

Research Article

Simulation of H.264 based Real time Video Encoder for Underwater Acoustic Channel

Uma B.V.^{A*}, Geetha K.S.^A, Prasanna Kumar S.C.^B and Naveen Kumar L.^A

^ADept of ECE, ^BDept of IT, R.V. College of Engineering, Karnataka, India

Accepted 15 May 2014, Available online 01 June 2014, Vol.4, No.3 (June 2014)

Abstract

In order to transmit a better quality video in under water acoustic channel, compression schemes with high compression ratio and less time complexities are required. Conventional coding schemes can deliver high compression ratios but need to be optimized for real time embedded applications. This paper discusses about the development and software implementation of a real-time coding scheme for underwater video compression based on H264/MPEG-4 Part-10 coding standard with a compression ratio greater than 200:1 in a low bit rate (between 20 to 50 Kb/s) underwater acoustic channel. The proposed encoder utilizes object-based coding and sprite coding approach for efficient low rate compression. The software is implemented using OpenCV for image acquisition and the main implementation of the software is in C language. The coding scheme is tested on a underwater video sequence to give a compression ratio of 204:1 and a peak signal to noise ratio (PSNR) of 41.38 dB, which satisfies the requirements of real-time underwater video transmission with good quality.

Keywords: Compression ratio, H264, OpenCV, PSNR, Underwater Acoustic Channel, Video Compression.

1. Introduction

Implementation of high speed underwater wireless communication links is currently an area of active research. One of the major applications of underwater wireless communication is underwater video transmission which is used in many fields like underwater defence applications, real-time monitoring of marine life and reef, and even sports, entertainment, education and more.

There are two major ways of underwater communication. One is using electromagnetic waves and the other is using acoustic waves (L.V. Pie, *et al*, 2008). Recent research shows promising results using electromagnetic waves in the blue-green range of the visible spectrum. This is achieved by maintaining accurate optical channel between transmitter and receiver. This makes mobile platforms impractical to be used in such applications. Other major drawbacks are scattering of optical waves and limitation of short distance coverage (<100m).

On the other hand, acoustic waves, due to high density of water they travel through water exceptionally well. Since the speed of sound is several times less than the speed of light, the data rate is much lower (typical data rates ranges from 20 to 50 Kb/s) than that of light wave communication (Mikhalevsky, *et al*, 1999). Since the data rate in acoustic transmission will always be less than the

equivalent optical transmission solution. This limitation brings about the need for compression when attempting to transmit video in an underwater acoustic channel. The fact that there is a very low usable channel band width makes it important to find a powerful compression technique that can reduce the bit rate requirements for real-time underwater video transmissions. This necessitates the coding schemes with high compression ratio, but it leads to high computational requirements.

H.264/MPEG-4 Part 10 is a block based compression standard which makes it difficult to implement on low computational power devices such as mobile and other handheld devices which are used in real time applications. To overcome this, object-based coding and sprite coding are used with H.264 to make it feasible for real time embedded systems. Uneven multi-hexagonal(UMH) search algorithm is used in motion estimation part because the real-time implementation requires good search speed without a significant sacrifice in PSNR. This can be found in UMH. These components are discussed briefly in preceding sections.

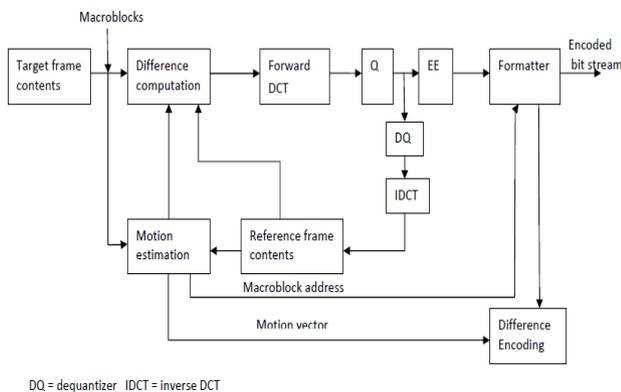
The paper is organized as follows. Section 2 gives the H.264 based encoder design, section 3 discusses object-based coding and sprite coding. Section 4 presents the simulation results. Conclusions are drawn in section 5.

2. Encoder Design

H.264 standard is widely used in high bit rate channel because of its high quality and compression ratio. The

*Corresponding author **Dr Uma B.V, Dr. Geetha K.S** and **Dr. Prasanna Kumar S.C** are working as Prof., **Naveen Kumar L** as Research Staff

standard uses different modes of prediction and motion compensation techniques. This increases the computational complexity of the standard. Hence H.264/MPEG-4 is not suitable for low bit rate application. To overcome this problem, there is a need to select proper algorithms and components for the encoder and make it possible to implement the encoder for low bit rate channel. In this work, an encoder based on H.264 standard is developed for low data rate real time video compression



DQ = dequantizer IDCT = inverse DCT

Fig.1 Block diagram of H.264 Encoder

Figure 1 shows the block diagram of the proposed H.264 based encoder. Transform, Quantization (Q), motion estimation and entropy (EE) are the main blocks of the encoder (JVT Recommendation, 2003). Raw frame contents are fed to the encoder where it initially gets converted to frames for further process. Spatial and temporal redundancies are exploited to greatly reduce the amount of data to be encoded in prediction part. Prediction may be of intra or inter type. Intra prediction is for the generation of I frames whereas the later one is for multi frame generation.

Further, transform and quantization expresses energy of the frames in the form of matrix by applying mathematical techniques like DCT (Discrete cosine transform) and trellis respectively. Motion estimation and compensation uses already encoded to reduce the prediction errors by keeping reference frames.

Final stage is the entropy coding. H.264 uses improved entropy coding techniques like context-adaptive variable-length coding (CAVLC) and content-adaptive binary arithmetic coding (CABAC). Both are lossless entropy coding techniques, CAVLC takes advantage of quantized block by scanning in zig zag direction to find out highest non-zero coefficients whereas as CABAC selects probability models based on elements context.

Motion estimation, Rate control and Quantization are the main blocks in the encoder that decides the compression efficiency and data rate. Hence this paper concentrates on developing novel technique for the above three functionalities of the encoder.

2.1 Motion Estimation

The most time consuming and complex part of H.264 is motion estimation. If this complexity and time

consumption is reduced, it offers better performance for low computational power devices. There are several motion estimation algorithms available in literature. For example, successive elimination exhaustive search, diamond search and hexagon search algorithms (Ismail et.al., 2012), (Zhu et.al., 2012).

Successive Elimination Exhaustive search algorithm is a brute-force search algorithm that checks every block available to find the closest match to the block under consideration. This algorithm is used when quality is the most important factor in the motion estimation process (Meritt L. et al., 2007). The Diamond Search (DS) algorithm performs block-matching motion estimation by employing a square-shaped search pattern in four directions around the checking point under consideration: up, down, left and right. This algorithm is used when search speed takes priority over image quality.

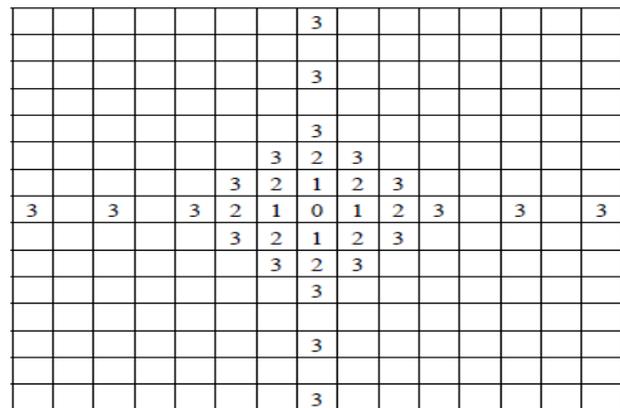


Fig.2 Search approach for early termination decision in Uneven Multi-Hexagon search

In this paper, uneven multi-hexagonal (UMH) search algorithm which is a combination of the diamond search and the hexagon search algorithms (Zhu et.al., 2000) is used. This combination is utilized in order to perform computations faster and obtain reliable motion vectors. This is done by applying the diamond search algorithm first, then using the hexagon search algorithm. The scheme uses these two algorithms alternatively until a terminating condition is reached. The motion vectors found by the algorithm are reliable and suitable for real-time applications due to lower computations required in comparison with schemes that rely only on either the diamond search or the hexagon search. Early termination along with UMH greatly reduces the time consumption. Figure 2 shows the search pattern for the early termination decision in UMH. Diamond search or radius one is labeled as '1', diamond search of radius two is labeled as '2', and symmetric cross search of radius '7' and octagon search of radius two are labeled as '3'.

2.2 Rate Control

Rate control is about controlling the output bit rate by controlling the quantization parameters. It is mostly empirical. In this work, one-pass scheme is used for real

time operations. The one-pass method involves using faster motion estimation method over a half-resolution version of the frame, and uses the Sum of Absolute Hadamard Transformed Differences. The real-time implementation requires quick processing.

2.3 Macroblock Mode Decision and Quantization

In macroblock mode decision, for real time compression, 8x8 blocks for both inter-prediction and intra-predictions are used. To speed up mode decision and for the final refinement, two early terminations are used.

After selecting a macroblock type and motion vector (and during each RD evaluation), the residual is computed, which is the DCT of difference between the input frame and the intra-or inter-predicted macroblock. Integer values for representing the DCT coefficients are selected and this is known as quantization.

Table 1. Configuration parameters of the Encoder

Motion Estimation	UMH
Block Size (Intra & Inter Prediction)	8x8
GOP	IPBBPBBIBB
Hadamard Transform	Used
Early Termination	Used
Entropy Coding	CAVLC
Quantization	Trellis 1
Video Size	176x144

The configuration parameters of the encoder used is summarized in table 1. Other parameters which are not mentioned in the table are same as that of the standard H.264/MPEG-4, since they support real time video compression scheme.

3. Object-Based Coding and Sprite Coding

In this section, logical structure of object-based video bit stream and sprite coding used with the proposed encoder are explained.

3.1 Object-Based Coding

Object-based coding allows access and manipulation of objects instead of pixels with good error robustness. This results in large bit rates and high quality compression. Table 2 shows logical structure of a video bit stream of object based encoding. Hierarchy of the different types of objects encapsulated in a video sequence can be observed.

Table 2. MPEG-4 video bit stream logical structure

Visual Object Sequence (VS)
Video Object (VO)
Video Object Layer (VOL)
Group of VOPs (GOV)
Video Object Plane (VOP)

Start codes are used to access the objects in each level. Visual Object Sequence (VS), Video Object (VO), Video Object Layer (VOL), Group of Video Object Planes (GOV) and Video object Plane (VOP) represents the scene of a video.

VS refers to the complete MPEG-4 scene, which may contain any 2D or 3D natural or synthetic objects and their enhancement layers. VO represents particular object in the instance. This may corresponds to object of the background scene. Video object layer supports multi layer or single layer encoding. Group of video object layer forms a GOV. GOVs can provide points in the bit stream where VOPs are encoded independently from each other. Thus they can provide random access points into the bit stream. Time sampling of video object produces video object plane. Dependent or independent encoding of each other VOPs are available. This results in faster processing speed and good quality.

3.2 Sprite Coding

Any video sequence will contain temporal redundancies, i.e. parts of the video sequence remain still and unchanging over a period of time. This quality of a video sequence is exploited in sprite coding.

A sprite, also referred to as a mosaic, is an image composed of pixels belonging to such a video object visible through out a video sequence. Sprite coding is a well-known and efficient technique for object based video representation and compression. Global motion estimation (GME) and automatic segmentation are used to generate the background sprite image. The GME technique is a key component in the sprite generation for estimating the motion information of background object.

Main advantage of sprite coding is reconstruction of background image by applying geometrical projections. This increases the coding efficiency and speed. Sprite coding can also be applied for block or frame based encoding to achieve better performance. Therefore, proposed encoder makes use of the sprite coding to increase the efficiency of the encoder.

4. Simulation Results

This section presents results of performance evaluation conducted on the proposed compression scheme. The compression scheme has been tested in a personal computer on a raw underwater video sequence.



Fig. 4. Snapshots of video frames used for performance evaluation: (a) uncompressed (b) compressed

Table 3 Performance measurement results for underwater video sequence

Frame count		1950
Video duration (Sec)		115
Time taken to process one frame (ms)		57.14
Compression	Uncompressed (KB)	73865
	Compressed (KB)	362
	Ratio	204:01:00
	Time(Sec)	8.48
PSNR (dB)		41.38
Compression speed (frames/ sec)		230

H.264 based encoder is developed as described in section 2 and 3 and used in the compression process. Following are the performance evaluation parameters used:

4.1 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio is defined as the ratio between the maximum possible power of the signal, to the power of the corrupting noise that affects the fidelity of its representation. It is expressed in terms of logarithmic decibel scale.

With respect to video compression, the PSNR is defined via the Mean Squared Error, as expressed in eqn. (1) and (2).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (1)$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{\max_f}{MSE} \right) \text{ dB} \quad (2)$$

Where $I(i, j)$ is the two dimensional vector equivalent of compressed data and $k(i, j)$ is the two dimensional vector equivalent of uncompressed data.

4.2 Peak Signal-to-Noise Ratio (PSNR)

Compression ratio is used to quantify the reduction in data representation size produced by a data compression algorithm. It is defined as the ratio of the uncompressed file size to the compressed file size as expressed in eqn. (3).

$$\text{Compression ratio} = \frac{\text{Uncompressed size}}{\text{Compressed size}} \quad (3)$$

4.3 Frames Per Second (FPS)

Frames per second is the rate (frequency) at which an imaging device produces unique consecutive frames. Human eye can process 10 to 12 separate frames per second. For better quality and strain less vision of a video, it is preferred to record at more than 15 FPS. Therefore, in this work, it is ensured that the required FPS achieved using proposed encoder.

A sample raw video sequence of size 176x144 taken from an underwater video camera is used as the test

sequence for the evaluation. Snapshots of 561th frame are shown in figure 4. Figure 4.(a) is the uncompressed raw video and 4.(b) is the compressed video. Only slight variation in the quality of the compressed video can be observed which shows the efficiency of the encoder.

Table 2 shows performance of the encoder. The result of simulation shows a compression ratio of 204:1 and PSNR of 41.38dB at a data rate of 37.6 Kbps, which meets the requirements of underwater acoustic channel for real-time application. The compression speed achieved using a personal computer is 230 fps which shows the performance of the designed encoder. The performance is achieved due to the object based coding and sprite coding approach.

Conclusions

Real time underwater video compression scheme for low bit rate underwater acoustic channel is designed using Object based coding and sprite coding approaches. The encoder is developed in OpenCV and C language is based on the popular H.264/MPEG-4 Part-10 encoder. The developed encoder is simulated on a personal computer and its performance is evaluated. The results of the evaluation shows that the algorithm used is suitable for real time applications due to its high compression ratio of 204:1, PSNR of 41.38 dB and processing the speed of 230 FPS.

Acknowledgment

Our sincere thanks to Naval Research Board (NRB), India as this work is part of the project funded by NRB.

References

- L.V.Pei; H.E.Junhua; Zhou Renkui and Liu Haiying (2008). The application of underwater optics and its development, *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference*, March 2008, Volume 6837, Article ID: 68370I.
- P.N. Mikhalovsky; A.N. Gavrilov and A.B. Baggeroer (1999). The transarctic acoustic propagation experiment and climate monitoring in the arctic, *IEEE Journal of Oceanic Engineering*, pp. 183–201.
- M. Owaid, M. Koziri, I. Katsavounidis, and G. Stamoulis (2009). A high performance and low power hardware architecture for the transform quantization stages in H.264, in *IEEE International Conference on Multimedia and Expo 2009 (ICME 2009)*, pp. 1102–1105.
- S.-C. Hsia, S.-H. Wang, and Y.-C. Chou (2007). A configurable IP for mode decision of H.264/AVC encoder, in *Second NASA/ESA Conference on Adaptive Hardware and Systems 2007 (AHS 2007)*, pp. 146–152.
- N. Sebastiao, T. Dias, N. Roma, P. F. Flores, and L. Sousa (2008). Application specific programmable IP core for motion estimation: Technology comparison targeting efficient embedded co-processing units, in *11th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools (DSD 2008)*, pp. 181–188.
- Joint Video Team of ITU-T and ISO/IEC JTC 1 (2003). *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)*, document JVT-G050r1.
- Y. Ismail, J. B. McNeely, M. Shaaban, H. Mahmoud, M. A. Bayoumi (2012). Fast Motion Estimation System Using Dynamic Models for H.264/AVC Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 28–42.
- Xingguo Zhu; Lu Yu (2012). Binarization and context model selection of CABAC based on the distribution of syntax element, *2012 Picture Coding Symposium (PCS)*, pp. 77–80.
- Meritt L. et al. (2007). *Improved Rate Control and Motion Estimation for H.264 Encoder*, *IEEE International Conference on Image Processing, ICIP 2007*, V309-V312.
- S. Zhu, K. K. Ma (2000). A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation, *IEEE Transactions on Image Processing*, pp. 287–290.