Design on the Fly Encryption Algorithm Video Compression Standard

1Ramalakshmi.S, 2Rajagopal. S
1, 2 National Engineering College

Abstract— Video Encryption is done at the encoding stage to achieve On The Fly encryption. On The Fly Encryption means encryption is done during the Compression stage. Motion Vector is one of the key elements in the Motion Estimation process. First the video are converted into Frames. To estimate the Motion Vector, Block Matching Algorithm was used. Block matching algorithm is used to increase the chance to find the true motion vector and reduce the computational requirements and also it was simple and effective for motion vector calculation. Full Step search block matching algorithm was used to find the best candidate block. These motion vector values are encrypted by using zig-zag permutation algorithm. The perceptual feature requires that the quality of visual data be partially degraded by encryption. Then these encrypted values are given as input to coding stage. Finally it provide the Encrypted and Encoded Bit streams.

Key Terms—CAVLC, H.264/AVC, Zigzag Scan, Coefficient Token, CABAC

I. INTRODUCTION

H.264 is the most widely-deployed video compression system. The H.264 standard has also been extended to allow scalable video coding papers. In the past decade, with the rapid development of digital video and network technology, a series of video compression standards have been proposed to meet the increasing requirements of video applications. At the same time, a significant amount of research has been carried out on protecting the video stream. Selective encryption is one of the most promising techniques for practical applications as it can meet real-time processing requirements and provide an effective perceptual scrambling effect with no or very minimal impact on the compression performance. The latest video compression standard, H.264/AVC (Part 10 of MPEG-4), was collaboratively issued by the joint video team (JVT) of ISO/IEC MPEG and ITU-T VCEG in 2003, and was recently updated by JVT. From the point of view of compression performance, it has been reported that H.264/AVC can significantly outperform previous standards. H.264/AVC offers two entropy coding methods to encode nonzero quantized transform coefficients, CAVLC (Context-Adaptive Variable Length Coding) and CABAC (Context-Adaptive Binary Arithmetic Coding). For CAVLC, it has no effect on the compression ratio since it does not change the bit length that represents the intra prediction mode; for CABAC, the compression ratio is only very slightly affected.

II. LITERATURE SURVEY

Z. Shahid et al (2009), proposed the Fast Protection of H.264/AVC by Selective Encryption[21]. In this paper, proposes a new method for the protection of copyrighted multimedia data. Selective Encryption was performed in the CAVLC module. In this paper proposes, entropy coding scheme serves the purposes of encryption step without affecting the coding efficiency of H.264/AVC by keep the bit rate unchanged and generating compliant bit stream. A. Alattar, et al (1999), proposed an Improved Selective Encryption Techniques for Secure transmission on MPEG Video Bit Streams[23]. In this paper, the author has proposed a three new “Selective Encryption Techniques for Secure transmission on MPEG Video Bit Streams”. By applying encryption to only the data associated with the I frames, it reduce the encode/decode processing time of the video bit stream and also the predicted frames (P and B frames) will inherit their security of I frames. So this method does not provide an adequate level of security especially for high sensitivity videos. Zafar Shahid, et al (2011), proposed a Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I and P frames[22]. In this paper the author has proposed a novel method for the protection of bitstreams. The problem of the selective encryption is addressed along with the compression in the entropy coding modules. In this paper, the encryption step is done simultaneously with the entropy coding CAVLC or CABAC. For CAVLC, SE is performed on equal length codewords. For CABAC, SE is done on equal length bit strings. The proposed Selective Encryption algorithm provide the encryption method doesnot affect the bit rate and the H.264/AVC bit stream compliance. W. Zeng et al (2003), proposed an Efficient Frequency Domain Selective Scrambling of Digital Video [20]. In this paper the author has proposed Multimedia data security is very important for multimedia commerce on the internet such as video, Real time video multicast. Traditional Cryptography algorithm is not provide the data security are not fast enough to process the vast amount of data generated by the multimedia application to meet the real time constraints. So, we use Selective Encryption compression in
which video data are scrambled efficiently. This approach is very easy to implement, provides the better security, different levels of transparency, have limited compression efficiency and no impact on error resiliency. Thomas Wiegand et al (2003), proposed an Overview of the H.264/AVC Video Coding Standard [2], H.264/AVC is newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a “network-friendly” video representation addressing “User friendly” (video telephony) and “non user friendly” (storage, broadcast, or streaming) applications. H.264/AVC was attaining a significant improvement in rate-distortion Efficiency relative to existing standards. H.264/AVC represents a number of advances in standard video coding technology, in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application domains. Z.Wang et al (2004), proposed an Image Quality Assessment From Error Visibility to Structural Similarity[25]. Most existing approaches are known as full-reference, meaning that a complete reference image is assumed. In many practical applications, however, the reference image is not available, and a no-reference or “blind” quality assessment approach is desirable. In this type, the reference image is only half available, in the form of a set of extracted features made available as side information to help evaluate the quality of the distorted image. Detlev Marpe et al (2003), proposed a Context Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard[18]. Context-Based Adaptive Binary Arithmetic Coding (CABAC) as a normative part of the new ITU-T/ISO/IEC standard H.264/AVC for video compression was presented. By combining an adaptive binary arithmetic coding technique with context modeling, a high degree of adaptation and redundancy reduction is achieved. The CABAC framework also includes a novel low-complexity method for binary arithmetic coding and probability estimation that is well suited for efficient hardware and software implementations. Jiangto Wen et al (2001), proposed a Format Compliant Configurable Encryption Framework for Access Control of Multimedia[17]. One of the major goals of content access control for entertainment purposes is to disallow unauthorized users to view the video with satisfactory quality, similar to a set top cable TV scrambler. One common method for access control is through encryption. The basic idea of the approach is to encrypt only certain fields of the compressed bit stream in a manner that allows the encrypted bit stream to maintain format compliance, but to have unacceptable quality for the target application without decryption of these fields. Feng Dai et al (2011), proposed the Restricted H.264/AVC Video Coding for Privacy Protected Video Scrambling[26]. Privacy region scrambling is an effective method to protect privacy in video. To ensure that non privacy regions are not affected by scrambling, methods should be taken to intercept drift error in privacy protected video scrambling. However, existing methods have significantly reduced the coding efficiency. In this paper, focus on improving coding efficiency while preventing drift error in privacy protected H.264/AVC video scrambling, which is the state-of-the-art coding standard. Ci Wang et al (2003), proposed the A DCT based MPEG-2 Transparent Scrambling Algorithm[24]. Recently, several encryption algorithms have been proposed, which are applied to MPEG-2 video streams. These algorithms try to optimize the encryption process and improve encryption speed by exploiting the spatial and temporal properties of video retrieval and display process. As a new branch of encryption, transparent scrambling has its merits and special prospects. Many video service operators do not always want their programs totally unrecognizable to unauthorized receivers. They would rather permit them to view sufficiently degraded programs, since the viewers may become future customers. To satisfy this demand, a scrambling-degree controllable algorithm is needed. Yi Ming Lin et al (2006), proposed An Efficient Implementation of CAVLC for H.264/AVC[19]. In H.264/AVC, the context-based adaptive variable length coding (CAVLC) is used for lossless compression. Direct table-lookup implementation requires higher cost because it employs a larger memory to produce the encoded results. A carefully reduced table is constructed and used for encoding in the design. With the helps of symbol mapping and codeword generating modules encoder can generate the CAVLC bit stream accurately. Compared with those previous designs, this method reduces up to 81% of memory size for storing those VLC tables used in H.264/AVC. Thus, this design is a good candidate for low-cost.

In this paper, first the videos are converted into frames. To find Motion Vector we use the Block Matching Algorithm(BMA). The main Objective of Block Matching Algorithm is to find a candidate block in the search region best matched to the source block. There are several BMA are presented. Here Full Step Search BMA was used. Then the motion values are converted into 4*4 matrix by the use of Discrete Cosine Transform, then the values are Quantized by using Quantization, then the values are encrypted by the use of Zigzag Permutation Based Encryption. These encrypted values are given to the input of CAVLC.

In CAVLC there are 5 parameters are present. The matrix values are first scanned using Zigzag scan, then the reordered matrix are generated. After that the 5 parameters are separated and gives the output as the encoded bitstream.

III. PROPOSED SCHEME

Figure 1 shows the Block Diagram of the proposed scheme. In the proposed scheme, the input is given as video. First the video was converted into frames. To find the difference between one frame to another frame we use the Motion Vector. To find Motion Vector we use the Block Matching Algorithm. The main objective of Block Matching Algorithm is to find a candidate block in the search region best matched to the source block. Here Full Step Search BMA was used. Full search (FS) is the optimal algorithm for BMME After that the values are Transform and Quantized. The values are encrypted by the use of Permutation Based Encryption. These encrypted values are given as input to CAVLC.
Adaptive Root pattern search

C. Comparisons of Block Matching Algorithm:

1. Full Search Method

In the Full Search method the search algorithm evaluates all positions in the window search of \((2W+1)\times(2W+1)\) size. It involves high Computational cost. Its is simple and it guarantees a high accuracy in finding the best match.

II. Three Step Search

Three step search is one of the earliest attempts in block matching algorithms. The general idea behind the Block Matching Algorithm for Three Step Search is that it starts with the search location in the center and set the step size as \(S=4\). It then searches the eight location and center. From these nine search regions it find the lowest cost and find the new search origin. At \(S=1\) it find the best location with minimum cost and find the best match.

III. Simple and Efficient Search

Simple and Efficient Search is another form of the Three Step Search. In the Simple and Efficient Search the search area is divided into 4 phases like A,B,C,D and search the three locations A,B and C. Depending on the lowest cost the search area was fixed. There are some rules for searching the second phase. The rules are

- If \(\text{MAD}(A) < \text{MAD}(B)\) and \(\text{MAD}(A) < \text{MAD}(C)\), select (d);
- If \(\text{MAD}(A) < \text{MAD}(B)\) and \(\text{MAD}(A) = \text{MAD}(C)\), select (b);
- If \(\text{MAD}(A) < \text{MAD}(B)\) and \(\text{MAD}(A) > \text{MAD}(C)\), select (c);
- If \(\text{MAD}(A) > \text{MAD}(B)\) and \(\text{MAD}(A) < \text{MAD}(C)\), select (d);
- If \(\text{MAD}(A) > \text{MAD}(B)\) and \(\text{MAD}(A) > \text{MAD}(C)\), select (e).

IV. Four Step Search

Four Step Search was similar to the New Three Step Search. Four Step Search sets the fixed pattern size of \(S=2\). It also search the 9 locations and find the least weight around the center and jump to the fourth step. If the least weight is found at anyone of the eight locations except center then that lowest weight is search origin for the next step. The location with the lowest weight is the best matching macroblock.

V. Diamond Search

Diamond Search is exactly same as the Four Step Search, but the search point is changed as Square to Diamond. In the Diamond Search method there was no search point. In Diamond Search method there are two types of patterns 1. Large Diamond Search Pattern 2. Small Diamond Search Pattern.

VI. Adaptive Root Pattern Search

In the Adaptive Root Pattern Search the frames are converted into macroblocks. If the macroblocks along with the current macroblocks moved in the particular direction means these two macroblocks are having the same motion vector. This algorithm uses the motion vector of the macroblock to its immediate left to predict its own motion vector.

D. Full Step Search algorithm

Block-matching motion estimation is an efficient algorithm for reducing the temporal redundancy in video
coding and is adopted by video coding standards. Many fast block-matching algorithms have been devised to reduce the computational complexity without degrading the estimation quality. Block matching motion estimation (BMME) plays a very important role in video coding. The performance of BMME greatly affects the quality of the encoded bitstream. Full search (FS) is the optimal algorithm for BMME. Full search block matching algorithm (FSBMA) for motion estimation requires a high level of computations. Full Search Algorithm without losing too much quality at the output. Full step search gives the best psnr among all algorithms and average search time faster than the all algorithms. FSBM used for different standard video sequences.

In order to get the best match block in the reference frame, it is necessary to compare the current block with all the candidate blocks of the reference frames. Full search motion estimation calculates the sum absolute difference(SAD) value at each possible location in the search window. Full search computed the all candidate blocks intensive for the large search window.

Consider a block of \( N \times N \) pixels from the candidates frame at the coordinate position \((r, s)\) as shown and then consider a search window having a range \( \pm w \) in both and directions in the references frame, as shown. For each of the \((2w + 1)2\) search position (including the current row and the current column of the reference frame), the candidate block is compared with a block of size \( N \times N \) pixels, and the best matching block, along with the motion vector is determined only after all the \((2w+1)2\) search position are exhaustively explored. By exhaustively testing all the candidate blocks within the search window, this algorithm gives the global minimum block distortion position which corresponds to the best matching block. However, a substantial computation load is demanded. Normally ME is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search (FS) is used by exhaustively evaluating all possible candidate blocks within the search window.

E. Transform

In H.264/AVC Video Ompression standard uses the Discrete Cosine Transform. In the DCT Transform the frames are converted into into \( 4 \times 4 \) blocks. The following matrix are defines the size of the \( 4 \times 4 \) DCT transform

\[
A = \begin{pmatrix}
0.5 & 0.5 & 0.5 & 0.5 \\
0.6532 & 0.2706 & -0.2706 & -0.6532 \\
0.5 & -0.5 & -0.5 & 0.5 \\
0.2706 & -0.6532 & 0.6532 & -0.2706
\end{pmatrix}
\]

The transform is used to facilitate frequency-based compression techniques. The human eye is more sensitive to the information contained in low frequencies than to the information contained in high frequencies. Therefore, the DCT helps separate the more perceptually significant information from less perceptually significant information.

F. Quantization

Quantization in image processing is a process that attempts to determine what information can be discarded safely without a significant loss in visual fidelity.

In quantization, involved in image processing, is a lossy compression technique achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible.

G. Zigzag based Permutation Encryption

The ZigZag-Permutation algorithm is also a selective encryption algorithm. The idea here is to use a permutation of the zig-zag ordering of DC and AC coefficients in the DCT transformation of \( 8 \times 8 \) blocks in the MPEG or JPEG format. The video encryption algorithms in this class mainly use different permutation algorithms to scramble or encrypt the video contents. It is not necessary to scramble each and every byte.

IV. CAVLC AND ITS MODULES

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. In entropy Coding there are two types of methods are available. They are 1.CAVLC and 2.CABAC. CAVLC is easier to decode than CABAC. CAVLC supports in all higher profiles.

Context-adaptive variable-length coding (CAVLC) is a form of entropy coding used in H.264/MPEG-4 AVC video encoding. It is an inherently lossless compression technique, like almost all entropy-coders. In H.264/MPEG-4 AVC, it is used to encode residual, zig-zag order, blocks of transform coefficients. It is an alternative to context- based adaptive binary arithmetic coding (CABAC). CAVLC requires considerably less processing to decode than CABAC.
CAVLC is used to encode residual, zig-zag ordered 4x4 (and 2x2) blocks of transform coefficients. CAVLC is designed to take advantage of several characteristics of quantized 4x4 blocks:
1. After prediction, transformation and quantization, blocks are typically sparse (containing mostly zeros).
2. The highest non-zero coefficients after zig-zag scan are often sequences of +/- 1. CAVLC signals the number of high-frequency +/-1 coefficients in a compact way.
3. The number of non-zero coefficients in neighboring blocks is correlated. The number of coefficients is encoded using a look-up table; the choice of look-up table depends on the number of non-zero coefficients in neighboring blocks.
4. The level (magnitude) of non-zero coefficients tends to be higher at the start of the reordered array (near the DC coefficient) and lower towards the higher frequencies.

In CAVLC, there are 5 syntax elements present. Five parameters of each 4x4 block are coded separately. They are 1. Coefficient token 2. Trailing ones 3. Remaining non-zero Coefficient 4. Total Zeros Encoder 5. Each run of Zeros.

First the input video was given as input, then the video are converted into frames. To find the difference between one frame to another we use the Block Matching Algorithm. Block Matching Algorithm is a way to locating matching blocks in a sequence of digital video frames for the purposes of Motion Estimation. This can be used to discover temporal redundancy in the video sequence, increasing the effectiveness of interframe video compression. Then the motion vector values are transformed by the use of DCT transform. After that the values are quantized by the use of quantization. These quantised values are encrypted by the Permutation Encryption. These encrypted values are given as input to the CAVLC.

In CAVLC there are 5 syntax elements are present. They are 1. Coefficient token 2. Trailing ones 3. Remaining non-zero Coefficient 4. Total Zeros Encoder 5. Each run of Zeros.

In CAVLC, the 5 modules are separately encoded. Finally it produce the encrypted and encoded bitstreams.

IV a. Modules of CAVLC

In CAVLC scanning, the Quantized Coefficients of a block are grouped into nonzero coefficients which are encoded separately. The presente modules of CAVLC are

A. Coefficient token:

Coeff_token, encodes both the total number of non-zero coefficients (TotalCoeffs) and the number of trailing +/-1 values (T1). TotalCoeffs can be anything from 0 (no coefficients in the 4x4 block) 1 to 16 (16 non-zero coefficients). T1 can be anything from 0 to 3; if there are more than 3 trailing +/-1s, only the last 3 are treated as “special cases” and any others are coded as normal coefficients. There are 4 choices of look-up table to use for encoding coeff_token, described as Num_VLC0, Num_VLC1, Num_VLC2 and Num_VLC3 (3 variable-length code tables and a fixed-length code). The choice of table depends on the number of non-zero coefficients in upper and left-hand previously coded blocks Nu and NL. A parameter N is calculated as follows 1. If blocks U and L are available (i.e. in the same coded slice), \( N = (N_u + N_l)/2 \) 2. If only block U is available, \( N = N_u \); if only block L is available, \( N = N_l \); if neither is available, \( N = 0 \).

B. Trailing Ones:

For each T1 (trailing +/-1) signalled by coeff_token, a single bit encodes the sign (0=+, 1=-). These are encoded in reverse order, starting with the highest-frequency T1.

C. Remaining Nonzero Coefficients:

The level (sign and magnitude) of each remaining non-zero coefficient in the block is encoded in reverse order, starting with the highest frequency and working back towards the DC coefficient. The choice of VLC table to encode each level adapts depending on the magnitude of each successive coded level (context adaptive).

There are 7 VLC tables to choose from, Level_VLC0 to Level_VLC6. Level_VLC0 is biased towards lower magnitudes; Level_VLC1 is biased towards slightly higher magnitudes and so on. The choice of table is adapted in the following way: (a) Initialize the table to Level_VLC0 (unless...
there are more than 10 non-zero coefficients and less than 3 trailing ones, in which case start with Level_VLC1).
(b) Encode the highest-frequency non zero coefficient.
(c) If the magnitude of this coefficient is larger than a pre-defined threshold, move up to the next VLC table.

D. Total Number of Zeros:

Total Zeros is the sum of all zeros preceding the highest non-zero coefficient in the reordered array. This is coded with a VLC. The reason for sending a separate VLC to indicate TotalZeros is that many blocks contain a number of non-zero coefficients at the start of the array and this approach means that zero-runs at the start of the array need not be encoded.

E. Each Run of Zeros:

The number of zeros preceding each non-zero coefficient (run_before) is encoded in reverse order. A run_before parameter is encoded for each non-zero coefficient, starting with the highest frequency, with two exceptions.
(a) If there are no more zeros left to encode (i.e. \(\sum \text{run\_before} = \text{TotalZeros}\)), it is not necessary to encode any more run_before values.
(b) It was not necessary to encode run_before for the final non-zero coefficient.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this proposed work, First the video was converted into frames. Then to find the difference between one frames to another we use the motion vector. Motion Vector is the key element of the Motion Estimation process. Block Matching algorithm is used to find the best match among all the blocks. Here Full Step Search block matching algorithm are used. Then the motion vector values are transformed, quantised and encrypted by the Zigzag permutation encryption. These encrypted values are given as input to the CAVLC. In CAVLC the syntax elements are separately encoded. Finally we get the encrypted and encoded bitstreams.
VI. REFERENCES


