Two-layer and Adaptive Entropy Coding Algorithms for H.264-based Lossless Image Coding

Jun-Ren Ding, Jiun-Yu Chen, Fu-Chun Yang and Jar-Ferr Yang

Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan
jfyang@ee.ncku.edu.tw

ABSTRACT

In this paper, we propose two-layer coding algorithm to improve the performance of the H.264-based lossless (H.264-LS) image coding. From universal access point of view, the proposed method is based on the H.264 lossy image coding with other CABAC layer to compensate the lossy portion. Besides, the H.264-LS with DPCM (H.264-LS_DPCM) and H.264-LS achieve different coding performance in use of CABAC and CAVLC entropy coders without the DCT and quantization. We further suggest an adaptive entropy coding (AEC) algorithm to determine the best entropy coder by using the image content variations, which is calculated from the sum of absolute difference of intra prediction. Simulation results show that the proposed AEC method have good correct detection rates and improvement of compression rate for H.264-LS and H.264-LS_DPCM coders. The two-layer H.264-LS almost have the same compression rate than the H.264-LS_DPCM.

Index Terms— *Image coding, image compression.*

1. INTRODUCTION

Lossless (LS) image coding is a very important technique to efficiently and perfectly preserve valuable information of medical images, seismic data, digital archives, and digital documentations, which do not allow any distortions, for storage and transmission. The JPEG lossless (JPEG-LS), which combine adaptive prediction and variable length coding, is the first and simple loss image coding standard [1]. However, JPEG-LS is the best lossless image coding method up to now [6], but it can not provide any scalable feature. Based on wavelet transform, the JPEG2000 image coding standard can effectively provide both lossy and lossless image compression [2]. Recently, the H.264-based advance video coding (H.264/AVC), which adopted several advanced coding features, achieves much better video coding performance than the existed video coding standards [3]. For image coding, H.264 only can be operated in intra frame coding to provide a satisfactory lossy image compression. Unfortunately, the H.264 lossless (H.264-LS) can be only based on intra prediction and entropy coding techniques, can not achieve a better performance than the existed lossless image coding standards [1], [2]. Several research works have studied some improvement of H.264-LS [4]-[6]. To reduce the possible transformation error, the designs of discrete cosine transform (DCT) for H.264-LS can be found in [4] and [5]. However, the image quality might still be lost while it involving any transform. In [6], a pixel-by-pixel differential pulse code modulation (DPCM) has present as an enhancement of H.264/AVC standard. With the DPCM technique, the H.264-LS produces the minimum intra prediction residua values to achieve a robust compression in use of entropy coding. Unfortunately, it can not be operated for lossy image coding. To design a universal H.264-based image coding method, as JPEG2000-LS, to simultaneously achieve a better efficient compression for lossy, near lossless and lossless image compression methods should be an important topic for image storage and transmission.

In this paper, we suggest that the context-based adaptive binary arithmetic coding (CABAC) [7], and context-based adaptive variable length coding (CAVLC) [3], should be adaptively adopted for advancing the H.264-based lossless image coding method. We utilize the existing intra prediction residual values to obtain an image content variations (ICV) feature, which can be simply obtained by sum of absolution difference (SAD) after intra prediction. For universal image coding, we suggest two-layer image coding based on H.264 lossy image coding. Simulation results show that the proposed adaptive entropy coding (AEC) method can enhance higher compression rate for the H.264-LS_DPCM and H.264-LS. The two-layer H.264-LS almost have the same compression rate than the H.264-LS_DPCM. The rest of this paper is organized as follows. In Section 2, we briefly review the 4×4 intra prediction coding processes of H.264-LS and H.264-LS_DPCM. In Section 3, a universal image coding will be described. In Section 4, we propose the adaptive entropy coding by using the ICV feature to improve the coding performance for H.264-based lossless coding methods. Finally, the simulations results with some discussions and the conclusions are addressed in Sections 5 and 6, respectively.

2. OVERVIEW OF H.264-LS AND H.264-LS_DPCM

In H.264-LS coding standard, the intra prediction is conducted in a block-based fashion. The intra prediction for YUV image formats utilizes three different prediction blocks (4×4, 8×8 and 16×16) with various prediction modes. As shown in Fig. 1, for example, the predictions for Mode 2 (DC), Mode 0 (Vertical) and Mode 1 (Horizontal), in 4×4 blocks are given by:

Mode 2:

$$p_n^2 = (b_1 + b_2 + b_3 + b_4 + b_9 + b_{10} + b_{11} + b_{12})/8, \text{ for } n = 1, 2, 3, ..., 15, 16; \quad (1)$$

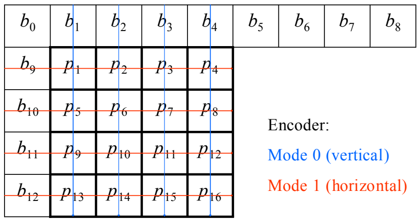
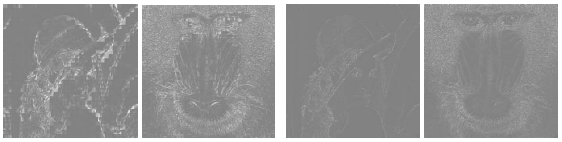


Fig. 1. 4×4 intra prediction Mode 0 and Mode 1 for H.264-LS encoder.



(a) 16×16 (b) 4×4

Fig. 2. Residual values (shift 128) after 4×4 and 16×16 intra predictions for "Lena" and "Baboon" Y gray images.

Mode 0:

$$p_n^0 = b_1, \text{ for } n = 1, 5, 9, 13, \quad (2a)$$

$$p_n^0 = b_2, \text{ for } n = 2, 6, 10, 14, \quad (2b)$$

$$p_n^0 = b_3, \text{ for } n = 3, 7, 11, 15, \quad (2c)$$

$$p_n^0 = b_4, \text{ for } n = 4, 8, 12, 16; \quad (2d)$$

Mode 1:

$$p_n^1 = b_9, \text{ for } n = 1, 2, 3, 4, \quad (3a)$$

$$p_n^1 = b_{10}, \text{ for } n = 5, 6, 7, 8, \quad (3b)$$

$$p_n^1 = b_{11}, \text{ for } n = 9, 10, 11, 12, \quad (3c)$$

$$p_n^1 = b_{12}, \text{ for } n = 13, 14, 15, 16, \quad (3d)$$

where p_n^m presents the n^{th} predicted value in Mode m prediction and b_1 - b_4 and b_9 - b_{12} present the boundary pixels in the upper and left 4×4 blocks, respectively. The residual values, r_n^m after 4×4 intra prediction for Mode m can be expressed by

$$r_n^m = s_n - p_n^m, n = 1, 2, 3, ..., 15, 16, \quad (4)$$

where s_n presents the original spatial domain pixels at the same locations of p_n^m. To determine the best prediction mode, which has the optimal compression, all the modes should be tested to achieve the minimum rate. After intra prediction, the residual values will be coded by entropy coding without the DCT and quantization. As shown in Fig. 2, it is obvious that 4×4 intra prediction has smaller residual values than 16×16 intra prediction in the Y component gray images.

Recently, the DPCM concept has been suggested to achieve a better pixel-by-pixel intra prediction [7]. For Mode 2, the DPCM-based intra prediction has the same prediction as the block-based prediction, which is addressed in (1). As to Mode 0 and Mode 1, the DPCM-based intra prediction values in the encoder are expressed by:

Mode 0:

$$p_n^0 = b_n, n = 1, 2, 3, 4. \quad (5a)$$

$$p_n^0 = s_{n-4}, n = 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16. \quad (5b)$$

Mode 1:

$$p_1^1 = b_9, \ p_5^1 = b_{10}, p_9^1 = b_{11}, p_{13}^1 = b_{12}, \quad (6a)$$

$$p_n^1 = s_{n-1}, n = 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16. \quad (6b)$$

Of course, the residual values, r_n as the block-based prediction can be computed by (4). For lossless compression, the residual values, r_n will be completely reconstructed after entropy decoding. As shown in Fig. 3, in the decoder, the reconstruct signal rs_n^m of the DPCM-based intra prediction of Mode 0 and Mode 1 can be decoded as

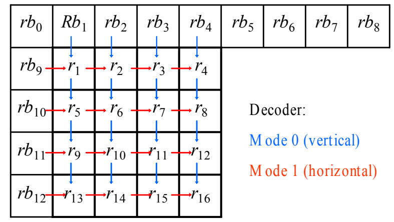
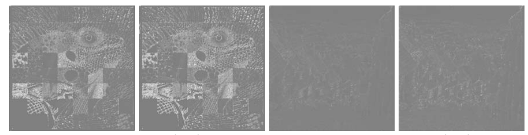


Fig. 3. Reconstruction of 4×4 intra prediction Mode 0 and Mode 1 for H.264-LS_DPCM decoder.



DPCM- Block- DPCM- Block-

Fig. 4. Residual values (shift 128) after DPCM- and block-based intra predictions for "Frymire" and "Goldhill" Y component images.

Mode 0:

$$rs_n^0 = rb_n + r_n, n = 1, 2, 3, 4, \quad (7a)$$

$$rs_n^0 = rs_{n-4}^0 + r_n, n = 5, 6, 7, ..., 15, 16; \quad (7b)$$

Mode 1:

$$rs_1^1 = rb_9 + r_1, rs_5^1 = rb_{10} + r_5, rs_9^1 = rb_{11} + r_9, rs_{13}^1 = rb_{12} + r_{13}, \quad (8a)$$

$$rs_n^1 = rs_{n-1}^1 + r_n, n = 2,3,4,6,7,8,10,11,12,14,15,16, \quad (8b)$$

It is obvious that the residual values after the DPCM pixel-by-pixel intra prediction have smaller than the block-based intra prediction. Fig. 4 shows that the DPCM-based intra prediction has smaller residual values than the block-based intra prediction.

3. H.264-BASED IMAGE CODING FOR UNIVERSAL ACCESS

In general, transform and quantization can not be used for lossless image coding such as JPEG-LS, H.264-LS and H.264-LS_DPCM, since they would cause transform and quantization errors. If we want to transmit one bit-stream, which can provide lossless and lossy images at the same time for different clients, either JPEG-LS or H.264-LS can not offer such capability. To solve this universal access problem, we propose a lossless image coding method based on H.264 lossy image coding as shown in Fig. 5. We can control the lossy image compression bit-stream by controlling the quantization parameters (QP). For lossless applications, we use the existed entropy coding to encode the difference between the original and reconstructed images after executing the inverse DCT and inverse quantization obtained from lossy bitstream. As shown in Fig. 6, it is obvious that difference have the distribution of high frequency in different QP. The D bitstream is used to represent the coding results of difference after executing CABAC. With lossy and D bitstreams, we can achieve lossless compression. To achieve the best lossless image compression, we train 20 images from QP = 1 - 30 to find the optimal compression rate [8]. Fig. 6 shows the distribution of difference values from QP = 5 - 25. According to our analysis, the QP = 10 have the optimal compression rate in average. Table 1 shows the performance of 20 training images with proposed two-layer H.264-LS and previous lossless image coding methods [10]. Besides, the

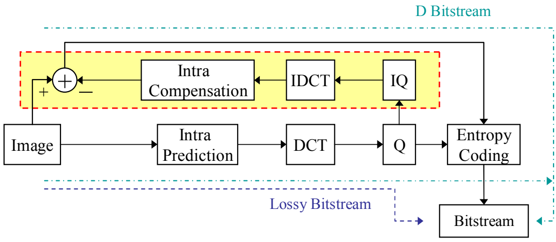


Fig. 5. Block diagram of the proposed H.264-LS based on H.264 lossy image coding.

$QP = 20$ $QP = 30$ $QP = 40$

Fig. 6. Distribution of difference value of Y component gray image "F16" in different QP.

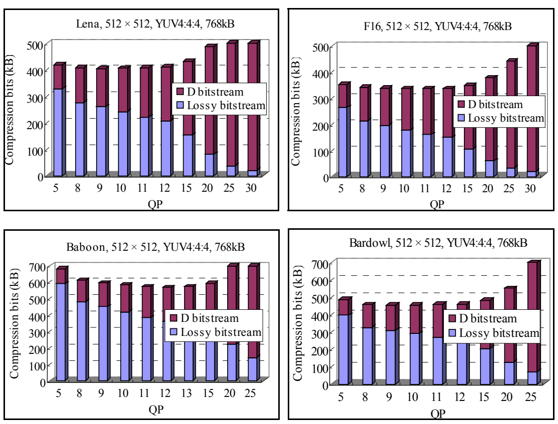


Fig. 7. D and Lossy bitstreams versus QP achieved by the proposed two-layer H.264-LS.

CABAC have better compression rate than the CAVLC in the lossy image coding. Therefore, we only suggest that adopt CABAC for proposed two-layer H.264-LS.

4. ADAPTIVE ENTROPY CODING

Since H.264 has suggested two entropy coding methods, CABAC and CAVLC, simulation results depicted in Table 1 show that H.264-LS and H.264-LS_DPCM will achieve different coding performances while using the CABAC and CAVLC entropy coders. Because the residual values are not operated by DCT transform and quantization after executing intra prediction, either the CABAC or the CAVLC can not achieve the best coding performance. In order to determine a proper entropy coder, in this paper, we suggest that the SAD of the k^{th} 4×4 intra prediction block be defined as

$$SAD_k = \sum_{n=1}^{16} | s_n - p_n^m |_k = \sum_{n=1}^{16} | r_n^m |_k \tag{9}$$

as the ICV feature. In this paper, the sum of SAD_k defined by

$$T = \left\lfloor \log_2 \sum_{k=0}^{K-1} | SAD_k | \right\rfloor \tag{10}$$

is used to decide the entropy coding method to achieve the optimal compression. The suggested AEC detection mechanism becomes

Table 1. Comparisons of H.264-based lossless image coding methods with JPEG-LS and JPEG2000-LS (training images).

A: CAVLC B: CABAC Images 768 kB	JPEG LS	JPEG 2000 LS	H.264-LS			H.264-LS_DPCM			Two-layer H.264-LS
			A	B	T_{block}	A	B	T_{DPCM}	B
	kB					kB			kB
Lena	435	435	441	421	19.96	409	364	19.75	405
F16	378	369	381	362	20.04	344	303	19.73	336
Baboon	592	578	622	759	21.62	587	680	21.44	584
Peppers	456	473	488	497	20.02	452	418	19.89	449
Goldhill	420	389	418	396	20.25	361	321	19.92	362
Barbara	442	378	435	448	20.75	361	335	20.38	357
House	408	404	445	464	20.50	385	360	20.17	382
Peppers2	329	290	383	381	19.12	263	237	18.92	263
Oldmill	505	466	461	488	21.39	439	449	21.12	441
Frymire	386	463	539	697	21.91	390	405	21.71	523
Boats	363	338	377	355	20.08	306	270	19.61	306
Sailboat	502	512	523	568	20.76	496	510	20.58	481
Splash	342	381	378	340	19.20	342	301	19.04	337
Tiffany	370	442	438	415	19.71	401	358	19.53	404
Zelda	374	374	390	350	19.54	342	302	19.16	341
Bardowl	512	488	479	502	21.01	452	446	20.82	454
Anhinga	412	426	401	412	20.78	381	374	20.59	389
Quietime	483	464	453	470	21.13	425	419	20.90	429
Salzburg	406	435	397	374	20.30	374	344	20.00	393
Butrfly1	466	464	429	415	20.42	420	397	20.28	424
Average	**429**	**429**	**444**	**456**		**397**	**380**		**403**
Average with AEC	**N/A**		**433**		**N/A**	**374**		**N/A**	**N/A**
AEC correct rate			**95%**			**95%**			

$$AEC = \begin{cases} T \geq t, & \text{CAVLC entropy coder is activated;} \\ T < t, & \text{CABAC entropy coder is activated,} \end{cases} \tag{11}$$

where t is a threshold, which is determined by experiments from 20 training images as shown in Table 1. Generally, for larger ICV, the prediction errors in the CABAC coding procedure will not be as effective as the CAVLC. From Table 1, the thresholds t of the AEC algorithm for H.264-LS and H.264-LS_DPCM are defined as $t_{DPCM} = 21$ and $t_{block} = 20.5$, respectively.

5. SIMULATION RESULTS

In this paper, we adopt H.264 standard software JM12.2 version, which includes the so-called fidelity range extensions (FRExt) [9], to execute lossless image coding. The relational control parameters for the experiments are defined as Table 2. In all experiments, we only conduct 4×4 intra prediction for all H.264-LS coders. For the H.264-LS_DPCM method, we only execute Method 1 suggested in [7] since it has the primary advance of compression rate than the H.264-LS. As shown in Fig. 8, we test 20 out-side test images obtained from Internet by for ensuring the correct rate [8]. The out-side test images are captured by the newest camera and have more practical application of lossless image compression.

As shown in Table 3, we also compare the other lossless image coding standards such as JPEG-LS. The simulation results show that H.264-LS_DPCM has best compression bits than the other lossless image coders. Generally, the 4×4 intra prediction cooperated with CAVLC will achieve better compression bits than the CABAC if the images are with the higher ICV. For example, training images of "Baboon," "Barbara," and "House," are the images with higher ICV. On the contrary, the 4×4 intra prediction with CABAC will achieve better compression than the CAVLC for the lower ICV. It is obvious that the proposed AEC algorithm, which adaptively adopts the CABAC and CAVLC for different ICV features, will improve the coding performance a lot. However, H.264-LS, H.264-LS_DPCM, and JPEG-LS can not provide the capability for universal access. The proposed two-layer H.264-LS, as the JPEG2000-LS, is designed in multiple layers, which could provides lossy, nearly lossy, and lossless image coding bit streams simultaneously. As shown in Table 4, it is obvious the correct detection rates of the total test images are 80% and 90% for H.264-LS and H.264-DPCMLS, respectively. The drawbacks and advantage of H.264-LS_DPCM and two-layer H.264-LS methods are depicted in Table 4. It is obvious that we should adopt the two-layer H.264-LS for universal access.

6. CONCLUSIONS

In this paper, we proposed the two-layer H.264-LS image coding method, the lossy bitstream provides the original intra frame coding result and the D bitstream offers the compensation of lossy portion. The coding performance of two-layer H.264-LS is only worse than the H.264-LS_DPCM. However, the two-layer H.264-LS could be used for applications of multiple clients with different receiving capability. Since H.264-based lossless image coding methods achieve different coding performance in use of CABAC and CAVLC entropy coders. The proposed AEC algorithm to properly determine the best entropy coder in average achieves 88% and 93% correct detection rates of the optimal entropy coder for H.264-LS and H.264-LS_DPCM coders, respectively. In future, we will advance the compression rate of two-layer H.264-LS for scalable lossless image/video coding.

Table 2. Control parameter setting for H.264-based lossless image coding

Parameter set files \ Coder		H.264-LS, H.264-DPCMLS	Two-layer H.264-LS
Profile_ main.cfg	sourceWidth, sourceHeight	512	
	profileIDC	144 (4:4:4)	
	IDRIntraEnable	1	
	QPISlice	##	3
	selectiveIntraEnable	1	
	symbolMode (0, 1)	(CAVLC, CABAC)	CABAC
configfile.h	yuv_format	3	
	lossless_qpprime_y_zero_flag	1	
Mode_ decision.c	enc_mb→valid[I4MB]	1	
	enc_mb→valid[I16MB]	0	

Table 3. Comparisons of H.264-LS image coding methods with JPEG-LS (outside test images).

A: CAVLC B: CABAC Images 768 kB	JPEG -LS	H.264-LS			H.264-LS_DPCM			Two-layer H.264-LS
		A	B	T_{Block}	A	B	T_{DPCM}	B
		kB			kB			kB
Map1	546	529	633	22.26	398	458	21.88	423
Map2	464	465	484	21.92	349	322	20.41	347
Map3	544	427	497	20.77	306	337	21.36	324
Map4	557	477	539	21.74	372	401	21.46	375
Scenery1	477	550	645	21.38	412	415	21.01	421
Scenery2	411	471	508	20.41	328	302	20.05	335
Scenery3	450	434	462	21.29	318	312	20.92	324
Satellite1	380	389	395	20.35	304	281	20.04	306
Satellite2	452	401	447	21.12	288	291	20.71	289
Satellite3	459	398	433	21.15	279	281	20.76	283
photo1	449	371	366	20.65	279	256	20.45	298
Photo2	383	397	379	20.07	378	345	19.98	399
Photo3	450	422	465	20.9	290	281	20.49	301
Photo4	401	468	535	20.74	331	319	20.28	323
Photo5	347	348	347	20.1	231	212	19.60	246
Paint1	362	403	414	20.21	276	248	19.77	266
Paint2	421	440	464	20.78	300	284	20.33	291
Paint3	380	381	378	20.36	258	231	19.86	260
Paint4	371	411	410	20.34	276	247	19.86	269
Paint5	342	358	341	20.14	240	215	19.66	240
Average	**432**	**427**	**457**	N/A	**311**	**302**	N/A	**316**
Average with AEC	**N/A**	**427**			**296**			**N/A**
AEC correct rate		**80%**			**90%**			

Table 4. Performance improvement by the proposed AEC and two-layer H.264-LS algorithms.

Lossless image coding		H.264-LS	H.264-LS_ DPCM	Two-layer H.264-LS
		AEC		
Lossy and near lossless coding		No	No	Yes
AEC correct rate for 40 images (inside and outside tests)		88	93	**N/A**
Computational complexity		Low	Low	High
Improvement with Initial coder (compression rate)	Only using CABAC	6.3 %	1.8 %	**N/A**
	Only using CAVLC	1.4 %	5.7 %	
Improvement with H.264-LS (CABAC) (compression rate)		6 %	28.4 %	20 %
Improvement with H.264-LS_DPCM (CABAC) (compression rate)		– 26.1 %	1.8 %	– 5.1 %

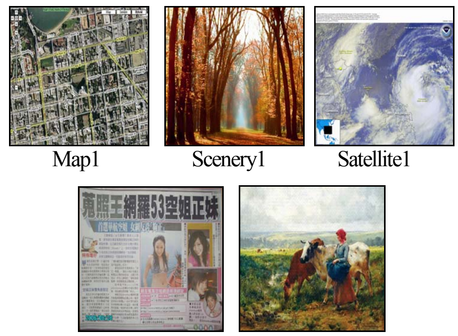


Fig. 8. out-side test images.

REFERENCES

[1] William B. Pennebaker and Joan L. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1992.

[2] D. S. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560 – 576, Jul. 2003.

[4] K. Wahid, V. Dimitrov, and G. Jullien, ”New Encoding of 8x8 DCT to make H.264 Lossless,” *IEEE Asia Pacific Conference Circuits and Systems* (APCCAS), pp. 780 – 783, Dec. 2006.

[5] S. Takamura and Y. Yashima, “H.264-based Lossless Video Coding Using Adaptive Transforms,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (ICASSP ‘05). vol. 2, pp. 301–304, Mar. 2005.

[6] Y. L. Lee, K. H. Han, and G. J. Sullivan, “Improved Lossless Intra Coding for H.264/MPEG-4 AVC,” *IEEE Trans. on Image Processing*, vol. 15, no. 9, pp. 2610 – 2615, Sep. 2006.

[7] H. Schwarz, D. Marpe and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620 – 636, Jul. 2003.

[8] Inside and out-side test Images downloads: http://sipi.usc.edu/database/database.cgi?volume=misc. http://decsai.ugr.es/cvg/dbimagenes/c512.php. ftp://140.116.163.181, username: testimages, password: testimages.

[9] Draft Text of H.264/AVC Fidelity Range Extensions Amendment, Joint Video Team, DOC. JVT-L047, Redmond, USA, July 2004.

[10] Software downloads: http://www.hpl.hp.com/loco. http://www.morgan-multimedia.com.