

HD Video Coding based on Fractal and H.264

Yun Chen

Beihang University, Beijing, China

chenyun1121@163.com

Keywords: High-definition video; Video coding; Fractal coding; H.264

Abstract. This paper studies the HD video and remote sensing image compression algorithm based on fractal and H.264. In order to study the technology of image storage and transmission suitable for HD resolution video. The system further optimizes the intra-frame coding mode currently adopted by H.264, and proposes a new fast intra-block block algorithm to reduce the prediction mode selection. For inter prediction, a new motion estimation and motion compensation algorithm is proposed based on fractal. In order to further reduce the number of coded bits, transform, quantize and entropy encode the fractal residuals. Since the motion vector MV and the fractal coefficients cannot be quantized, their differences are directly entropy encoded. The high-definition video coding system based on fractal and H.264 not only reduces the coding complexity, but also improves the coding efficiency.

Introduction

With the continuous advancement of technology and the improvement of video technology, the demand for high quality video is increasing [1-3]. However, how to improve compression efficiency in the case of limited transmission and storage space has become a key to video technology [4-9]. In order to better adapt to the requirements of high-definition video compression, H.264 makes full use of the spatial correlation of video frames to propose a tree-block partitioning mode and multiple prediction modes. However, due to multiple prediction modes in different division modes, the coding time is relatively long, therefore, we first perform a statistic on the different block partitions in the intra prediction of H.264 to facilitate a subsequent related improvement on intra prediction.

In current fractal video coding systems, intraframe coding of I frames is performed, the processing method: First, the I frame is divided into fixed-size blocks, and then each block is subjected to a series of operations such as transformation, quantization, and entropy coding. The code stream formed by this method is encoded, and the decoded video image quality formed after decoding, inverse quantization, and inverse transform at the decoding end is worse than the original image, and the decoded image quality is poor. Because the H.264 encoding effect is relatively better, in order to improve the quality of intraframe coding, this paper draws on the intra prediction method in the video coding standard H.264/AVC and improves it to some extent [10-16].

Block Division Structure

Intra block partition. In this paper, the video sequence to be encoded is divided into consecutive frames, each frame is divided into 16×16 macroblocks of uniform size, and then encoded for each macroblock [18]. First, the macroblock is divided into 16×16 , and then the prediction modes of the four prediction modes are respectively performed, and the rate distortion cost in different prediction modes is calculated; Then divide the current macroblock into $16 \times 4 \times 4$ blocks, perform prediction of nine prediction modes for each block and perform rate distortion cost calculation for each prediction mode, and then calculate the sum of rate distortion costs of the entire 16×16 block; The prediction mode with the lowest rate distortion cost is found in the 16×16 and 4×4 division modes as the final division mode and prediction mode [19]. This paper proposes a fast intra block partitioning algorithm.

(1) The left macroblock A is 4×4 , and the upper macroblock B is also 4×4 , Calculating the rate distortion cost J of various prediction modes in the two block division modes for the current block;

(2) The left macroblock A is 4×4 partition, the upper macroblock B is 16×16 partition, or the left macroblock A is 16×16 partition, and the upper macroblock B is 4×4 partition. Two block partitioning attempts are made to the current block and the rate distortion cost J is calculated for each mode.

(3) The left macroblock A is 16×16 partition, the upper macroblock B is 16×16 partition, and the current block is 16×16 partition, so that each macroblock can reduce the 144 (16×9) rate distortion cost calculation.

The calculation process of the rate distortion cost is as follows:

(1) Calculate the prediction residual of the current block.

$$Diff(x, y) = Orig(x, y) - Pred(x, y) \quad (1)$$

Among them, $Orig(x, y)$ and $Pred(x, y)$ re the pixel values of the original block and the prediction block respectively.

(2) The Hadamard transform is performed on the prediction residual, and the absolute value and SATD after the residual transformation are calculated, that is

$$SATD = \sum_{x,y} |HAD(Diff(x, y))| \quad (2)$$

(3) The SATD cost is calculated using a simplified SATD cost function.

$$J = SATD + \lambda_{mode} \cdot R_{mode} \quad (3)$$

Among them, λ_{mode} is a Lagrange multiplier, R_{mode} is the bit value encoded in the current intra mode.

Fractal coding

Fast motion estimation algorithm. Inter-frame prediction generally uses the time-domain correlation of video frames, using motion estimation and motion compensation to find the best matching block [20]. Since the amount of full search calculation in the search range space is too large, which is not conducive to improving coding efficiency, we propose a new fast motion estimation algorithm. The motion vectors of adjacent blocks are utilized to calculate the initial search points of the current block, then the fast motion estimation algorithm is carried out based on the initial search points, the best matching block is calculated by fractal coding, finally, the residual is transformed by integer DCT, quantized and entropy coding [21, 22]. In the interframe predictive coding mode, each block will have one that needs to be encoded. When a small-sized blocking mode is used, for example, if a macroblock is completely divided into 4×4 blocks, 16 encodings are required; An MV contains four variables, motion vectors x, y, s, and o, x and y are motion vectors, s and o are fractal parameters. If the MV is not compressed, the number of encoded bits of the MV may be larger than the prediction residual. The MV of the current block is predicted by the correlation between adjacent blocks MV and the difference between the predicted MV_p and the actual MV is encoded:

$$MVD = MV - MV_p \quad (4)$$

As shown in Figure 1, it is predicted by the left A, the upper B and the upper right block C (arbitrary size) of the current block. Suppose E is the current block, and A is the block to which the uppermost 4×4 neighbor block belongs to the left side of E. B is the block to which the leftmost 4×4 neighboring block above E belongs, and C is the block to which the diagonal upper 4×4 neighboring block belongs to the upper right corner of E.

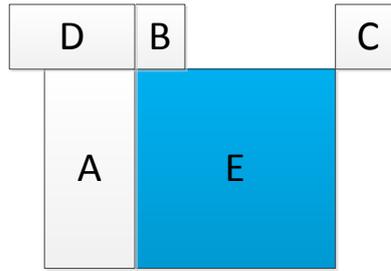


Figure 1 predicted neighbor block position

The prediction rules are as follows:

- (1) In addition to the 16x8 and 8x16 blocks, the MV_p takes the median of the A, B, and C block MV.
- (2) In the 16x8 block, the upper part of the MV_p is predicted by B, and the part MV_p below the MV is predicted by A.
- (3) In the 8x16 block, the left part MV_p is predicted by A, and the right part MV_p is predicted by B.
- (4) When A, B, and C are not available, they are directly encoded.
- (5) When C is not available, if D is available in the upper left corner, replace it with D.
- (6) When only one of A, B, and C is available, the MV_p is equal to the MV of the block.
- (7) When two blocks are available, the block of the unavailable block is counted as 0, and the median of the three blocks is still taken.

Based on the acquired coordinates of the current block, the starting search point coordinates of the motion estimation are calculated. Based on the starting search point, the fast motion estimation algorithm is used to calculate the corresponding matching error.

The process of the fast loop cross search algorithm as follows:

Step 1: Perform a ring search in (-2, -2) to (2, 2) starting from the initial search point, and perform fractal matching calculation for each position. If the best match is in the range of (-1, -1) to (1, 1), skip to step 3, otherwise skip to step 2;

Step 2: In the range of 2 in the ring cross search algorithm, select the two positions with the smallest matching error, and perform cross search for the two positions respectively, select the least cost location in all four neighborhoods as its final, skip to step 3.

Step 3: Stop this search and continue the motion estimation of the next block.

Fractal matching. The purpose of motion search is to find the best matching block of the current block in the search range, which requires a matching criterion to minimize the error [17, 20]. For fractal matching errors, the formula is:

$$diff = \sum_{i=1}^N |r_i - [s \cdot d_i + o]| \tag{3.5}$$

r_i is the pixel value of the current block, and d_i is the predicted block pixel value corresponding thereto N is the number of pixels in the current block, s is the scale factor, and o is the offset factor.

$$\begin{cases} s = \frac{N(\sum_{i=1}^N r_i d_i) - \sum_{i=1}^N r_i \sum_{i=1}^N d_i}{N \sum_{i=1}^N d_i^2 - (\sum_{i=1}^N d_i)^2} \\ o = \frac{1}{N} (\sum_{i=1}^N r_i - s \sum_{i=1}^N d_i) \end{cases} .$$

Specific Implementation Steps

The specific steps of the fractal video and H.264-based HD video encoding process proposed in this paper are as follows:

Step 1: If the current frame is an I frame, first perform a fast block division decision block mode for each macro block, and then perform various prediction modes corresponding to the block size for prediction. The actual block is compared with the prediction block, the prediction residual is calculated, and then the residual is transformed, quantized, reordered, and entropy encoded. At the same time, the quantized residual coefficients are inverse quantized and inversely transformed, and reconstructed according to the block division mode and the prediction mode. Performing loop filtering on the reconstructed block, then using the filtered block as a reference block of the subsequent macroblock, and then performing encoding of the next macroblock until all macroblocks of the current frame are encoded;

Step 2: If the current frame is a non-I frame, inter prediction is performed for each macro block, firstly, the distribution range of several consecutive frames before the judgment. If the ratio of MV in $(-2, -2) \sim (2, 2)$ exceeds the threshold, the fast motion estimation mode is entered, otherwise the full search mode is executed. Perform fractal coding, calculate fractal parameters and residuals, perform variation quantization on residuals, etc. Because the fractal coefficients cannot be quantized, the fractal coefficients are encoded directly, and then the next macroblock is encoded until all the macroblocks in the current frame are encoded;

Step 3: If all frames have been encoded, stop it; otherwise skip to step one.

Experimental Result

The experimental platform of this paper is Intel(R) Core(TM) i7-3770 CPU@3.40 GHz, the memory is 8 GB, and the programming environment is Visual Studio 2008. Simulation experiments were performed on the first 50 frames of different resolution video sequences. In order to evaluate the performance of the proposed method for compressing high-definition video sequences, the experimental results were compared with the reference software JM18.5 of the video coding standard H.264.

Compared with JM coding performance.

In order to verify the performance of the algorithm, the video is compressed for different resolutions, and compared with JM18.5 to verify the compression effect of 50 frames simultaneously under different QP. Table 1 is the result of encoding the video (Flowervase_832×480). In terms of coding efficiency, the system has an average reduction of 55.9% with respect to the JM18.5 encoding time; In the decoding video quality evaluation, the PSNR of the system is increased by 0.58dB relative to JM18.5; Compared with JM18.5, the proposed algorithm obtains a lower PSNR when the code rate is lower, but the algorithm is better than JM18.5 when the code rate is high.

Table 1 Flowervase_832×480

Algorithms		QP					
		20	24	28	32	36	40
PSNR(dB)	JM18.5	46.79	44.11	41.31	38.28	35.36	32.63
	Proposed	46.80	44.26	41.38	38.72	36.44	34.36
	Increment	+0.01	+0.15	+0.07	+0.44	+1.08	+1.73
TOL_TIME (sec)	JM18.5	344.9	342.9	335.5	335.4	327.2	324.3
	Proposed	148.0	150.8	146.9	142.8	151.2	146.4
	Increment	-57.1%	-56.0%	-56.2%	-57.4%	-53.8%	-55.1%

Experimental results. It can be seen from the above experimental data that the decoded image can maintain a higher quality than the original image. For the test sequence, the image quality of JM

is better when the QP is smaller, and the PSNR of the algorithm is higher when the QP is at the middle or larger value; In coding time, the coding time of this algorithm is significantly lower than that of JM; when the code rate is high, the PSNR obtained by this algorithm is higher. The experimental results prove that: Compared with the international video coding standard H.264/AVC reference software JM18.5, the proposed video coding system has certain advantages in coding speed and compression efficiency.

Summary

The intraframe fast block partitioning algorithm proposed in this paper greatly reduces the computational complexity of the intra prediction algorithm. For inter-frame coding, it can use fast motion estimation algorithm when the MV is within a certain range. Otherwise, it can adopt full search algorithm and adopt fractal coding, which can ensure the coding quality and improve the coding efficiency. In order to further reduce the number of coded bits to transform, quantize and entropy the fractal residuals, because the motion vector MV and fractal coefficient cannot be quantified, entropy is directly encoded on the difference. The high-definition video coding system based on fractal and H.264 not only reduces the coding complexity, but also improves the coding efficiency.

References

- [1] Richardson, Iain EG, H.264/mpeg-4 part 10 white paper, Mar. 2003 [Online] Available: <http://www.vcodex.com>
- [2] Puri, A., Chen, X, Luthra, A, Video coding using the H. 264/MPEG-4 AVC compression standard, *Signal processing: Image communication*, Vol.19 (2004) No. 9, p.793.
- [3] Cote, G., Erol, B., Gallant, M., Kossentini, F. H., 263+: Video coding at low bit rates, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8 (1998) No. 7, p. 849.
- [4] Shiping Zhu and Xianzi Zong, Fractal lossy hyperspectral image coding algorithm based on prediction, *IEEE Access*, Vol. 5 (2017), p.21250.
- [5] Shiping Zhu and Ziyao Xu, Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network, *Neurocomputing*, Vol. 275 (2018), p.511.
- [6] Acharjee, S., Biswas, D., Dey, N., Maji, P. and Chaudhuri, S.S., An efficient motion estimation algorithm using division mechanism of low and high motion zone, *IEEE International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, (2013), p. 169.
- [7] Wang M., Liu R., Lai C. H., Adaptive partition and hybrid method in fractal video compression, *Computers & Mathematics with Applications*, Vol. 51 (2006) No. 11, p. 1715.
- [8] Shiping Zhu, Shupeizhang, and Chenhao Ran, An improved inter-frame prediction algorithm for video coding based on fractal and H.264, *IEEE Access*, Vol. 5 (2017), p.18715.
- [9] Shiping Zhu and Chunyan Zhang, A fast algorithm of intra prediction modes pruning for HEVC based on decision trees, *Multimedia Tools and Applications*, Vol. 76 (2017) No. 20, p.21707.
- [10] Lazar M. S., Bruton L. T., Fractal Block Coding of Digital Video, *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 4 (1994) No. 3, p.297.
- [11] Lai C M, Lam K M, Siu W C., A fast fractal image coding based on kick-out and zero contrast conditions, *IEEE Transactions on Image Processing*, Vol. 12 (2003) No. 11, p.1398.
- [12] Shiping Zhu, Ruidong Gao and Zheng Li, Stereo matching algorithm with guided filter and modified dynamic programming, *Multimedia Tools and Applications*, Vol. 76 (2017) No. 1, p.199.
- [13] Farkade, S.G., Kamble, S.D., A hybrid block matching motion estimation approach for fractal video compression, *Circuits Power and Computing Technologies (ICCPCT)*, (2013), p. 1151.
- [14] Shiping Zhu, Dongyu Zhao and Liyun Li, Adaptive fast intra prediction for high efficiency video coding, *Multimedia Tools and Applications*, Vol. 75 (2016) No. 13, p. 7559.

- [15] Chaudhari, R.E., Dhok, S.B., Acceleration of fractal video compression using FFT, *15th International Conference on Advanced Computing Technologies (ICACT)* (2013), p. 1.
- [16] Huang, Z., Frame-groups based fractal video compression and its parallel implementation in Hadoop cloud computing environment, *Multidimensional Systems and Signal Processing*, Vol. 29 (2018) No. 3, p.961.
- [17] Shiping Zhu, Yangshuan Hou, Zaikuo Wang, et al, Fractal video sequences coding with region-based functionality, *Applied Mathematical Modelling*, Vol. 36 (2012) No. 11, p.5633.
- [18] Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A. Overview of the H. 264/AVC video coding standard, *IEEE Transactions on circuits and systems for video technology*, Vol.13 (2007) No.7, p.560.
- [19] Shiping Zhu, Xiaoyan Bu, and Qin Zhou, A novel edge preserving active contour model using guide filter and harmonic surface function for infrared image segmentation, *IEEE Access*, Vol. 6 (2018), p.5493.
- [20] Chaudhari, Ravindra E.; Dhok, Sanjay B., Fractal video coding using fast normalized covariance based similarity measure, *Mathematical Problems in Engineering*, (2016) No. 1725051, p.1.
- [21] Song-Hak Ri, Yuri Vatis and Joern Ostermann, Fast inter-mode decision in an H.264/AVC encoder using mode and Lagrangian cost correlation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19 (2009) No. 2, p. 302.
- [22] D. S. Turaga and T. Chen, Classification based mode decisions for video over networks, *IEEE Transactions on Multimedia*, Vol. 3, (2001) No. 1, p. 41.