CONTEXT BASED ADAPTIVE VARIABLELENGTH CODING FOR H.264 ENCODER

P. Sangeetha
Faculty, Department of Electronics
Engg. RCERT Chandrapur, India

Anjum Sheikh
Faculty ,Department of Electronics
Engg. RCERT Chandrapur,India

M.R.Bhongade
HOD Department of Electronics Engg.RCERT
Chandrapur,India

ABSTRACT
Video compression it makes it possible to use digital video in transmission and storage environments that would not support uncompressed ('raw') video. Most practical video compression techniques are based on lossy compression, in which greater compression is achieved with the penalty that the decoded signal is not identical to the original. The goal of a video compression algorithm is to achieve efficient compression whilst minimising the distortion introduced by the compression process.

Keywords
MPEG4, video compression, H.264 CAVLC Algorithm,entropy,coding

1. INTRODUCTION
Video compression (video coding) is the process of compacting or condensing a digital video sequence into a smaller number of bits. 'Raw' or uncompressed digital video typically requires a large bit rate (approximately 216 Mbits for 1 second of uncompressed TV-quality video) and compression is necessary for practical storage and transmission of digital video. Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder). The encoder converts the source data into a compressed form (occupying a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data. The encoder/decoder pair is often described as a CODEC (enCOder/ DECoder).

A compressed sequence consists of coded motion vector parameters, coded residual coefficients and header information. The video decoder reconstructs a video frame from the compressed bit stream. The coefficients and motion vectors are decoded by an entropy decoder after which the spatial model is decoded to reconstruct a version of the residual frame. The decoder uses the motion vector parameters, together with one or more previously decoded frames, to create a prediction of the current frame and the frame itself is reconstructed by adding the residual frame to this prediction. Video compression has two important benefits. First, it makes it possible to use digital video in transmission and storage environments that would not support uncompressed ('raw') video.For example, current Internet throughput rates are insufficient to handle uncompressed video in real time (even at low frame rates and/or small frame size). A Digital Versatile Disk (DVD)can only store a few seconds of raw video at television-quality resolution and frame rate and so DVD-Video storage would not be practical without video and audio compression. Second, video compression enables more efficient use of transmission and storage resources. If a high bitrates transmission channel is available, then it is a more attractive proposition to send high-resolution compressed video or multiple compressed video channels than to send a single, low-resolution, uncompressed stream. Even with constant advances in storage and transmission capacity, compression is likely to be an essential component of multimedia services for many years to come.

The goal of a video compression algorithm is to achieve efficient compression whilst minimising the distortion introduced by the compression process. Video compression algorithms operate by removing redundancy in the temporal, spatial and/or frequency domains. The H.264 and MPEG-4 Visual standards share a number of common features. Both standards assume a CODEC 'model' that uses block-based motion compensation, transform, quantisation and entropy coding. The new video standard known as H.264/AVC presents a rich collection of state-of-the-art video coding capabilities that can provide interoperable video broadcast or communication with degrees of capability that far surpass those of prior standards.

2. H.264 CAVLC ALGORITHM
CAVLC algorithm is designed to take advantage of the characteristic transformed and quantized 4x4 blocks. After transformation and quantization, blocks typically contain zeros. The highest non-zero coefficients after the zig-zag scan are often sequences of +/-1. The number of non-zero coefficients in neighboring blocks is correlated. The non-zero coefficients tend to be higher near the DC coefficient and lower towards the higher frequencies. CAVLC encoding of a block of transform coefficients proceeds as follows

A. Encode the number of coefficients and trailing ones (coeff_token).

The first VLC, 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

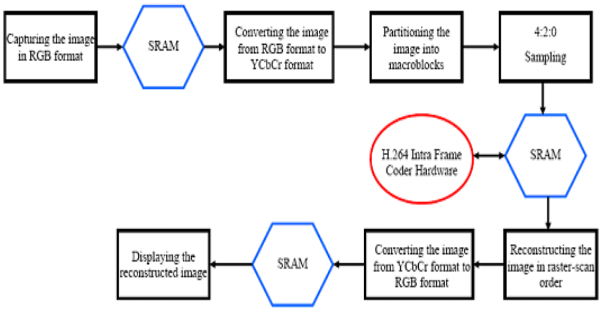


Fig.1 H.264/MPEG4 part 10 Encoder Block Diagram

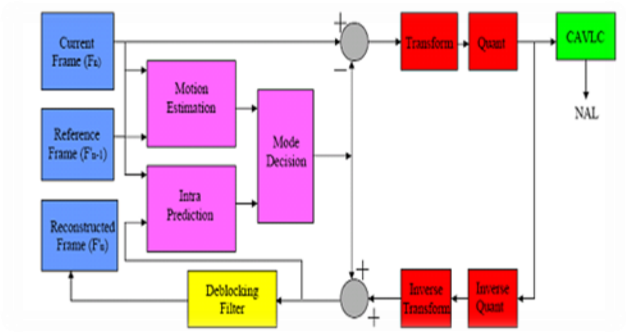


Fig.2 H.264/MPEG4 part 10 Encoder Block Diagram

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-FLC (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:If blocks U and L are available (i.e. in the same coded slice), N = (Nu + NL)/2.If only block U is available, N=NU ; if only block L is available, N=NL ; if neither is available, N=0. N selects the look-up table (Table I) and in this way the choice of VLC adapts depending on the umber of coded coefficients in neighbouring blocks (**context adaptive**). Num-VLC0 is "biased" towards small numbers of coefficients; low values of TotalCoeffs (0 and 1) are assigned particularly short codes and high values of TotalCoeff particularly long codes. Num-VLC1 is biased towardsmedium numbers of coefficients (TotalCoeff values around 2-4 are assigned relatively short codes),Num-VLC2 is biased towards higher numbers of coefficients and FLC assigns a fixed 6-bit code to every value of TotalCoeff.

TABLE I

CHOICE OF LOOK-UP TABLE FOR COEFF_TOKEN

N	**Table for coeff_token**
0, 1	Num-VLC0
2, 3	Num-VLC1
4, 5, 6, 7	Num-VLC2
8 or above	FLC

B. Encode the sign of each T1.

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. Encode the levels of the remaining non-zero 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:

1) Initialise 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).

2) Encode the highest-frequency non zero coefficient.

3) If the magnitude of this coefficient is larger than a pre-defined threshold, move up to the next VLC table.In this way, the choice of level is matched to the magnitude of the recently-encoded coefficients. The thresholds are listed in Table II; the first threshold is zero which means that the table is always incremented after the first coefficient level has been encoded.

TABLE II

THRESHOLDS FOR DETERMINING WHETHER TO INCREMENT LEVEL TABLE NUMBER

Current VLC table	Threshold to increment table
VLC0	0
VLC1	3
VLC2	6
VLC3	12
VLC4	24
VLC5	48
VLC6	N/A (highest table)

D. Encode the total number of zeros before the last coefficient**.**

TotalZeros 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 Total Zeros is that many blocks contain a number of non-zero coefficients at the start of the array and (as will be seen later) this approach means that zero-runs at the start of the array need not be encoded.

E. Encode 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. ? [run_before] = TotalZeros), it is not necessary to encode any more run_before values.

(b) It is not necessary to encode run_before for the final (lowest frequency) non-zero coefficient.The VLC for each run of zeros is chosen depending on (a) the number of zeros that have not yet been encoded (ZerosLeft) and (b) run_before. For example, if there are only 2 zeros left to encode,run_before can only take 3 values (0,1 or 2) and so the VLC need not be more than 2 bits long; if there are 6 zeros still to encode then run_before can take 7 values (0 to 6) and the VLC table needs to be correspondingly larger.

3. RESULTS AND DISCUSSIONS

3.1 Quality

In order to specify, evaluate and compare video communication systems it is necessary to determine the quality of the video images displayed to the viewer. Measuring visual quality is a difficult and often imprecise art because there are so many factors that can affect the results.Visual quality is inherently subjective and is influenced by many factors that make it difficult to obtain a completely accurate measure of quality. For example, a viewer's opinion of visual quality can depend very much on the task at hand, such as passively watching a DVD movie, actively participating in a videoconference, communicating using sign language or trying to identify a person in a surveillance video scene. Measuring visual quality using objective criteria gives accurate, repeatable results but as yet there are no objective measurement systems that completely reproduce the subjective experience of a human observer watching a video display.

3.2 Subjective Quality Measurement

1)Factors Influencing Subjective Quality:

Our perception of a visual scene is formed by a complex interaction between the components of the Human Visual System (HVS), the eye and the brain. The perception of visual quality is influenced by spatial fidelity (how clearly parts of the scene can be seen, whether there is any obvious distortion) and temporal fidelity (whether motion appears natural and 'smooth').

However, a viewer's opinion of 'quality' is also affected by other factors such as the viewing environment, the observer's state of mind and the extent to which the observer interacts with the visual scene. A user carrying out a specific task that requires concentration on part of a visual scene will have a quite different requirement for 'good' quality than a user who is passively watching a movie. For example, it has been shown that a viewer's opinion of visual quality is measurably higher if the viewing environment is comfortable and non-distracting (regardless of the 'quality' of the visual image itself). Other important influences on perceived quality include visual attention (an observer perceives a scene by fixating on a sequence of points in the image rather than by taking in everything simultaneously) and the so-called 'recency effect' (our opinion of a visual sequence is more heavily influenced by recently-viewed material than older video material) [2, 3]. All of these factors make it very difficult to measure visual quality accurately and quantitatively.

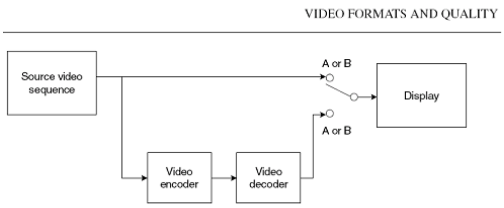


Fig 3. DSCQS Testing System

3.3 ITU-R 500

Several test procedures for subjective quality evaluation are defined in ITU-R Recommendation BT.500-11 [4]. A commonly-used procedure from the standard is the Double Stimulus Continuous Quality Scale (DSCQS) method in which an assessor is presented with a pair of images or short video sequences A and B, one after the other, and is asked to give A and B a 'quality score' by marking on a continuous line with five intervals ranging from 'Excellent' to 'Bad'.

In a typical test session, the assessor is shown a series of pairs of sequences and is asked to grade each pair. Within each pair of sequences, one is an unimpaired "reference" sequence and the other is the same sequence, modified by a system or process under test.

Figure 3 shows an experimental set-up appropriate for the testing of a video CODEC in which the original sequence is compared with the same sequence after encoding and decoding. The selection of which sequence is 'A' and which is 'B' is randomised. The order of the two sequences, original and "impaired", is randomised during the test session so that the assessor does not know which is the original and which is the impaired sequence.

This helps prevent the assessor from pre-judging the impaired sequence compared with the reference sequence. At the end of the session, the scores are converted to a normalized range and the end result is a score (sometimes described as a 'mean opinion score') that indicates the relative quality of the impaired and reference sequences. Tests such as DSCQS are accepted to be realistic measures of subjective visual quality. However, this type of test suffers from practical problems. The results can vary significantly depending on the assessor and the video sequence under test. This variation is compensated for by repeating the test with several sequences and several assessors. An 'expert' assessor (one who is familiar with the nature of video compression distortions or 'artefacts') may give a biased score and it is preferable to use 'nonexpert' assessors. This means that a large pool of assessors is required because a nonexpert assessor will quickly learn to recognise characteristic artefacts in the video sequences (and so become 'expert'). These factors make it expensive and time consuming to carry out the DSCQS tests thoroughly.

3.4 PSNR

Peak Signal to Noise Ratio (PSNR) (Equation 2.7) is measured on a logarithmic scale and depends on the mean squared error (MSE) of between an original and an impaired image or video frame, relative to (2n −1)2 (the square of the highest-possible signal value in the image, where n is the number of bits per image sample).

$$PSNR_{dB} = \frac{\mathbf{10\log_{10}(2^n - 1)^2}}{\mathbf{MSE}}$$

PSNR can be calculated easily and quickly and is therefore a very popular quality measure,widely used to compare the 'quality' of compressed and decompressed video images.The PSNR measure suffers from a number of limitations. PSNR requires an unimpaired original image for comparison but this may not be available in every case and it may not be easy to verify that an 'original' image has perfect fidelity. PSNR does not correlate well with subjective video quality measures such as those defined in ITU-R 500. For a given image or image sequence, high PSNR usually indicates high quality and low PSNR usually indicates low quality. However, a particular value of PSNR does not necessarily equate to an 'absolute' subjective quality.

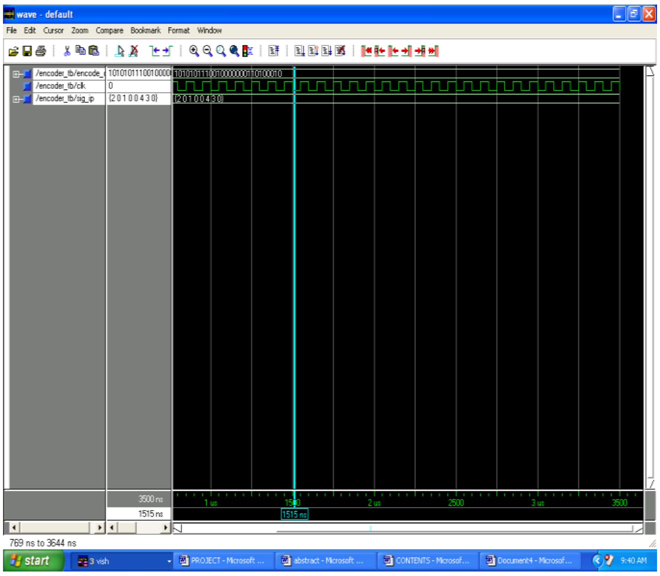
3.5 Other Objective Quality Metrics

Because of the limitations of crude metrics such as PSNR, there has been a lot of work in recent years to try to develop a more sophisticated objective test that more closely approaches subjective test results. Many different approaches have been proposed but none of these has emerged as a clear alternative to subjective tests.

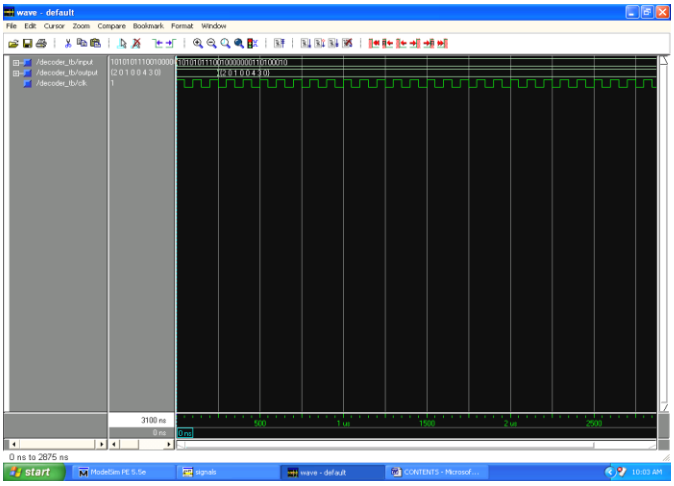
As yet there is no standardised, accurate system for objective ('automatic') quality measurement that is suitable for digitally coded video. In recognition of this, the ITU-T Video Quality Experts Group (VQEG) aim to develop standards for objective video quality evaluation. The first step in this process was to test and compare potential models for objective evaluation.

4. OUTPUTS

4.1 Encoder Output



4.2 Decoder Output



5. CONCLUSIONS

The emerging H.264/AVC video coding standard has been developed and standardized collaboratively by both the ITU-T VCEG and ISO/IEC MPEG organizations.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. Its video coding layer design is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important differences relative to prior standards.We summarize some of the important differences thusly:

- Enhanced motion prediction capability
- Use of a small block-size exact-match transform
- Adaptive in-loop deblocking filter
- Enhanced entropy coding methods

When used well together, the features of the new design provide approximately a 50% bit rate savings for equivalent perceptual quality relative to the performance of prior standards (especially for higher-latency applications which allow some use of reverse temporal prediction).The new video standard known as H.264/AVC presents a rich collection of state-of-the-art video coding capabilities that can provide interoperable video broadcast or communication with degrees of capability that far surpass those of prior standards.

6. REFERENCES

[1] ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVTE022.

[2] JVT Document JVT-C028, G.Bjontegaard and K. Lillevold, “Context-Adaptive VLC Coding of Coefficients”,Fairfax, VA, JVT Document JVT-L13, D. Marpe, G. Blattermann and T. Wiegand, “Adaptive Codes for H.26L”, Eibsee.

[3] ISO/IEC 13818, Information Technology – Generic Coding ofMoving Pictures and Associated AudioInformation.ISO/IEC 14496-2, Coding of Audio-Visual Objects – Part 2:Visual.

[4] ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding.

[5] F. Pereira and T. Ebrahimi (eds), The MPEG-4 Book, IMSC Press.

[6] A. Walsh and M. Bourges-S´evenier (eds), MPEG-4 Jump Start, Prentice-Hall.

[7] ISO/IEC JTC1/SC29/WG11 N4668, MPEG-4 Overview, http://www.m4if.org/resources/Overview.pd