N} * C(u) * C(v) * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) cos\left(\frac{(2x+1)u\pi}{2N}\right) cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

with u, v, x, y = 0, 1, 2, ... N - 1
where x and y are the spatial coordinates
where u and v are the coordinates in the transform

and with C (u), C (v) = {

and the reverse transformation, the IDCT is defined as follows:

$$f(x,y) = \frac{2}{N} * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u)C(v)F(u,v) cos\left(\frac{(2x+1)u\pi}{2N}\right) cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

In practice in MPEG encodings, DCT is used on blocks. The block is transformed in the frequency domain by DCT. When decoding IDCT (Inverse Direct Cosine Transform) is applied which allows to go back to the spatial domain.

Quantization takes place just after transforming a block via the DCT. Quantization the same as dividing an array by another array. Each value obtained is divided by DCT by the corresponding value in an array. The classic method used by MPEG4, H263, divide all the coefficients by the same value. This makes it possible to "simplify" the information contained, and therefore to make compression easier.

2.3 Type of frames and global operation

There are three types of frame in MPEG compression: I frames, called Intra Frames, or key frames The P frames, called Inter Frames, Delta Frames, or prediction frames The B frames, or Bidirectional frames

I Frames

I frames look a lot like JPEG compression. Their coding returns simply take the image, cut it into 8x8 pixel blocks, and encode each block via DCT, quantization and compression (see Figure). We modify the quantization coefficient if we wants to change the final size, since this coefficient allows you to play on the remaining information present. Their name, Intra, derives from the fact that they are completely independent of any other information in the video; these frames are of course very quickly greedy in memory, but have the advantage of having an entire encoded image.

P Frames and B Frames

The P frames are obtained in a more complex way. First "motion blocks" or blocks of movement are found. These blocks are in fact blocks that are found in the previous frame and which have simply moved. This corresponds to the idea that the next frame will be largely similar to the existing frame, and that many pieces will only have moved. This idea makes it possible to gain greatly in compression since each block found by motion compensation will not need to be completely encoded, it will suffice to store its vector. We therefore seek a maximum of these blocks to be able to reduce the amount of information needed. Information that cannot be found by comparison thought of movement will be coded traditionally; as in an I frame, by encoding DCT. The P frame is therefore a mosque of blocks composed of a vector (to indicate where located the block in the previous frame) and complete blocks. In practice, these vectors, called "motion vectors", are sought on a fixed size of 16x16 pixel (i.e. 4 blocks) (in below, the vector will have an efficiency much lower than that of a complete coding).

B frames push the concept of P frames further; whereas a P frame is based only on the previous frame, the B frame also uses frames "in the future" to make motion compensation. This still allows a new gain (in particular when the compression is strong) but requires the ability to decode frames in advance.

2.4 Motion estimation

To calculate the motion compensation, the motion estimate is precomputed. This technique works by precomputing motion vectors. For that a block is taken and compared it in a given radius to all the surrounding blocks in the next frame. If it corresponds to a block then the vector which corresponds to it is calculated. This vector will allow then to determine the next P frame / B frame, much more easily.

Flow and quality control

The quantization allows to instantly regulate the size of the frames by increasing / decreasing the total amount of information, to the detriment of quality. The amount of research done per- optimizes the size without losing quality, but requires more processing (this is therefore not a viable solution for hardware processing where the quantity of calculations is fixed). Finally, when looking for compensation vectors for the P frames and B frames, a coefficient to judge whether two blocks are sufficiently similar or not is established. This value greatly influences the number of vectors found, and therefore the final size of the image. By giving back this very small coefficient, a large number of corresponding vectors are found, but as these do not correspond exactly, the appearance of visible blocks in the image is seen, very classic phenomenon in time compression by block. Increase quantization also deteriorates the quality, and makes visible frequency noise, not only showing blocks, but also many small "mosquitoes", this is Gibbs' quantization noise.

For quality control it is a question of using an appropriate quality measurement. The most used is generally the PSNR (peak signal to noise ratio) which provides a good estimate of the "distance" from one image to another. This measure, however, does not reflect human perception and it may be that an image with a very good psnr (a high psnr) is of poor quality. An image with a good psnr, however, usually indicates good quality.

2.5 Differential Pulse Code Modulation

DPCM code words represent differences between samples unlike PCM where code words represented a sample value. Basic concept of DPCM - coding a difference, is based on the fact that most source signals show significant correlation between successive samples so encoding uses redundancy in sample values which implies lower bit rate.

The DPCM value is the difference in the corresponding motion vectors.

For P Frame Mblock, the DPCM value is calculated by

- dpcmMV = currMV(:,1)-prevMV(:,1);

For B frame Mblock, the DPCM value is calculated by

- dpcmMV = currMV-prevMV;

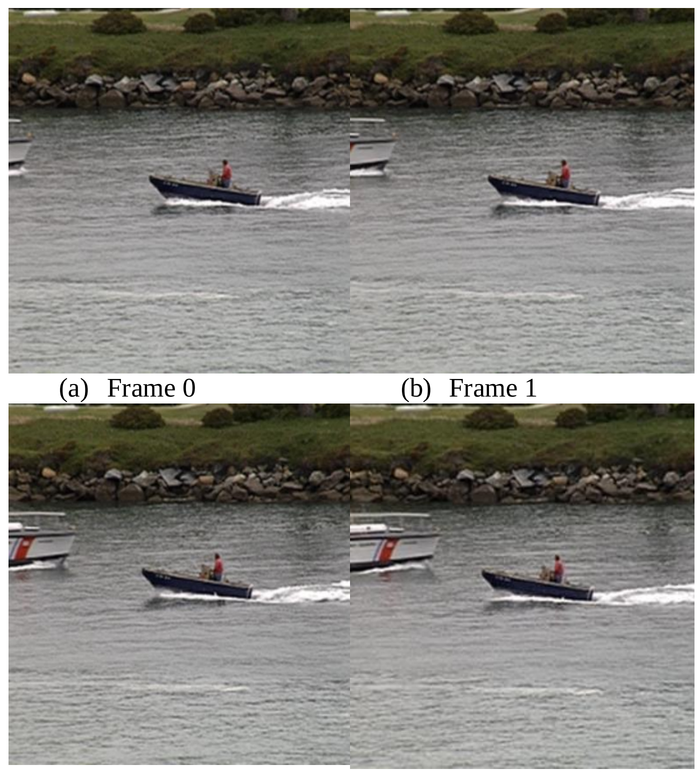
2.6 VLC

Variable-length codes can allow sources to be compressed and decompressed with zero error (lossless data compression) and still be read back symbol by symbol. With the right coding strategy an independent and identically-distributed source may be compressed almost arbitrarily close to its entropy. This is in contrast to fixed length coding methods, for which data compression is only possible for large blocks of data, and any compression beyond the logarithm of the total number of possibilities comes with a finite (though perhaps arbitrarily small) probability of failure.

Table 2: VLC codes

-16	p = '00000011001';
-15	p = '00000011011';
-14	p = '00000011101' ;
-13	p = '00000011111' ;
-12	p = '00000100001' ;
-11	p = '00000100011' ;
-10	p = '0000010011' ;
-9	p = '0000010101' ;
-8	p = '0000010111' ;
-7	p = '00000111' ;
-6	p = '00001001' ;
-5	p = '00001011' ;
-4	p = '0000111' ;
-3	p = '00011' ;
-2	p = '0011' ;
-1	p = '011' ;
0	p = '1' ;
1	p = '010';
2	p = '0010';
3	p = '00010';
4	p = '0000110' ;
5	p = '00001010' ;
6	p = '00001000';
7	p = '00000110' ;
8	p = '0000010110' ;
9	p = '0000010100' ;
10	p = '0000010010' ;
11	p = '00000100010' ;
12	p = '00000100000' ;
13	p = '00000011110' ;
14	p = '00000011100' ;
15	p = '00000011010' ;
16	p = '00000011000';

III. EXPERIMENTAL RESULTS



(a) Frame 0 (b) Frame 1

(c) Frame 2 (d) Frame 3

Fig 2: Input Video Frames

With this data the encoder achieves encoding which produces a file size **87.625KBytes** . The original size of the 4 images is calculated as follows. If we consider that we have a GRB system with 8bit color representation and 720x576 images then I need

720 * 576 * 3 bytes for each image, so in total for all 4 images 720 * 576 * 3 * 4 = 4976640 bytes that is about **4.976MB**

So we see that the compression achieved is 4976Kbytes / 87.625KBytes = 56.8

Therefore we have reduced the volume of data **by 57 (approximately) times** .

The average absolute error for each frame is:

Mean_absolute_error of frame0 = 3.694568

Mean_absolute_error of frame1 = 9.923167

Mean_absolute_error of frame2 = 10.200811

Mean_absolute_error of frame3 = 8.893836

In addition we can observe the dependence on the quantization scale of results on compression size and image quality.

For qScale = 16 we have:

Compressed size = 57.7 KBytes and errors per image

Mean_absolute_error of frame0 = 3.812056

Mean_absolute_error of frame1 = 10.019465

Mean_absolute_error of frame2 = 10.378625

Mean_absolute_error of frame3 = 9.188182

Proposed motion vector compression based MPEG

Running the proposed method with the same data as in the previous case it turns out that their compressed stream size is equal to **82072** Bytes or **82.072KBytes.** Comparing this result with the previous size 87.625KBytes we see that the our compression increased 4976.664Kbytes / 82.072KBytes = **60.63**. This result is consequence of the Motion vector coding by dpcm which achieves decoupling of successive samples. The above results from the observation that successive motion vectors it is possible that they have an equal or similar value with the result that their difference is small or even zero. Using the corresponding VLC codes to encode the differences we see that the representation of small numbers needs fewer bits and therefore the more the smaller the difference the fewer bits we need.

IV. CONCLUSION

The proposed modified MPEG coding increased the speed of the operation and compression ratio compared to traditional MPEG coding. The use of compression on motion vectors reduced the encoded data size thereby increasing the compression ratio. The use of Adaptive Rood Pattern Search for block matching reduces the operation time while maintaining the quality of the reconstructed video.

References

[1] Schwarz S, Preda M, Baroncini V, Budagavi M, Cesar P, Chou PA, Cohen RA, Krivokuća M, Lasserre S, Li Z, Llach J. Emerging MPEG standards for point cloud compression. IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 2018 Dec 10;9(1):133-48.

[2] Karwowski D, Grajek T, Klimaszewski K, Stankiewicz O, Stankowski J, Wegner K. 20 years of progress in video compression–from MPEG-1 to MPEG-h HEVC. General view on the path of video coding development. InInternational Conference on Image Processing and Communications 2016 Sep 7 (pp. 3-15). Springer, Cham.

[3] He P, Jiang X, Sun T, Wang S. Detection of double compression in MPEG-4 videos based on block artifact measurement. Neurocomputing. 2017 Mar 8;228:84-96.

[4] Wu HR, Rao KR, editors. Digital video image quality and perceptual coding. CRC press; 2017 Dec 19.

[5] McCormack RM, Szymanski EP, Hsu AP, Perez E, Olivier KN, Fisher E, Goodhew EB, Podack ER, Holland SM. MPEG1/perforin-2 mutations in human pulmonary nontuberculous mycobacterial infections. JCI insight. 2017 Apr 20;2(8).

[6] Chen J, Jiang X, Sun T, He P, Wang S. Detecting double MPEG compression with the same quantiser scale based on MBM feature. In2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2016 Mar 20 (pp. 2064-2068). IEEE.

[7] Mahalakshmi I, Mahendran S. Analysis of Video Compression Technique. International Journal of Advanced Research in Computer Science & Technology (IJARCST). 2017 Jan;5:52-5.

[8] Amada S, Kamamoto Y, Harada N, Sugiura R, Moriya T, Makino S, Yamada T. Experimental evaluation of encoding parameters of MPEG-4 ALS for high-resolution audio. In2017 IEEE 6th Global Conference on Consumer Electronics (GCCE) 2017 Oct 24 (pp. 1-4). IEEE.

[9] Grois D, Nguyen T, Marpe D. Coding efficiency comparison of av1/vp9, h. 265/mpeg-hevc, and h. 264/mpeg-avc encoders. In2016 Picture Coding Symposium (PCS) 2016 Dec 4 (pp. 1-5). IEEE.