Towards a Stringent Bit-Rate Conformance for Frame-Layer Rate Control in H.264/AVC

Evan Tan[1] and Jing Chen[2]

[1] School of Computer Science and Engineering,
University of New South Wales, NSW 2052, Australia
evant@cse.unsw.edu.au
[2] National ICT Australia, Neville Roach Laboratory,
223, Anzac Parade, Kensington, NSW 2052, Australia
jing.chen@nicta.com.au

Abstract. This paper presents a novel frame-layer rate control technique that adaptively determines the frame complexity for bit allocation in order to satisfy the target bit-rate constraints without degrading the decoded video significantly. To do this, we first obtain the edge energy of each frame to measure the frame complexity as well as to determine the weighting of a frame for bit allocation. We then present a new bit-rate traffic model for bit allocation to achieve a better conformance to the target bit-rate. Finally, we integrate the edge energy complexity measure into the rate-quantization (R-Q) model. Our results shows robust improvements over the current rate control methods adopted in H.264/AVC in terms of meeting the target bit-rate as well as determining the quality of the decoded video.

Keywords: bit allocation, complexity measure, frame-layer, H.264/AVC, linear R-Q model, rate control.

1 Introduction

The rate control component regulates the coded video bit-stream in order to meet the network bandwidth and buffer constraints as well as to enhance the video quality as much as possible. This makes rate control one of the key components of a video coder especially in video streaming applications. A typical rate controller first allocates a target number of bits for each frame based on a bit budget. Then for each frame a quantization parameter (QP) is selected either for the whole frame (frame-layer) or for each macroblock (MB layer) based on some rate-quantization (R-Q) model in order to meet the specified target bits.

One of the main issue with rate control in H.264/AVC is that the the bit allocation and QP selection are conducted before the selection of INTER and INTRA modes. This means that various vital information about a frame, such as the mean absolute difference (MAD), is not readily available to the rate controller. As a consequence, many current rate control techniques make use of different kinds of estimates to obtain information about the frame.

H.H.S. Ip et al. (Eds.): PCM 2007, LNCS 4810, pp. 128–137, 2007.

The original rate control scheme that was adopted by the H.264/AVC standard and proposed by Li et al [1] solves this issue by performing a linear prediction of a P-frame's MAD value based on the MAD of the previous P-frame. However, this approach has problems handling scene changes within a video [5]. And the rate control algorithm is only performed on P-frames, while I-frames and B-frames have their QPs estimated based on the QPs calculated for P-frames without consideration of the actual characteristics of the I-frames and B-frames [2]. This makes it only ideal for videos with IPPP group-of-pictures (GOP) format.

Recently, Leontaris and Tourapis [2] have attempted to fix this problem by introducing a complexity measure for P-frames and made use of complexity ratio parameters to determine the complexity of I-frames and B-frames. This technique made the adopted rate control algorithm much more compliant to the bit-rate constraints. However, sudden complexity changes such as scene changes are not handled implicitly. Furthermore, the performance of the improved algorithm is strongly correlated to the fixed parameters introduced, which has to be tuned for various videos. Their approach also assumed the total size of the sequence is available at the start, which may not always be the case (e.g. for real-time video communication).

Liu et al [5] and Yu et al [9] proposed a technique that uses preanalyzed information from the video. This approach is ideal for the transmission of stored videos like video-on-demand, but is normally not suitable for applications such video-conferencing. Other approaches made use of complexity measures that are predicted [8,10] or by using some image processing techniques [4,6] or histogram difference techniques [7]. All these approaches have shown that scene change can be detected reasonably well, however, none of them account for the differences between I-frames, P-frames and B-frames in their complexity measures. Normally, I-frames uses more bits than P-frames which in turn uses more bits than B-frames, but this relationship might change in a high motion video sequence due to the increased intra-coded macroblocks (MB) being introduced into the frame. A proper complexity measurement scheme should take into account of *all* these.

Additionally, current approaches made use of the fluid flow traffic model with each frame assumed to take up R_B/R_F bits, where R_B is the channel bit-rate and R_F is the frame rate. Some approaches set a fixed upper bound on target bits [7,8], this is not a proper approach as the bit-rates may vary dramatically in high-motion videos. Other approaches [1,2,6,9,12] made use of the number of remaining bits in a GOP as an upper bound. The problem with making use remaining GOP bits is that it may cause the allocated bits to exceed the target bit-rate.

To illustrate this problem, we define a *bit-rate period* to be the amount of bits available for the bit allocation of R_F frames in one second (time needed for encoding is assumed to be negligible here for simplification), in this case it would be R_B bits. Let us assume the GOP structure is defined as a I-B-P-B format, Fig. 1 then illustrates this setup.

Suppose a scene change occurs on 6^{th} frame of the second GOP (unshaded section) resulting in a high complexity value for that frame. Since the second GOP only has an I-frame (5^{th} frame) coded, the 6^{th} frame could potentially use up a large fraction of the remaining GOP bits. This will cause the rate controller to allocate more bits than allowed within a bit-rate period, thus exceeding the bit-rate constraint.

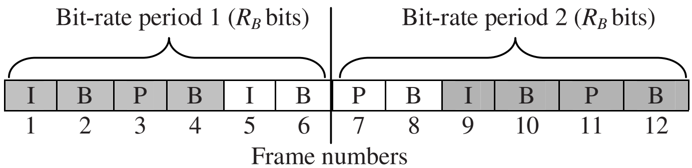

Fig. 1. An illustration of a bit-rate period imposed on the IBPB GOP structure

In this paper, we address these issues by firstly, calculating the edge energy of each frame as a form of perceptual complexity measure and make use of this to handle bit-rate variations due to scene change as well as to allocate bits to a frame with respect to the complexities of other frames within a bit period. Then, we define a bit-rate traffic model based on a bit-rate period to ensure that the bit allocations by the rate controller meet the target bit-rate. Finally, we modify the linear quadratic R-Q model [3] to account for the different bit allocations for different frames.

The rest of this paper is organized as follows. Section 2 describes how the complexity measure of a frame is calculated and weighted with other frames. Section 3 introduces the bit-rate traffic model and shows how frame bit allocation is performed. Section 4 describes our modification to the R-Q model to account for the different frame complexity. Section 5 discusses the experiments we performed. Finally, Section 6 concludes this paper.

2 Frame Complexity Measure

2.1 Edge Energy

In our proposed method, we made use of the edge energy extracted from a frame to calculate the frame complexity. This is because the AC coefficients represent edge information, so a frame with higher edge energy would tend to imply containing more AC coefficients which typically means that the I-frame would end up using more bits. Furthermore, motion information can be represented by the localized differences of edge energy between frames, as edge energy tends to change more when there is high motion.

To calculate the edge energy of a frame, we made use of the edge filters by Won et al [11]. Our modified algorithm starts by linearly quantizing the Y component of the frame into 128 levels as a way of noise removal. We then set an image block to be of size 8x8. Given $m_v(i,j,k)$, $m_h(i,j,k)$, $m_{d\text{-}45}(i,j,k)$, $m_{d\text{-}135}(i,j,k)$ and $m_{nd}(i,j,k)$ represents the vertical, horizontal, 45° diagonal and 135° diagonal edge magnitudes respectively for the *(i,j)*th image block on frame *k*. The edge energy can be calculated by:

$$B_E(i,j,k) = \sum m_x(i,j,k) \quad x \in \{v, h, d\text{–}45, d\text{–}135, nd\} \ , \tag{1}$$

Given a frame with *MxN* image blocks, the edge energy of an I-frame is:

$$E_I(k) = \sum_{j=0}^{M} \sum_{i=0}^{N} B_E(i,j,k) \ , \tag{2}$$

Let ρ be the previous anchor frame, then the edge energy of a P-frame is:

$$E_P(k) = \sum_{j=0}^{M}\sum_{i=0}^{N}\left|B_E(i,j,k) - B_E(i,j,\rho)\right| , \tag{3}$$

Let η be the next anchor frame, then the edge energy of a B-frame is:

$$E_B(k) = \sum_{j=0}^{M}\sum_{i=0}^{N}\frac{\left|B_E(i,j,k) - B_E(i,j,\rho)\right| + \left|B_E(i,j,k) - B_E(i,j,\eta)\right|}{2} , \tag{4}$$

Finally, the complexity measure of frame K is calculated by:

$$C(k) = \frac{E_x(k)}{\left(\frac{1}{k-1}\right)\sum_{l=0}^{k-1}E_x(l)} \quad x \in \{I,P,B\} . \tag{5}$$

3 Bit Allocation

3.1 Bit-Rate Traffic Model

We proposed a traffic model based on the actual bit-rate of the system as illustrated in Fig. 2.

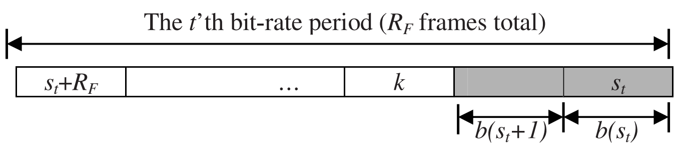

Fig. 2. Illustration of the bit-rate traffic model for one bit-rate period

Given a frame rate R_F, frame s_t as the frame at the start of the t'th bit-rate period and the instant available bit-rate $R_B(k)$ for the current frame k where $k \geq 0$. The available bits for frame s_0 in the first bit-rate period are:

$$A_0(s_0) = R_B(s_0) , \tag{6}$$

If the actual bits used by an encoded frame k is $b(k)$ then the bits left for frame allocation for frame k in the t'th bit-rate period are:

$$A_t(k) = A_t(k-1) - b(k-1) + (R_B(k) - R_B(k-1)) , \tag{7}$$

Subsequently, frame s_{t+1} on the $t+1$'th bit-rate period has its bit allocation updated as:

$$A_{t+1}(s_{t+1}) = R_B(s_{t+1}) + A_t(s_t + R_F) - b(s_t + R_F) . \tag{8}$$

3.2 Frame-Layer Bit Allocation

To determine the bits allocated to a frame, we first calculate the frame complexity value as described in section 2. As the complexity of the remaining frames to be encoded in the bit period is not known beforehand, we estimate it by using the mean complexity value of each I/P/B-frame.

To do this, a complexity value sliding window for *each* I/P/B-frame is maintained. The mean complexity value for each I/P/B frame is then calculated by averaging the values in the sliding windows. In our experiments, the sliding window sizes were set to 2 for I-frame complexity and 3 for both P and B frame complexity. This sliding window technique is used to make the system more responsive to bit allocation changes due to scene change.

The target bits for frame *k* on the *t*'th bit-rate period is then calculated by:

$$T(k) = \frac{C(k)}{NR_I \cdot AvgC_I + NR_P \cdot AvgC_P + NR_B \cdot AvgC_B} \times A_t(k) \times \frac{T(k-1)}{b(k-1)} \cdot \quad (9)$$

Where $AvgC_{I/P/B}$ is the mean complexity for I/P/B-frame and $NR_{I/P/B}$ are the remaining number of I/P/B-frames left to code in the bit-rate period.

4 QP Selection

QP selection is conducted using the quadratic R-Q model [3]. Three quadratic models are used for each I-frame, P-frame and B-frame respectively as the linear prediction model is different for each frame type. For I-frames, the model is:

$$\frac{T(k) - h(k)}{C(k)} = \frac{a_1}{QP(k)} + \frac{a_2}{QP^2(k)} , \quad (10)$$

While the model for P-frames as well as the model for B-frames is:

$$\frac{T(k) - h(k)}{\alpha \cdot PMAD(k) + (1-\alpha) \cdot C(k)} = \frac{a_1}{QP(k)} + \frac{a_2}{QP^2(k)} \cdot \quad (11)$$

Where *h(k)* is the header bits, a_1 and a_2 are the first and second order coefficients respectively, *QP(k)* is the quantization level for frame *k*, *PMAD(k)* is the linearly predicted mean absolute value (MAD) for frame *k* as defined in [1] and α is a weighting factor (set to 0.5 in our experiments).

5 Experimental Results

5.1 Setup

We tested our proposed method on seven different video sequences of CIF size, comprising of both high and low motion contents. The H.264/AVC reference software JM12.2 was used to conduct our simulations. RD optimizations were turned off in the software and the GOP structure was specified as I-B-B-P-B-B-P-B-B (GOP size of 9).

We ran our simulations with a frame rate of 15Hz with no frame skipping and at a constant bit-rate. We then compared our proposed method, named here as RC4, with the original adopted H.264/AVC rate control scheme [1], called RC0 here, as well as the modified H.264/AVC rate control scheme by Leontaris and Tourapis [2], called RC3 here. The frame-layer rate control was enabled for RC0 and RC3. The parameters for RC3 were fixed with RCISliceBitRatio set to 1, RCBSliceBitRatio0 set to 0.5, RCBoverPRatio set to 0.45 and RCIoverPRatio set to 3.8. Hierarchial coding was disabled.

5.2 Satisfying the Bit-Rate Constraint

To check if the method meets the bit-rate constraint at the given frame rate, we summed up all the actual bits used by the frames in a bit-rate period (the actual bits used by every set of 15 frames in this experiment). Note that the assumption made here is that the coding time is negligible. This assumption is made purely for an easier comparison between the different methods. The actual mean bit-rate is computed along with the bit-rate deviation (error) for each method as shown in Table 1. The breakdown of the actual bits used for each bit-rate period for Foreman is shown in Fig. 3.

Discussion. RC0 shows a large deviation (almost 4 times) from the target bit-rate, this highlights the inability of RC0 to do a proper rate control on I-frames and B-frames. In contrast, RC3 shows a much better conformance to the bit-rate and frame rate constraints compared to RC0. However, RC4 still outperforms RC3 by a fair amount. The main reason for this is the problem of using the number of remaining bits in a GOP as an upper bound as discussed previously. Also the larger bit-rate variation of RC3 due to frequent scene changes is evident in the results for high motion sequences (i.e. Football and Stefan), while our proposed method shows a much smaller bit-rate variation in the same high-motion sequences.

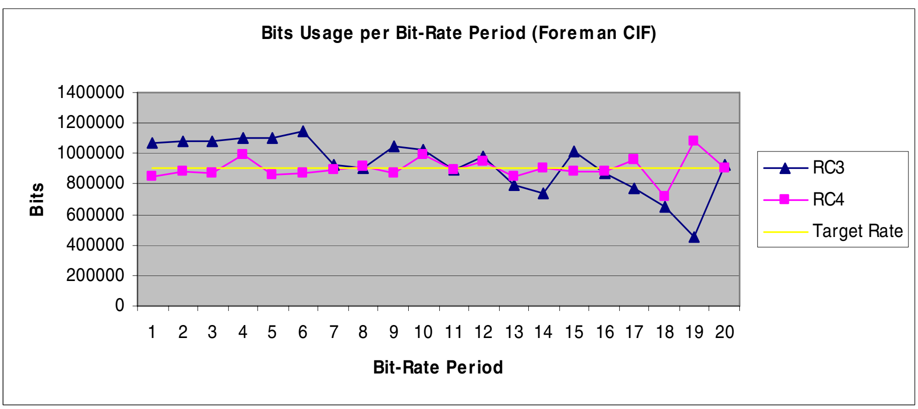

Fig. 3. The actual bits used for each bit-rate period for Foreman. RC0 is not shown here as the excessive bit-rates it generates skew the graph.

Table 1. Results showing the actual mean bit-rates and error of the proposed method (*RC4*), the original H.264/AVC method (*RC0*) and the modified H.264/AVC method (*RC3*)

Sequence	Target Rate (kbps)	RC0		RC3		RC4	
		Actual Rate (kbps)	Error	Actual Rate (kbps)	Error	Actual Rate (kbps)	Error
Container	500	3460	5.920	513	0.026	497	0.006
	700	3987	4.696	704	0.006	696	0.006
	900	4149	3.610	903	0.003	897	0.003
	1200	4682	2.902	1208	0.007	1193	0.006
	1500	4823	2.215	1524	0.016	1493	0.005
Football	500	1122	1.244	528	0.056	493	0.014
	700	1122	0.603	730	0.043	693	0.010
	900	1122	0.247	947	0.052	891	0.010
	1200	1963	0.636	1284	0.070	1195	0.004
	1500	1963	0.309	1603	0.069	1497	0.002
Foreman	500	4551	8.102	515	0.030	501	0.002
	700	4771	5.816	722	0.031	700	0.000
	900	5182	4.758	928	0.031	900	0.000
	1200	5618	3.682	1238	0.032	1198	0.002
	1500	5998	2.999	1545	0.030	1498	0.001
Mobile	500	7249	13.498	515	0.030	499	0.002
	700	7249	9.356	718	0.026	700	0.000
	900	7249	7.054	927	0.030	900	0.000
	1200	8945	6.454	1234	0.028	1198	0.002
	1500	8945	4.963	1538	0.025	1499	0.001
News	500	2720	4.440	508	0.016	497	0.006
	700	2925	3.179	711	0.016	697	0.004
	900	3015	2.350	914	0.016	896	0.004
	1200	3415	1.846	1226	0.022	1193	0.006
	1500	3474	1.316	1536	0.024	1491	0.006
Salesman	500	4178	7.356	512	0.024	500	0.000
	700	4636	5.623	717	0.024	699	0.001
	900	4742	4.269	917	0.019	899	0.001
	1200	5088	3.240	1227	0.023	1197	0.002
	1500	5180	2.453	1524	0.016	1497	0.002
Stefan	500	1834	2.668	547	0.094	503	0.006
	700	1834	1.620	764	0.091	696	0.006
	900	1834	1.038	980	0.089	895	0.006
	1200	2922	1.435	1304	0.087	1195	0.004
	1500	2922	0.948	1584	0.056	1491	0.006
Mean Overall Error:			3.795		0.036		0.004

Table 2. Results showing the mean Y PSNR of the proposed method (*RC4*) and the modified H.264/AVC method (*RC3*)

Sequence	**Target Rate (kbps)**	**RC3 Mean Y PSNR (dB)**	**RC4 Mean Y PSNR (dB)**	**PSNR Gain (dB)**
Container	500	38.84	38.38	-0.46
	700	40.27	40.98	0.71
	900	41.36	42.29	0.93
	1200	43.04	43.99	0.95
	1500	44.38	45.21	0.83
Football	500	35.59	35.67	0.08
	700	37.15	37.33	0.18
	900	38.43	38.77	0.34
	1200	39.89	40.34	0.45
	1500	41.25	41.68	0.43
Foreman	500	35.41	36.5	1.09
	700	36.74	37.78	1.04
	900	37.82	39.17	1.35
	1200	39.2	40.7	1.5
	1500	40.5	41.92	1.42
Mobile	500	25.39	26.44	1.05
	700	26.8	27.75	0.95
	900	28.25	29.35	1.1
	1200	30.1	31.68	1.58
	1500	31.69	33.08	1.39
News	500	41.34	42.13	0.79
	700	43.16	44.1	0.94
	900	44.59	45.56	0.97
	1200	46.11	47.77	1.66
	1500	47.89	49.28	1.39
Salesman	500	38.78	39.22	0.44
	700	40.09	41.3	1.21
	900	41.36	42.76	1.4
	1200	42.87	44.96	2.09
	1500	43.82	46.15	2.33
Stefan	500	29.32	29.7	0.38
	700	31.02	32	0.98
	900	32.44	33.37	0.93
	1200	34.41	35.23	0.82
	1500	36.19	37.28	1.09
			Mean Overall Gain:	0.98

5.3 Video Quality Test

We did a comparison on the decoded video quality output to show that our proposed method do not compromise heavily on the quality in order to meet the bit-rate and frame rate. We chose not to include RC0 in this test due to the fact that it deviates far too much from the target bit rate to make a fair comparison on the decoded video quality. We calculate the mean output PSNR of the Y-component of the frames and the PSNR gain of our proposed method compared to RC3 as shown in Table 2. The breakdown of the Y PSNR for each frame for Foreman is shown in Fig. 4.

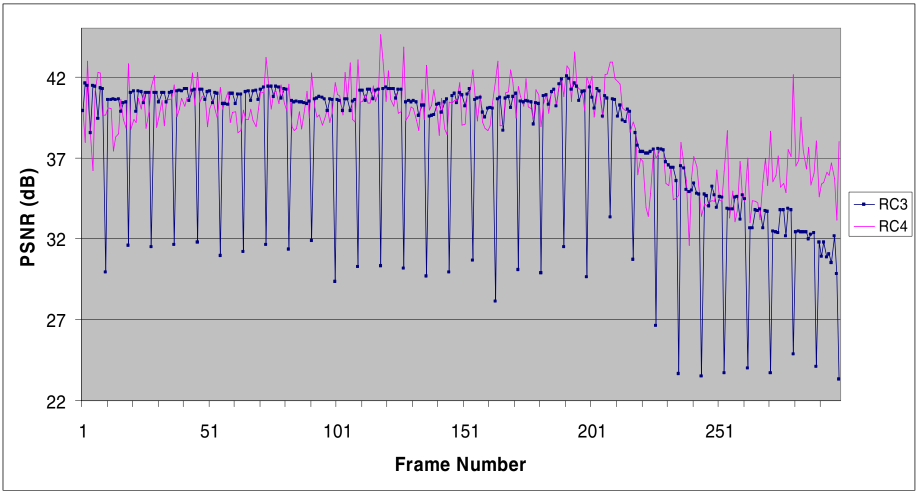

Fig. 4. The Y PSNR of each frame for Foreman

Discussion. The results show that our proposed method not only did not perform worse than RC3, but in general performed better by a fair amount in almost all cases with a mean PSNR gain of 0.98dB. RC3 requires its parameters to be tuned for each sequence and this is difficult to do in general. Using fixed parameter values causes RC3 to perform badly as shown in the results. Moreover, RC3 tends to allocate a smaller amount of bits for the I-frames, doing this may sometimes degrade the quality on subsequent P-frames and B-frames. Our proposed method, on the other hand, allocates bits purely based on the derived complexity measure of the frame, avoiding the issue of choosing parameters by providing an accurate estimated weighting to I-frames, P-frames and B-frames.

6 Conclusion

In this paper, we proposed a novel rate control scheme by using the edge energy of a frame to estimate the frame complexity and integrated it into the R-Q model. We also proposed a new bit-rate traffic model to replace the fluid flow traffic model. Results showed that our proposed method has a more stringent adherence to the target bit-rates without significantly sacrificing the quality of the video output. Additionally, our proposed method does not assume that any information on the whole video sequence is available, making it suitable for real-time video applications.

Acknowledgment. National ICT Australia (NICTA) is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

1. Li, Z.G., Pan, F., Lim, K.P., Feng, G.N., Lin, X., Rahardja, S.: Adaptive Basic Unit Layer Rate Control for JVT. JVT-G012, 7th JVT Meeting, Pattaya, Thailand (March 2003)
2. Leontaris, A., Tourapis, A.M.: Rate Control Reorganization in the Joint Model (JM) Reference Software. JVT-W042, 23rd JVT Meeting, San Jose, California, USA (April 2007)
3. Lee, H.J., Chiang, T., Zhang, Y.Q.: Scalable rate control for MPEG-4 video. IEEE TCSVT 10(6), 878–894 (2000)
4. Jing, X., Chau, L.P.: A novel intra-rate estimation method for H.264 rate control. In: IEEE ISCAS 2006, pp. 21–24 (May 2006)
5. Liu, Y., Li, Z.G., Soh, Y.C.: A Novel Rate Control Scheme for Low Delay Video Communication of H.264/AVC Standard. IEEE TCSVT 17(1), 68–78 (2007)
6. Yu, H., Pan, F., Lin, Z., Sun, Y.: A perceptual bit allocation scheme for H.264. In: IEEE ICME 2005 (July 2005)
7. Lee, J.H., Shin, I.H, Park, H.W.: Adaptive Intra-Frame Assignment and Bit-Rate Estimation for Variable GOP Length in H.264. IEEE TCSVT 16(10), 1271–1279 (2006)
8. Jiang, M.Q., Ling, N.: An improved frame and macroblock layer bit allocation scheme for H.264 rate control. In: IEEE ISCAS 2005, vol. 2, pp. 1501–1504 (May 2005)
9. Yu, H.T., Lin, Z.P., Pan, F.: An improved rate control algorithm for H.264. In: IEEE ISCAS 2005, vol. 1, pp. 312–315 (May 2005)
10. Lee, C.Y., Lee, S.J., Oh, Y.J., Kim, J.S.: Cost-Effective Frame-Layer H.264 Rate Control for Low Bit Rate Video. In: IEEE ICME 2006, pp. 697–700 (July 2006)
11. Won, C.S., Park, D.K., Park, S.J.: Efficient use of MPEG-7 edge histogram descriptor. ETRI Journal 24(1) (February 2002)
12. Li, Z.G., Zhu, C., Ling, N., Yang, X.K., Feng, G.N., Wu, S., Pan, F.: A Unified Architecture for Real Time Video Coding Systems. IEEE TCSVT 13(6), 472–487 (2003)