A novel Fuzzy based Medical video compression using H.264

J. Mohanalin Rajarathnam;
PhD student, *Indian Institution of technology Kanpur, Y6104096*

Abstract

This paper aims at applying fuzzy concepts along with the H.264 to out perform the existing H.264 algorithms and MPEG 4 in the field of video compression. We also check the bit rate of H.264 and MPEG4 and found the results are encouraging for H.264.The fist set of experiments shows that when compared to MPEG4, H.264 achieves a significant PSNR ratio for the test of video sequences and thus is more effective in medical video compression. We also give a brief outlook of the H.264 standards and highlight the advantages of it when compared to MPEG 4

Keywords: *Fuzzy, H.264, MPEG4, video compression, bit rate*

Introduction

In the clinical practice a large amount of 2D- and 3D-image and video data must be archived. However, existing techniques do not suffice (e.g., because of the high quality requirements imposed in medical applications). We develop better techniques for image- and video compression with high quality. We pay particular attention to aspects such as "user-friendliness" (e.g., fast algorithms, techniques that allow the extraction of a region of interest without having to decompress the complete image, ...). We also investigate techniques for "intelligent transmission" of medical image and video, e.g., techniques that adapt the image quality to the available bandwidth or that transmit material that is of high interest to the physician with higher priority.

This paper aims at applying H.264 in medical video compression applications and improving the H.264 rate control algorithm with better perceptual quality. First, H.264 is briefly reviewed and introduced to the area of medical video compression. Second, a new fuzzy based algorithm is proposed to fuzzify each frame of the video and then do the H.264 based compression. Third, two sets of experiments are conducted: the comparison between MPEG-4 and H.264, and the comparison between JVT-H014, which is the traditional H.264 control algorithm, and our proposed fuzzy scheme. In 1994, Tsai *et al.* developed a compression scheme for angiogram video sequence based on a full frame discrete wavelet transform. Many techniques make great contributions to video compression with removal of spatial and temporal redundancy in and between frames. However, compressed video is still rather large for applications such as surveillance system. In order to compress more, in video compression techniques, region-of-interest (ROI) and frames-of-interest (FOI) codings could be addressed to set high priority to ROI or FOI by allocating more bits than others. Normally, locations of related coefficients for the reconstruction of the ROI are calculated according to the filter length. However, it is not efficient. This full frame design deals with the local characteristics of the compensated difference signals and performs a higher compression gain. By adaptively searching the prediction error to find the locations of the block artifacts and by modifying the error coding, it is possible to eradicate the artifacts from the final image.

A texture modeling approach is used to encode the high-frequency sub band wavelet coefficients in such regions. This is only performed in regions which are considered diagnostically unimportant, with diagnostically important regions encoded as normal. A hybrid model has been discussed for the compression of CT frames. The model uses lossless compression in the region of interest (ROI) with high quality, and lossy compression in other regions with very high compression ratio and reasonably good quality.

The compression scheme is designed to automatically segment and utilize ROI in order to achieve a good balance between video quality and compression ratio. The digital imaging and communications in medicine (DICOM) is the most commonly used standard which facilitates the allocation and presentation of medical images. It allows both lossless and lossy compression for various kinds of video/image sequences including echocardiography and CT..

1. Brief evaluation of H.264

H.264 Overview

Broadcast television and home entertainment h ave been revolutionised by the advent of digital TV and DVD-video. These applications and many more were made possible by the standardisation of video compression technology. The next standard in the MPEG series, MPEG4, is enabling a new generation of internet-based video applications whilst the ITU-T H.263 standard for video compression is now widely used in video conferencing systems.

MPEG4 (Visual) and H.263 are standards that are based on video compression ("video coding")

Technology from circa. 1995. The groups responsible for these standards, the Motion Picture Experts Group and the Video Coding Experts Group (MPEG and VCEG) are in the final stages of developing a new standard that promises to significantly outperform MPEG4 and H.263, providing better compression of video images together with a range of features supporting high-quality, low-bit rate streaming video.

The history of the new standard, "Advanced Video Coding" (AVC), goes back at least 7 years. After finalizing the original H.263 standard for video telephony in 1995, the ITU-T Video Coding

Experts Group (VCEG) started work on two further development areas: a "short-term" effort to add extra features to H.263 (resulting in Version 2 of the standard) and a "long-term" effort to develop a new standard for low bit rate visual communications. The long-term effort led to the draft "H.26L" standard, offering significantly better video compression efficiency than previous ITU-T standards. In 2001, the ISO Motion Picture Experts Group (MPEG) recognized the potential benefits of H.26L and the Joint Video Team (JVT) was formed, including experts from MPEG and VCEG. JVT's main task is to develop the draft H.26L "model" into a full International Standard. In fact, the outcome will be two identical) standards: ISO MPEG4 Part 10 of MPEG4 and ITU-T H.264. The "official" title of the new standard is Advanced Video Coding (AVC); however, it is widely known by its old working title, H.26L and by its ITU document number, H.264.

2. H.264 Codec

In common with earlier standards (such as MPEG1, MPEG2 and MPEG4), the H.264 draft standard does not

explicitly define a CODEC (encoder / Decoder pair). Rather, the standard defines the syntax of an encoded video bit stream together with the method of decoding this bit stream. The Encoder (Figure 2.1) includes two dataflow paths, a "forward" path (left to right, shown in blue) and a "reconstruction" path (right to left, shown in magenta). The dataflow path in the Decoder (Figure2.2) is shown from right to left to illustrate the similarities between Encoder and Decoder.

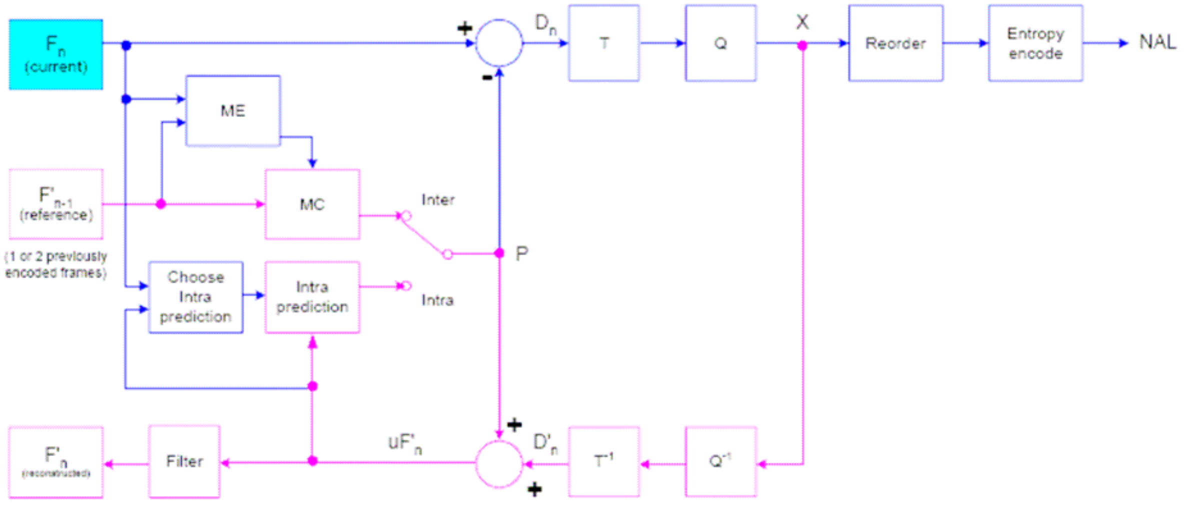


Figure 2-1 AVC Encoder

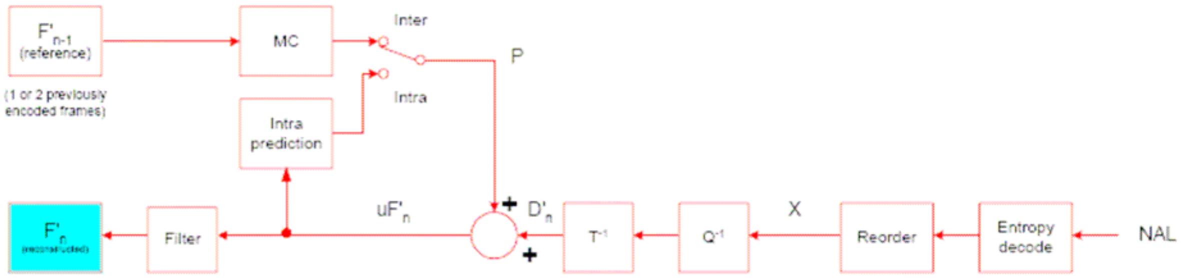


Figure 2-2 AVC Decoder

3. Encoder

An input frame Fn is presented for encoding. The frame is processed in units of a macroblock (corresponding to 16x16 pixels in the original image). Each macroblock is encoded in **intra** or **inter** mode. In either case, a prediction macroblock P is formed based on a reconstructed frame. The prediction P is subtracted from the current macroblock to produce a residual or difference macroblock Dn. This is transformed (using a block transform) and quantized to give X, a set of quantized transform coefficients. These coefficients are re-ordered and entropy encoded.

4. Decoder

The decoder receives a compressed bit stream from the NAL. The data elements are entropy decoded and reordered to produce a set of quantized coefficients X. These are rescaled and inverse transformed to give Dn' (this identical to the Dn' shown in the Encoder). Using the header information decoded from the bitstream, the decoder creates a prediction macroblock P, identical to the original prediction P formed in the encoder. P is added to Dn' to produce uF'n which this is filtered to create the decoded macroblock F'n.

5. Tree structured motion compensation

AVC supports motion compensation block sizes ranging from 16x16 to 4x4 luminance samples with many options between the two. The luminance component of each macroblock (16x16 samples) may be split up in 4 ways as shown in Figure 2-1: 16x16, 16x8, 8x16 or 8x8. Each of the sub-divided regions is a macroblock partition. If the 8x8 mode is chosen, each of the four 8x8 macroblock partitions within the macroblock may be split in a further 4 ways as shown in Figure 2-

2: 8x8, 8x4, 4x8 or 4x4 (known as macroblock sub-partitions). These partitions and sub-partitions give rise to a large number of possible combinations within each macroblock.

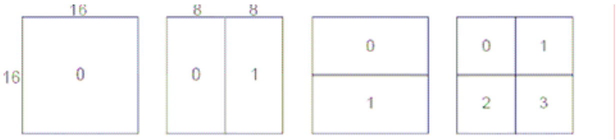


fig 5.1

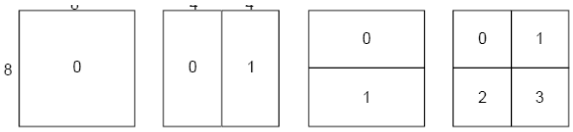


fig 5.2

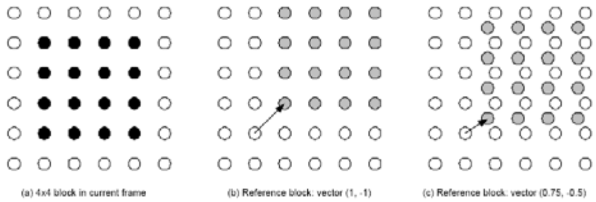

(a) 4x4 block in current frame
(b) Reference block: vector (1, -1)
(c) Reference block: vector (0.75, -0.5)

fig 5.3

A separate motion vector is required for each partition or sub-partition. Each motion vector must be coded and transmitted; in addition, the choice of partition(s) must be encoded in the compressed bitstream. Choosing a large partition size (e.g. 16x16, 16x8, 8x16) means that a small number of bits are required to signal the choice of motion vector(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in frame areas with high detail.

6. Motion vector prediction

Encoding a motion vector for each partition can take a significant number of bits, especially if small partition sizes are chosen. Motion vectors for neighbouring partitions are often highly correlated and so each motion vector is predicted from vectors of nearby, previously coded partitions. A predicted vector, MVp, is formed based on previously calculated motion vectors. MVD, the difference between the current vector and the predicted vector, is encoded and transmitted.

7. Transform and quantization

Each residual macroblock is transformed, quantized and coded. Previous standards such as MPEG-1, MPEG-2, MPEG-4 and H.263 made use of the 8x8 Discrete Cosine Transform (DCT) as the basic transform. The “baseline” profile of H.264 uses three transforms depending on the type of residual data that is to be coded: a transform for the 4x4 array of luma DC coefficients in intra macroblocks (predicted in 16x16 mode), a transform for the 2x2 array of chroma DC coefficients (in any macroblock) and a transform for all other 4x4 blocks in the residual data. If the optional “adaptive block size transform” mode is used, further transforms are chosen depending on the motion compensation block size (4x8, 8x4, 8x8, 16x8, etc). Data within a macroblock are transmitted in the order shown in Figure 6-1. If the macroblock is coded in 16x16 Intra mode, then the block labelled “-1” is transmitted first, containing the DC coefficient of each 4x4 luma block. Next, the luma residual blocks 0-15 are transmitted in the order shown (with the DC coefficient set to zero in a 16x16 Intra macroblock). Blocks 16 and 17 contain a 2x2 array of DC coefficients from the Cb and Cr chroma components respectively. Finally, chroma residual blocks 18- 25 (with zero DC coefficients) are sent.

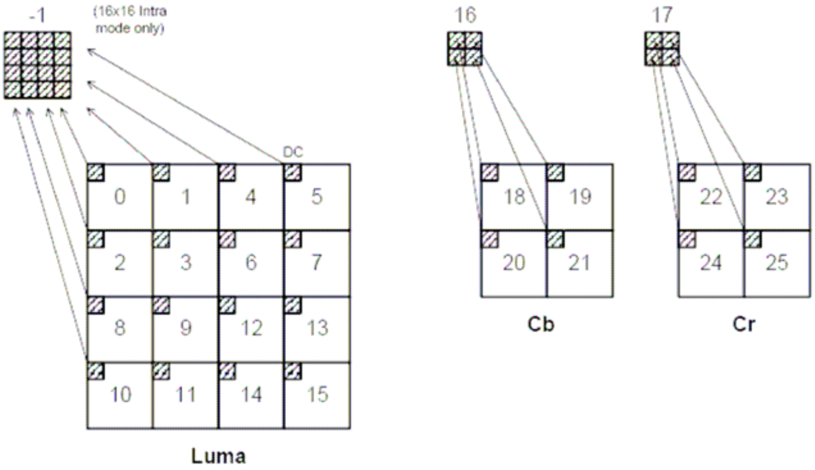


fig 6.1

Development from the 4x4 DCT:
The 4x4 DCT of an input array **X** is given by:

$$\mathbf{Y} = \mathbf{AXA}^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \left[\mathbf{X} \right] \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

Equation 2-1

where:

$$a = \frac{1}{2}$$

$$b = \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right)$$

$$c = \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right)$$

This matrix multiplication can be factorised [2] to the following equivalent form (Equation 2-2):

$$\mathbf{Y} = \left(\mathbf{CXC}^T\right) \otimes \mathbf{E} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \left[\mathbf{X} \right] \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

8. Quantization

H.264 uses a scalar quantizer . The definition and implementation are complicated by the requirements to (a) avoid division and/or floating point arithmetic and (b) incorporate the post- and pre-scaling matrices **Ef** and **Ei** described above.

The basic forward quantizer operation is as follows:

Zij = round(Yij/Qstep)

Example:

QP = 4, hence Qstep = 1.0
(i,j) = (0,0), hence PF = a2 = 0.25
qbits = 15, hence 2qbits = 32768
Qstep
PF
2

MF/ 2^{qbits} = , hence MF = (32768x0.25)/1 = **8192**

9. Context-Based Adaptive Arithmetic Coding (CABAC):

When entropy_coding_mode is set to 1, an arithmetic coding system is used to encode and decode H.264 syntax elements. The arithmetic coding scheme selected for H.264, Context-based Adaptive Binary Arithmetic Coding or CABAC, achieves good compression performance through (a) selecting probability models for each syntax element according to the element's context, (b) adapting probability estimates based on local statistics and (c) using arithmetic coding.

Coding a data symbol involves the following stages.

1. Binarization: CABAC uses Binary Arithmetic Coding which means that only binary decisions (1 or 0) are encoded. A non-binary-valued symbol (e.g. a transform coefficient or motion vector) is "binarized" or converted into a binary code prior to arithmetic coding. This process is similar to the process of converting a data symbol into a variable length code but the binary code is further encoded (by the arithmetic coder) prior to transmission.

Stages 2, 3 and 4 are repeated for each bit (or "bin") of the binarized symbol.

2. Context model selection: A "context model" is a probability model for one or more bins of the binarized symbol. This model may be chosen from a selection of available models depending on the statistics of recently-coded data symbols. The context model stores the probability of each bin being "1" or "0".

3. Arithmetic encoding: An arithmetic coder encodes each bin according to the selected probability model. Note that there are just two sub-ranges for each bin (corresponding to "0" and "1").

4. Probability update: The selected context model is updated based on the actual coded value (e.g. if the bin value was "1", the frequency count of "1"s is increased).

10. Rate Distortion Optimization

One of the novel features of H.264 video coding is the use of seven different macroblock (MB) coding modes so that the temporal and spatial details in an MB are best presented. These coding modes are SKIP, INTER 16*16, INTER 16* 8, INTER 8*16, INTER 8*8, INTRA 16 *16, INTRA 4 *4. In INTER 8* 8 mode, each block can be further divided independently into 8* 8, 8* 4, 4* 8, or 4* 4 subpartitions. To select the best mode for each MB, all the MB modes are tried and the one that leads to the least rate-distortion (RD) cost is selected. This is to achieve the best tradeoff between the rate and distortion performance, and is called rate distortion optimization (RDO). RDO is achieved by selecting the mode with the least RD cost using Lagrangian multiplier. The procedure can be defined as follows:

$$\min J_i : \quad J_i = D_i + \lambda_i \times R_i$$

where is the RD cost of an MB coded using Mode , is the data bits it consumed, and is the distortion it caused. The mode that has the minimum RD cost is selected as the optimum coding mode for this MB. In most medical images and video sequences, there exist some regions that are of special interests to the doctors, and the other regions are of no diagnostic significance. Hence, the employment of RDO in H.264 makes it especially suitable for coding of medical images: we may decrease the Lagrangian multiplier in to reduce distortion in the ROI, and increase to reduce bit consumption in regions of no medical interest. Therefore, it is possible to select the best coding mode so as to achieve a compromise between high compression performances.

Although H.264 is currently the most efficient video compression standard, its rate control algorithm is not very satisfactory and needs further improvement in order for it to achieve high coding efficiency in encoding medical video sequences. In the equation, the target bit is estimated solely based on the buffer fullness, regardless of the frame's content. This may lead to a drastic drop in peak signal-to-noise ratio (PSNR), especially in the case of high motion scenes or scene changes. Jiang *et al.* introduced the mean absolute difference (MAD) ratio as a measure of motion complexity (MC) to improve the video quality at scene changes. However, MAD ratio is not a good ways for representing the motion contents, as it can only represent the similarity between the current frame and its reference frame. Another problem of the adopted rate control algorithm is that MAD is selected to estimate quantization parameter (QP) and to decide coding modes.

However, it is difficult to obtain the actual bits consumed before entropy encoding. What we can do is only to estimate the target bit allocated to the current frame. Usually, there is a difference between the estimated bits and the actually consumed bits. The estimated target bit should represent the MC of current frame. Since there is a temporal correlation between consecutive frames, we can use the previous coded frames to estimate the current frame's MC, which can then be used to estimate the target bit.

11. New fuzzy scheme

A video is a sequence of frames. That is collection of sequence is nothing but video.. So a new algorithm is developed where the frames will be fuzzified and then fuzzy based segmentation is applied to it. Each medical image will have more background than its ROI. So we will isolate the ROI and the compression will be minimum here and maximum to the background. this approach will work satisfactorily because we are applying more compression ratio to the background and not to the ROI. The problem with the medical video compression is that it can always induce some artifacts inside which can mislead the doctors to a false conclusion. But using this method along with optimized bit rate scheme we can expect better results. Still the scheme is under development and we expect the results by a year.

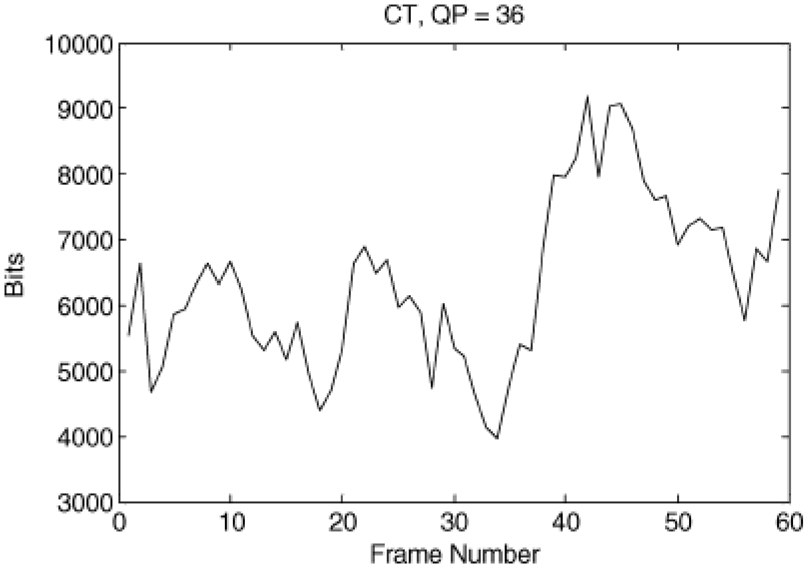
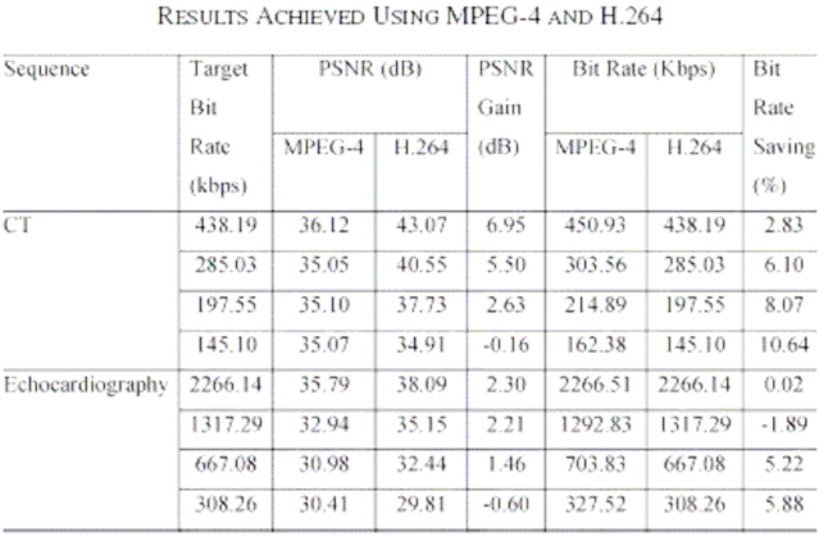


fig 12.1

12. Comparison Between MPEG-4 and H.264

The test conditions of MPEG-4 are set to match H.264 as much as possible. For example, the frame type is IPPP, the frame rate is 30 fps, and the search range is 32, etc. Table III shows the results achieved from MPEG-4 and H.264 without rate control. For each sequence, four bit rates from high to low are selected as the target bit rates. To see the difference between MPEG-4 and H.264, we show PSNR gain and bit rate saving in the table.



RESULTS ACHIEVED USING MPEG-4 AND H.264

Sequence	Target Bit Rate (kbps)	PSNR (dB)		PSNR Gain (dB)	Bit Rate (Kbps)		Bit Rate Saving (%)
		MPEG-4	H.264		MPEG-4	H.264	
CT	438.19	36.12	43.07	6.95	450.93	438.19	2.83
	285.03	35.05	40.55	5.50	303.56	285.03	6.10
	197.55	35.10	37.73	2.63	214.89	197.55	8.07
	145.10	35.07	34.91	-0.16	162.38	145.10	10.64
Echocardiography	2266.14	35.79	38.09	2.30	2266.51	2266.14	0.02
	1317.29	32.94	35.15	2.21	1292.83	1317.29	-1.89
	667.08	30.98	32.44	1.46	703.83	667.08	5.22
	308.26	30.41	29.81	-0.60	327.52	308.26	5.88

Table I

It is obvious from Table I that H.264 performs much better than MPEG-4 when applied in the test medical video sequences. For sequence “CT,” H.264 obtains positive PSNR gain and bit rate saving in most cases. Even a significantly high PSNR gain of 6.95 dB is earned at high bit rate. For sequence “Echocardiography,” H.264 also obtains positive PSNR gain and bit rate saving in most cases.

13. Conclusions

In this paper, we have introduced H.264 for compression of medical videos and 3-D medical data sets. Experimental results have shown that for medical video compression, H.264 performs much better than MPEG-4 both in PSNR gain and bit rate saving. This means that H.264 is more effective than MPEG-4, and hence is a good alternative solution for medical video applications. We have also proposed a new fuzzy based scheme is proposed where we target the individual frames. We give the sample results along with this paper. In this scheme, MC is used to estimate a frame’s target bit so that the bit allocation is in accordance with the complexity of frame’s motion contents, and perceptual mode decision is employed to allocate the MB’s bits perceptually. Experimental results have shown that our proposed rate control scheme outperforms H.264 rate control for medical video compression.

14. References

[1] S. B. Gokturk, C. Tomasi, B. Girod, and C. Beaulieu, “Medical image compression based on region of interest, with application to colon CT images,” in *Proc. 23rd Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, Istanbul, Turkey, Oct. 2001.

[2] D. Clunie and B. J. Erickson, “The new enhanced multiframe CT and MR DICOM objects,” in *Proc. 2005 SCAR (Society for Computer Applications in Radiology) Annual Meeting*, Orlando, FL, Jun. 2005.

[3] Reference Model MoMuSys, MPEG. [Online]. Available: http://www.ganesh.org/~pmeerw/MPEG4_MoMuSys-FDIS-V1.0-990 812-bugfixed.tar.gz

[3] C. Lau, J. E. Cabral, A. H. Rambhia, and Y. M. Kim, “MPEG-4 coding of ultrasound sequences,” in *Proc. SPIE*, vol. 3976, San Diego, CA, Feb.2000, pp. 573–579.

[4] P. Barbier, M. Alimento, G. Berna, D. Cavoretto, F. Celeste, M.Muratori, and M. D. Guazzi, “Clinical validation of different echocardiographic motion pictures expert group-4 algorithms and compression levels for telemedicine,” in *Proc. Medinfo 2004,* San Francisco, CA, Sep. 2004.

[5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview *Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003. of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst.*

[6] M. M. Ghandi and M. Ghanbari, “A Lagrangian optimization rate control algorithm for the H.264/AVC encoder,” in *Proc. 2004 IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 123–126.

[7] NEMA, Digital Imaging and Communications in Medicine (DICOM) Part 1: Introduction and Overview, National Electrical Manufactures Association, 2004.

[8] A. Joch, F. Kossentini, and P. Nasiopoulos, “A performance analysis of the ITU-T draft H.26L video coding standard,” in *Proc. 12th Int. Packetvideo Workshop*, Pittsburgh, PA, Apr. 2002.

[9] T. Wiegand, H. Schwarz, A. Joch, F. Lossentini, and G. J. Sullivan, “Rate-constrained coder control and comparison of video coding standards,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[10] M. Q. Jiang, X. Q. Yi, and N. Ling, “Improved frame-layer rate control for H.264 using MAD ratio,” in *Proc. 2004 IEEE Int. Symp. Circuits Syst.*, vol. III, Vancouver, BC, Canada, May 2004, pp. 813–816.

[11] S. Winkler, “A perceptual distortion metric for digital color video,” in *Proc. SPIE,* vol. 3644, San Jose, CA, Jan. 1999, pp. 175–184.

[12] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, “The JVT advanced video coding standard: complexity and performance analysis on a tool-by-tool basis,” in *Proc. 13th Int. Packetvideo Workshop*, Nantes, France, Apr. 2003

[12] C. J. Tsai, C. W. Tang, C. H. Chen, and Y. H. Yu, “Adaptive rate-distortion optimization using perceptual hints,” in *Proc. 2004 IEEE Int. Conf. Multimedia and Expo,* Taipei, China, Jun. 2004, pp. 667–670.

[13] F. Pan, “Adaptive image compression using local pattern information,” *Pattern Recognit. Lett.*, vol. 23, no. 14, pp. 1837–1845, Dec. 2002.

[14] A. M. Bazen and S. H. Gerez, “Systematic methods for the computation of the directional fields and singular points of fingerprints,” *IEEE Trans.Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 905–919, Jul. 2002.

[15] *Information Technology—Coding of Audio-Visual Objects—Part 10:Advanced Video Coding*, ISO/IEC FDIS 14496-10, 2003.

[16] T. Hallbach, “H.264 video compression standard,” in *Proceedings of 6th Nordic Signal Processing Symposium*, Bergen, Norway, Oct. 2003.

Article received: 2007-11-08