A Study on Joint Coding System using VF Arithmetic Code and BCH code

†Sukhee Cho ††Jihwan Park and †Ryuji Kohno

†Div. of Elec. & Comp. Eng., Faculty of Eng., Yokohama Nat'l Univ.,
79-5 Tokiwadai, Hodogaya, Yokohama, 240-8501 Japan
Email: {hee, kohno}@kohnolab.dnj.ynu.ac.jp
††Dept. of Computer Science, Pukyong National Univ.,
559-1, Daeyeon-Dong, Nam-ku, Pusan, 608-737 Korea
Email: jhpark@dolphin.pknu.ac.kr

Abstract

This paper is the research about a joint coding system of source and channel coding using VF(Variable-to-Fixed length) arithmetic code and BCH code. We propose a VF arithmetic coding method with EDC(Error Detecting Capability) and a joint coding method in that the VF arithmetic coding method with EDC is combined with BCH code. By combining both the VF arithmetic code with EDC and BCH code, the proposed joint coding method corrects a source codeword with t-errors in decoding of BCH code and carries out a improvement of the EDC of a codeword with more than $(t+1)$-errors in decoding of the VF arithmetic coding with EDC. We examine the performance of the proposed method in terms of compression ratio and EDC.

1 Introduction

There has been an increasing demand for efficient and reliable digital data transmission and storage systems. This demand has been accelerated by the emergence of large-scale, high-speed data networks for the exchange, processing, and storage of digital information. A great deal of effort has been expended on the problem of devising efficient encoding and decoding methods for error control in a noisy environment[3].

Recent developments have contributed toward achieving the reliability required by high-speed digital systems, and the use of coding for error control has become an integral part in the design of modern communication systems and digital computers. It was proposed in [1] how to integrate error detection into ordinary arithmetic coding. In order to adjust the unused coding space, at regular points in the encoding procedure, the encoder reduces the current coding interval by a fixed reduction factor R. If the transmitted string is in the unused coding space during decoding, the decoder determines that a communication error(s) occurred. It needs to predict the number of bits that were processed before an error is detected and request the retransmission of those bits before the string departs from the current interval. It guarantees only detection of single errors using exclusive OR's operation. If burst errors occur, it doesn't guarantee the detection of errors.

Arithmetic coding for data compression has gained widespread acceptance as the right method for optimum compression when used with a suitable source model[2]. Unlike Huffman coding, which recovers quickly from errors[2], a single inverted bit in arithmetically compressed text corrupts all subsequent output. Therefore, recently there have been some researching about arithmetic coding with some form of error control[4-6].

VF arithmetic coding methods in [4-6] combine the advantage of an ordinary stream arithmetic code with the simplicity of a block code. These coding methods parse an input sequence into variable length substrings according to symbol probability and encode each substring into a fixed length output codeword. NPVFAC(Non-proper VF arithmetic coding) in [6] is the method that achieves more efficient compression rate than that of other VF arithmetic codings in [4] and [5]. In these VF arithmetic codings, transmission errors(inverted bit errors) do not propagate infinitely and are restricted to the block in question, because the decoders do not lose synchronization of codewords with fixed length. However, these VF arithmetic codings don't have an error detecting and correcting capability.

Therefore, we propose the source coding method that can detect errors using NPVFAC. It has to detect the errors within a fixed-length codeword, because codeword length is fixed and each codeword is independent, unlike arithmetic coding. Hence, we propose VF arithmetic coding with EDC that uses the adaptive reduction factor AR and cyclic exclusive OR's operation.

We also consider a joint coding system like Fig. 1 with more efficient compression ratio and error-detecting/correcting capability. VF arithmetic codes have efficient compression ratio and can be easily combined with the channel codes because their codewords have fixed lengths. The BCH codes are easily defined in terms of the roots of their generator polynominals and are also capable of correcting all random patterns of t-errors. Their decoders are simple and easy to implement with reasonable amount of equipment[7]. Therefore, we also propose a joint coding system in that the VF arithmetic code with EDC is combined with the BCH code.

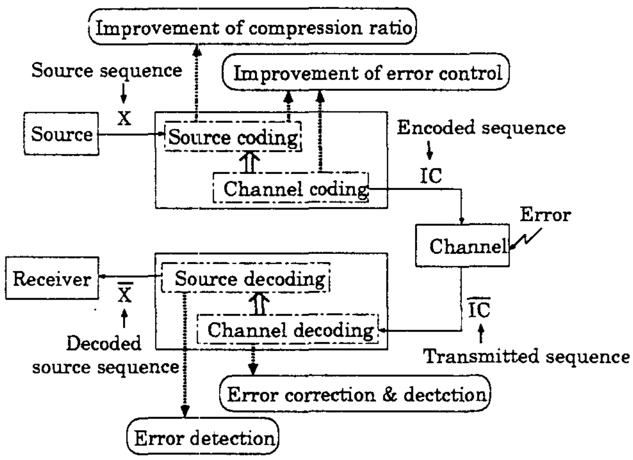

Figure 1: The proposed joint coding system.

2 VF(Variable-to-Fixed Length) Arithmetic coding

The VF arithmetic coder parses an input sequence into variable length substrings according to symbol probability and encodes each substring into a fixed length output codeword. A substring of the input is represented as an integer in the output codeword set $\{0, 1, \cdots, 2^w - 1\}$, where w is the codeword length in bits. Therefore, it is necessary that VF arithmetic coder should stop the splitting of the codeword set and reinitialize the codeword set if the codeword set has one codeword, like NPVFAC and BAC, or when a *collision* occurs, like AB-coding[4-6]. NPVFAC is the coding method that has the most efficient compression ratio among the VF arithmetic codings[6].

In this paper, let the input alphabet $A = \{a_1, a_2, \cdots, a_q\}$ be $p_l \geq p_{l+1}$ for $p_l = \text{Prob}(a_l)$. The cumulative probability C_l is as follows : $C_0 = 0$, $C_l = \sum_{i=1}^{l} p_i$, for $l = 1, 2, \cdots, q$. NPVFAC defines a new input alphabet subset $A' = \{a_1, a_2, \cdots, a_{q'}\}$ $(1 \leq q' \leq q)$ based on the number of codewords in the current codeword set and input symbol probability in original input alphabet set A when the codeword set can't be split completely for A. Therefore, the size of A' in splitting of the current codeword set is less than or equal to the size of A' in splitting of previous codeword set.

NPVFAC splits the codeword set into q' nonempty disjoint subsets and stops splitting when $K = 1$. K is the number of codewords in the current codeword set and the initial $K = 2^w$ and K_l is the number of codewords in a subset which is mapped at next input symbol a_l. K_l is computed with $K_{q'}, K_{q'-1}, \cdots, K_1$ in that order. Then, it outputs a codeword and reinitializes the codeword set. Considering NPVFAC's procedure from a viewpoint of parsing trees, the root has q children for A. The others have $q'(\leq q)$ children for a newly defined $A' = \{a_1, a_2, \cdots, a_{q'}\}$. Therefore, NPVFAC parses the codeword set completely for A'.

3 VF Arithmetic Coding with Error-Detecting Capability

We consider the source coding method that can detect errors using the Non-proper VF arithmetic coding.

A fixed redundancy for a codeword is introduced by adjusting the number of codewords in the current codeword set so that some codewords are never used by the encoder. During decoding, if a received codeword corresponds to a forbidden one the decoder determines that a communication error(s) occurred in the codeword and requests retransmission of the codeword. In the encoding procedure, in order to adjust the number of unused codewords the proposed coding scheme reduces the current codeword set by adaptive reduction factor AR. On decoding, the same reduction process is performed. The received codeword is checked after each reduction to see whether it lies within the reduced codeword set or not. If not, an error has occurred. Since codeword length is fixed and each codeword is independent, unlike the scheme in [2], VF arithmetic coding with error detecting capability has to detect the errors within a fixed-length codeword. Therefore, we propose the coding scheme that uses the adaptive reduction factor AR and cyclic exclusive OR's operation in NPVFAC.

3.1 Adaptive reduction factor AR

In VF coder, each input symbol is corresponded to various length within a fixed-length codeword in proportion to its probability. An input symbol with a high probability should be corresponded to a short length while one with a low probability should be corresponded to a long length. Therefore, we consider that the reduction factor is adaptive rather than fixed.

Let Rd_C be the number of unused bits(redundant bits) among each codeword length, wb its. Then, the added redundancy for compression ratio of NPVFAC, Rd_T, should be expected by eq. 1.

$$Rd_T \approx \frac{Rd_C}{w}. \tag{1}$$

The proposed scheme defines the number of unused codewords, NK, for the number of total codewords, 2^w, by eq. 2. NK in eq. 2 is initial value. When NPVFAC reinitializes the codeword

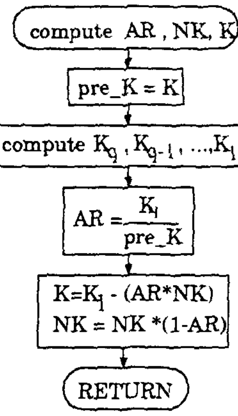

Figure 2: Algorithm of computations of AR, K and NK.

set, NK is also reinitialized by eq. 2.

$$NK = \frac{2^w \times Rd_C}{w}. \tag{2}$$

AR is determined according to K_l for K codewords in current codeword set which has been split by NPVFAC. K_l is the number of codewords in a codeword subset which is mapped at the next input symbol a_l. Hence, AR is computed by eq. 3.

$$AR = \frac{K_l}{K}. \tag{3}$$

Then, the proposed scheme doesn't use $AR \times NK$ codewords among K_l codewords. The algorithm in Fig. 2 shows the computations of AR, K and NK.

3.2 Cyclic exclusive OR's operation

We propose cyclic exclusive OR's operation in a codeword so that the errors in a codeword are detected within a fixed-length. In VF arithmetic coding, the closer the position of error bits to MSB(Most significant bit) of a codeword, the better the error-detecting capability. Therefore, we propose cyclic exclusive OR's operation in order to propagate the errors into the bits of the MSB's side although they occur in LSB's side.

Let sb be the position of the exclusive OR's starting bit except MSB in a codeword. The proposed scheme carries out exclusive OR for all bits except sb in a codeword. Exclusive OR's operation are carried out by the pair bits of $[(sb-1)\text{th}, sb\text{th}]$, $[(sb-2)\text{th}, (sb-1)\text{th}]$, $\cdots$, $[1\text{st}, w\text{th}]$, $[w\text{th}, (w-1)\text{th}]$, $\cdots$, $[(sb+2)\text{th}, (sb+1)\text{th}]$, respectively. Fig. 3 shows the procedure of cyclic exclusive OR in endoing.

4 Joint Coding of VF Arithmetic code with EDC and BCH code

We propose the coding method that implements the joint coding system like Fig. 1. This is done by encoding input symbols in the source code taking into consideration that channel coding should be carried out after source coding.

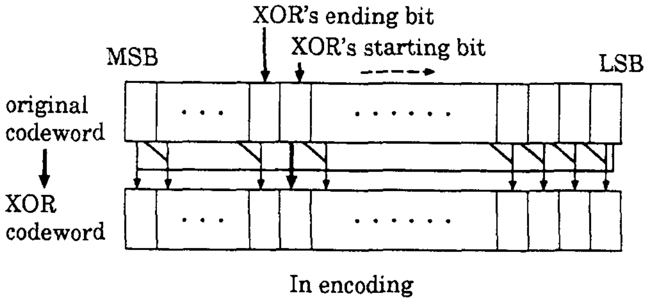

Figure 3: cyclic exclusive OR.

VF arithmetic codes have efficient compression ratio and can be easily combined with channel codes because the their codewords have fixed lengths. The BCH codes are are easily implemented with reasonable amount of equipment and are also capable of correcting all random patterns of t-errors by their decoders. Hence, we use VF arithmetic codes with EDC and BCH codes as source code and channel code, respectively.

Since VF arithmetic coding gets more efficient compression ratio when the codeword length is long, we consider a coding method that makes its codeword length longer by using the fact that (n,k)BCH coding should be carried out after VF arithmetic coding. Therefore, in the proposed joint coding system VF arithmetic coding sets the first codeword length to n bits and the rest codeword's lengths to $n'(\geq n)$bits even though all codewords length is $(n-k)$ bits in the original one; n is the number of the output codeword length of BCH code. Hence, the proposed coding system improves EDC for over $(t+1)$-errors within a codeword and corrects the errors for below t-errors within a codeword by the decoders of VF arithmetic code witn EDC and (n,k)BCH code, respectively.

NPVFAC stops splitting of the codeword set when $k=1$ and reinitializes it. However, VF arithmetic coding with EDC in the proposed joint coding system stops splitting of codeword set if the number of codewords, K, in the current codeword set is satisfied by eq. 4, because $(n-k)$ bits among n bits codeword length should be used as parity check bits of (n,k)BCH code.

$$log_2 K < n - k. \tag{4}$$

Let $W(T)$ be the codeword subset that is created by Tth splitting of the codeword set and $K(T)$ be the number of codewords in $W(T)$. If $K(T)$ is satisfied by eq. 4, in encoding Tth input symbol, VF arithmetic coding outputs a current codeword, then reinitializes the codeword set with $\{0,1,\cdots,2^{n'}-1\}$ by the next codeword length n' bits shown in eq. 6, and continues the encoding with Tth symbol again.

$$S = \lfloor log_2 K(T-1) \rfloor - (n-k), \tag{5}$$

$$n' = n + S. \tag{6}$$

The next codeword length n' is determined by eq. 6 because the number of available bits for $K(T-1)$ codewords in $W(T-1)$ is $\lceil log_2 K(T-1) \rceil$ bits and $(n-k)$ bits among them are used as parity check bits of the (n,k)BCH code. In this case, each codeword isn't independent to the previous codeword and a transmission error is capable to propagate infinitely. Therefore, the new solution is devised to avoid this problem as described below.

VF arithmetic coding needs at least $\lceil -log_2 p_l \rceil$ bits in order to encode an input symbol a_l in an input alphabet set A. The minimum number of bits is $\lceil -log_2 p_1 \rceil$ bits over symbol a_1 since

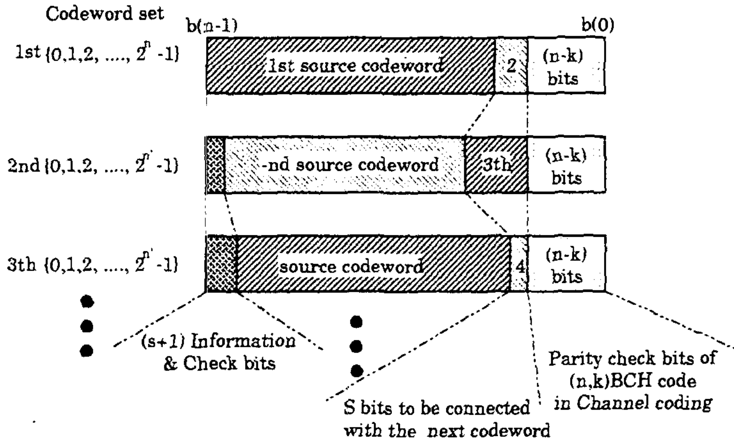

Figure 4: Organization of codewords.

A is sorted by symbols probabilities in a descending order. We define the parameter s by eq. 7.

$$s = \left\lceil \frac{log_2 K(T-1) - (n-k)}{-log_2 p_1} \right\rceil . \tag{7}$$

Let n' bits in eq. 6 be labeled by $b(n+(S-1))$, $b(n+(S-2))$, $\cdots$, $b(n)$, $b(n-1)$, $\cdots$, $b(1)$, $b(0)$ over from MSB to LSB. VF arithmetic coding sets $b(n-1) = b(n-2) = \cdots = b(n-s) = 1$ and $b(n-(s+1)) = 0$ in the next codeword bits to represent the amount of bits that are added to the original codeword length n. Hence, if the next codeword length is determined by eq. 8, then each codeword is independent to its previous codeword. Figure 4 shows the organization of codewords in the proposed joint coding system.

$$n' = n + S - (s+1). \tag{8}$$

In decoding of VF arithmetic coding, the decoder computes n' by eq. 8. It can obtain the value of s by counting the ones from $b(n-1)$ bit until a zero is founded. The amount of ones is equal to s. If the value of s by eq. 7 in the previous codeword is not equal to the value of s from counting 1's in current codeword, it means that some errors have occurred. Therefore, these $(s+1)$ bits have the functions of not only information-bits over the current codeword but also error-check-bits over the previous codeword and the current codeword.

5 Analysis of the Performance

We examined the performance of the proposed coding scheme for four source files from the Calgary Compression Corpus [2]. We represent the compression ratio as output length to input length and EDR(error detection ratio) as the number of detected codewords to the number of error codewords. We set $w = 32$ bits.

5.1 Results in VF arithmetic coding with EDC

We investigated the performance of NPVFAC with EDC using AR and cyclic exclusive OR operation.

Table 1: The added redundancy for compression ratio of NPVFAC with EDC.

Rd_C (bit)	The expected value (%)	Result of simulation(%)			
		bib	geo	obj1	paper1
5	15.63	15.29	19.56	15.50	15.45
6	18.75	19.43	21.61	18.58	19.03
7	21.88	22.94	23.14	22.14	22.84

Table 2: Comparison of compression ratio.

Output length/Input length

Data	NPVFAC	Conventional method	Proposed method
bib	0.6781	0.8089	0.8038
geo	0.7492	0.9395	0.9449
obj1	0.8038	0.9787	0.9533
paper1	0.6442	0.7770	0.7689

• Conventional method : NPVFAC → (31,26) BCH

First, about the added redundancy for the compression ratio of NPVFAC, we compared the expected ones by eq. 1 with the results of simulation. Table 1 shows that the values obtained from the simulation are near to the expected values, 5/32, 6/32 and 7/32, by eq. 1 in $Rd_c = 1, 2$and3, respectively. Therefore, it is noted that the proposed NPVFAC with EDC is capable to adjust the added redundancy Rd_T according to the error rate of the channel.

Next, we compared the proposed coding method with NPVFAC+(31,26)BCH that corrects single errors for each length of 31 bits. Again, NPVFAC+(31,26)BCH indicates the coding method that carries out the (31,26)BCH code over the generated codewords from NPVFAC. It is noted in Table 2 that in $Rd_c = 6$ bits, the compression ratio of the proposed coding method is almost the same to that of NPVFAC+(31,26)BCH.

Figure 5 shows that EDRs of the proposed method in $Rd_c = 6$ bits are about 97% for four files in case single errors occur within a codeword. The proposed scheme can also detect codewords with more than one error. It is noted in Fig. 6 that EDRs are about 97% although there are errors of 20 bits in a codeword.

5.2 Results in joint coding of VF arithmetic code with EDC and BCH code

We examined the performance of the proposed joint coding system and the conventional coding system in terms of compression ratio. here, we use (31,26)BCH code as the channel code in the proposed joint coding. The conventional method is the coding system that carries out the (31,21)BCH code over the generated codewords from NPVFAC with EDC. It is noted that the proposed and conventional coding systems are capable to correct one error and two error within a codeword, respectively.

Table 3 shows that the compression ratio of the proposed joint coding system in Rd_C = 1, 2 and 3 is more efficient that of the conventional one. Figures 7 ~ 10 show EDRs of the proposed joint coding system over all source files are about 98.8% for codewords with more than one error. It is also noted that the proposed coding system can detect errors with about 99.6 ~ 99.9% EDRs in $Rd_C = 3$ bits and they are almost near to those of conventional one.

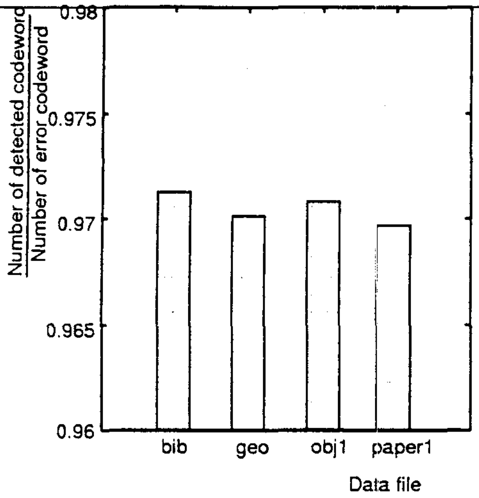

Figure 5: EDRs for single error codewords in Rd_C = 6bits.

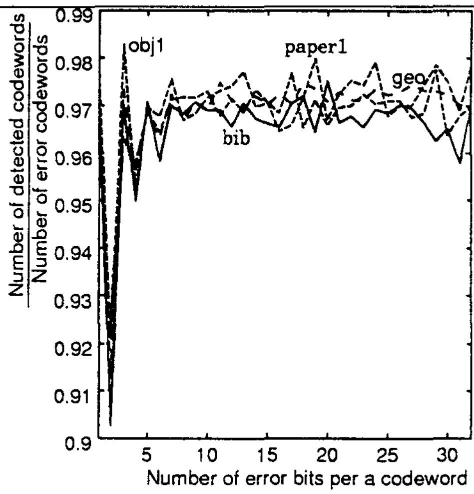

Figure 6: EDRs for more than one error in Rd_C = 6bits.

Table 3: Comparison of compression ratio.

Output length/Input length

Data	Conventional method	Joint coding method		
		NPVFAC with EDC + (31,26)BCH		
	NPVFAC → (31,21)BCH	$Rd_C = 1$	$Rd_C = 2$	$Rd_C = 3$
bib	1.0040	0.9080	0.9415	0.9737
geo	1.1099	0.9623	1.0192	1.0944
obj1	1.1974	1.0847	1.1383	1.1801
paper1	0.9705	0.9526	0.9236	0.8905

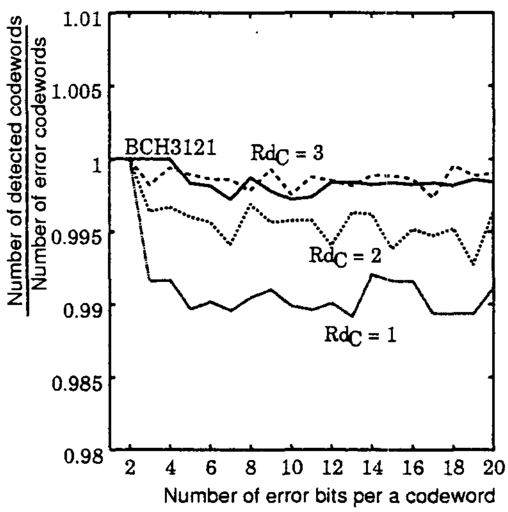

Figure 7: EDRs for 'bib'.

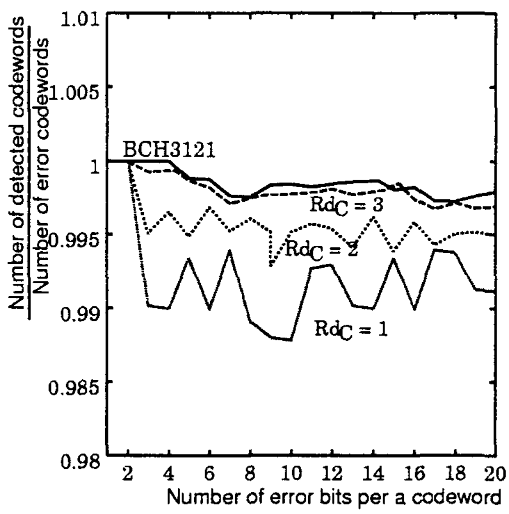

Figure 8: EDRs for 'geo'.

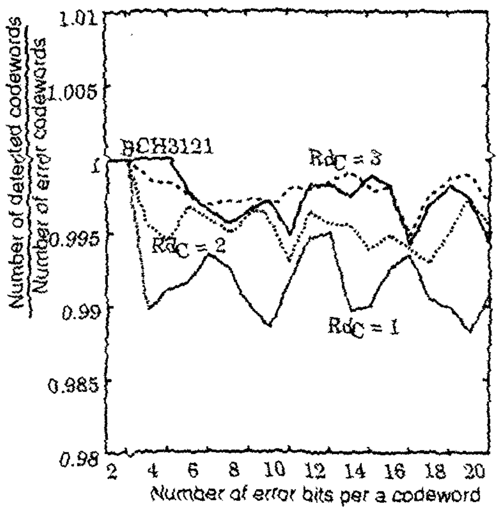

Figure 9: EDRs for 'obj1'.

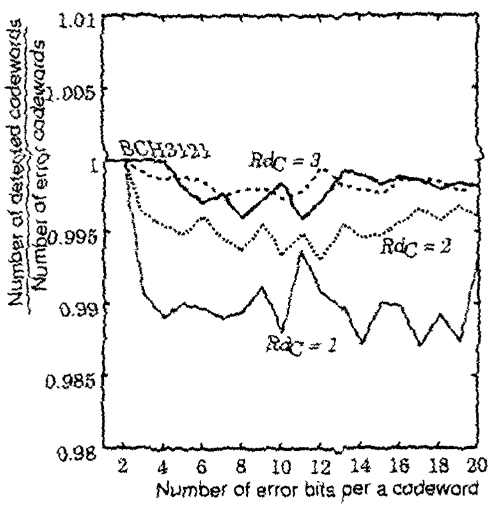

Figure 10: EDRs for 'paper1'.

6 Conclusions

In this paper, we proposed a VF arithmetic coding method with Error Detecting Capability that uses the adaptive reduction factor AR and cyclic exclusive OR's operation. The VF arithmetic coding method with EDC can detect errors with about 97% EDR in $Rd_C = 6$ bits and $w \approx 32$ bits even if there are more than one errors within a codeword, Therefore, we also proposed a joint coding method in that the VF arithmetic coding method with EDC is combined with BCH code. By combining both the VF arithmetic code with EDC and BCH code, the proposed joint *coding method corrects a source codeword with t errors in decoding of BCH code and carries* out a improvement of the EDC of a codeword with more than $(t+1)$-errors in decoding of the VF arithmetic coding with EDC. Further work to be solved is the study on a source coding with error correcting capability using how to select unused codewords by Hamming distance.

References

[1] Colin Boyd, John G. Cleary, "Integrating Error detection into Arithmetic Coding," IEEE Trans. on Comm. Vol. 45, No.1, 1997.1.

[2] T.C.Bell, J.K.Cleary, I.H.Witten, "Text Compression," p.318, Prentice Hall, 1990.

[3] J.W. Modestino, D.G. Daut, "Combiend Source-Channel Coding of Image," IEEE Trans. Comm., Vol.27, No.11, Nov. 1979.

[4] J.Teuholar, T.Raita, "Arithmetic Coding into Fixed-Length Codewords," IEEE Trans. Inf. Theory, Vol.40, No.1, pp.219-223, Jan.1994.

[5] G. Charles, Jr. Boncelet, "Block Arithmetic Coding for Source Compression," IEEE Trans. Inf. Theory, Vol.39, No.5, pp.1546-1553, Sept.1993.

[6] S.H.Cho, J.H.Park and R.Kohno, "Non-proper Variable-to-Fixed Length Arithmetic Coding," EICE, VolE81-A, No.8,pp.1739-1747, Aug. 1998

[7] M.Y.Rhee, "Error Correcting Coding Theory,", 1989