

Reed-Solomon Product Code의 에러 정정 능력 평가 방법

황 성 희, 이윤우, 한성후, 류상현, 신동호, 박인식
삼성 전자 디지털 미디어 연구소

The Performance evaluation of the Reed-Solomon Product Code(RSPC)

Sung-hee Hwang, Yoon-woo Lee, Sung-hyu Han, Sang-hyun Ryu, Dong-ho Shin, In-sik Park
Digital Media R&D Center Samsung Electroics Co., Ltd

Abstract - 광 디스크 channel상에서 RSPC의 error correction capability를 확률적인 계산 방법으로 계산하는 데는 많은 어려움이 있다. 그 이유는 바로 광 디스크 channel이 burst error channel이기 때문인데, 이 burst error를 어떻게 다루는가에 따라 그 error correction capability는 사뭇 달라진다. 이 논문에서는 Sony의 dust error distribution[1] 아래에서 4-state Morkov Chain[2]로 modeling하고 그 결과를 가지고 burst error를 channel의 특성과 ECC format의 특성에 맞게 제어할 수 있는 확률적인 계산 방법을 소개하고 그것을 simulation하고자 한다.

다음은 4-state Morkov chain으로 modeling한 것의 결과인 state transition probability이다.

$$\begin{aligned}
 P_{01} &= 2.7441 \times 10^{-3} & P_{10} &= 3.2236 \times 10^{-1} \\
 P_{02} &= 3.1102 \times 10^{-5} & P_{20} &= 3.7502 \times 10^{-2} \\
 P_{03} &= 7.3587 \times 10^{-6} & P_{30} &= 6.8840 \times 10^{-3} \\
 P_{12} &= 2.7441 \times 10^{-3} & P_{21} &= 3.2236 \times 10^{-1} \\
 P_{13} &= 3.1102 \times 10^{-5} & P_{31} &= 3.7502 \times 10^{-2} \\
 P_{14} &= 7.3587 \times 10^{-6} & P_{41} &= 6.8840 \times 10^{-3}
 \end{aligned}$$

1. 서 론

어떤 ECC format의 error correction capability를 평가하기 위해서는 우선 그 channel의 상태를 나타내는 error distribution이 있고 그 error distribution을 정량으로 다루기 위한 modeling 작업이 있어야 할 것이다.

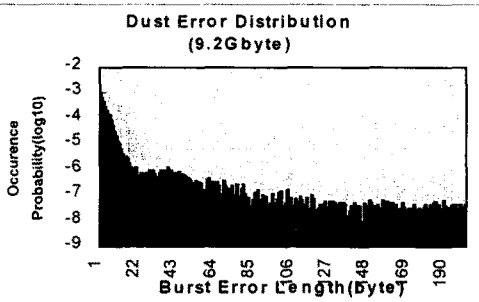


그림 1. Burst error length distribution[1]

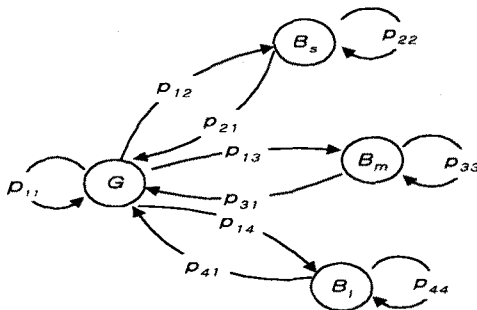


그림 2. 4-state Markov Chain[2]

4-state Morkov Chain으로 modeling한 것의 결과인 state transition probability들을 사용해서 (N1, K1, d1) × (N2, K2, d2) RSPC의 error correction capability를 평가해 보자. 여기서 di = Ni - Ki + 1, i = 1, 2.

2. 본 론

2.1 Decoding strategy

Decoder error가 없다고 가정을 하자.

- 1) Inner-code codeword (d1-1)/2이하의 에러가 존재할 경우에 정정, 그렇지 않은 경우에는 erasure 선언.
 - 2) 1)의 결과를 가지고 Outer-code codeword에서
 - 2-1) erasure의 개수가 d2-1이하일 경우에 erasure 정정 (PI→PO erasure 정정)
 - 2-2) erasure의 개수가 d2이상일 경우에 1)의 erasure 선언을 무시하고 (d2-1)/2이하의 에러가 존재할 경우에 정정, 그렇지 않을 경우에 decoder failure 선언. (PI→PO error 정정)
- RSPC의 가장 큰 장점은 반복 정정을 할 수 있다는 것이다. 위와 유사한 방식으로 반복 정정을 수행한다면 그 error correction capability는 더 향상될 것이다. 하지만 여기서는 2차 정정까지 정정하고자 한다.

2.2 정정 대상 세분화 하기

Modeling의 결과인 state transition probability로부터 우리는 j개의 연속적인 byte가 에러일 확률 P(i)를 참고 문헌 [2]에서 처럼 구할 수 있다. EF(j)을 하나의 error pattern에 의해 n개의 연속적인 inner-code codeword가 erasure로 선언되는 error pattern이라 하고, DF(j)을 EF(j)이 발생할 확률이라 하자(j ≥ 2). 예외적으로 DF(1)은 하나의 error pattern에 의해 연속적으로 inner-code codeword에서 erasure가 발생하는 경우가 아닌 단지 어떤 error pattern들에 의해 하나의 codeword만 erasure로 선언되는 확률이라고 정의하자. 즉, DF(1)은 m개의 error pattern들에 의해 (d1-1)/2 + 1symbol 이상의 에러가 발생하고, m-1개의 error patten들에 의해서는 (d1-1)/2symbol 이하의 에러가 발생할 확률을 m이 1 ~ (d1-1)/2 + 1까지 합한 확률이다. 그리고 DF(0)는 inner-code codeword를 정정할 확률이다. 그 때 우리의 P(i)를 이용해서 DF(j)을 다음과 같이 나타낼 수 있다.

$$j=0인 경우: DF(0) = 1 - \sum_{j=1}^{\infty} DF(j)$$

$$j=1인 경우:$$

① error pattern이 1개 일 때

$$DF(1)_1 = \sum_{i=(d-1)/2+1}^d (M+i-d) \cdot P(i) + \sum_{i=d+1}^{M+d-1} (M+d-i) \cdot P(i)$$

② error pattern이 2개 일 때

$$DF(1)_2 = \sum_{i+j=(d-1)/2, i, j < (d-1)/2+1}^M C_2 \cdot P(i) \cdot P(j)$$

error pattern이 3, 4, ..., (d-1)/2일 때 위와 똑 같은 방법으로 구하여 DF(1)을 구하면 다음과 같다.

$$DF(1) = \sum_{i=1}^{(d-1)/2} DF(1)_i$$

j ≥ 2인 경우:

$$DF(j) = \sum_{i=(j-2) \cdot M+d+1}^{(j-1) \cdot M+d-1} (i-(j-2) \cdot M-d) \cdot P(i) + \sum_{i=(n-1) \cdot M+d}^{n \cdot M+d-1} (n \cdot M+d-i) \cdot P(i)$$

Define : $[i]_j = [i_j][i_{j-1}] \dots [i_2]$

그 때 세분화된 정정 대상 ED $[i]_j$ 를 다음과 같이 정의할 수 있다.

$$ED [i]_j = \binom{M-(j-1) \cdot i}{(M-j) \cdot i, (j-2) \cdot i, \dots, i_2} C_{i_1} \cdot DF(j)^{i_1} \cdot C_{i_{j-1}} \cdot DF(j-1)^{i_{j-1}} \cdot \dots \cdot C_{i_2} \cdot DF(2)^{i_2} \cdot (DF(0) + DF(1))^{(M - \sum_{k=1}^j i_k)}$$

2.3 에러 정정

2.3.1 PI→PO erasure 정정

result2 $[i]_j$ 를 세분화된 정정 대상 ED $[i]_j$ 에서 PI→PO erasure 정정 후의 Block error rate라 하자. 그 때 result2 $[i]_j$ 는 다음과 같다.

$$result2 [i]_j = ED [i]_j - \sum_{\sum_{k=1}^j i_k + l \geq d} \binom{M-(j-1) \cdot i}{(M-j) \cdot i, (j-2) \cdot i, \dots, i_2} C_{i_1} \cdot DF(j)^{i_1} \cdot C_{i_{j-1}} \cdot DF(j-1)^{i_{j-1}} \cdot \dots \cdot C_{i_2} \cdot DF(2)^{i_2} \cdot \binom{M - \sum_{k=1}^j i_k}{(M - \sum_{k=1}^j i_k)} C_l \cdot DF(0) \cdot DF(1)^l$$

그리고 PI→PO erasure 정정 후의 Block error rate를 Result2라 하면, Result2는 다음과 같다.

$$Result2 = \sum_{i_2, \dots, i_j} result2 [i]_j$$

2.3.2 PI→PO error 정정

이제 세분화된 정정 대상 ED $[i]_j$ 에서 PI→PO erasure 정정으로 정정하지 못한 error pattern들을

PI→PO error 정정으로 정정한 후의 Block error rate를 result2_{temp} $[i]_j$ 라 하고 그것을 구해 보자. 먼저 ef(j)을 EF(j)에 의한 outer-code codeword의 byte error 개수의 평균값이라 하고(j ≥ 2), Sp를 DF(1)이 발생하는 codeword상에서의 평균 byte error rate라 하자.

$$\sum_{k=2}^j k \cdot i_k + l \geq 17 \text{이고 } \sum_{k=2}^j i_k \cdot ef(k) \leq (d-1)/2 \text{ 경우}$$

1개 중에서 $[(d-1)/2 - \sum_{k=2}^j i_k \cdot ef(k)]$ 이하의 symbol이 에러일 확률을 SCP(1)이라 하면, 여기서 ()는 Gauss number이다.

$$SCP(l) = \frac{\binom{(d-1)/2 - \sum_{k=2}^j i_k \cdot ef(k)}{\sum_{i=0}^l C_i}}{(1-Sp)^{l-i} \cdot Sp^i} \text{ 이고}$$

$$result2_{temp} [i]_j = result2 [i]_j \cdot (1-SCP(l))^M$$

위에서 $\sum_{k=2}^j i_k \cdot ef(k) > (d-1)/2$ 경우에 SCP(l) = 0.

PI→PO error 정정 후의 Block error rate를 Result2_{temp}라 하면 Result2_{temp}는 다음과 같다.

$$Result2_{temp} = \sum_{i_2, \dots, i_j} result2_{temp} [i]_j$$

3. 결 론

앞의 에러 정정 능력 평가 방법을 토대로 그림 1.의 dust error distribution하에 그림 3.과 같은 ECC format((154, 148, 7) × (248, 224, 25) RSPC 두 개를 Block interleave)의 에러 정정 능력은 그림 4.와 같다.

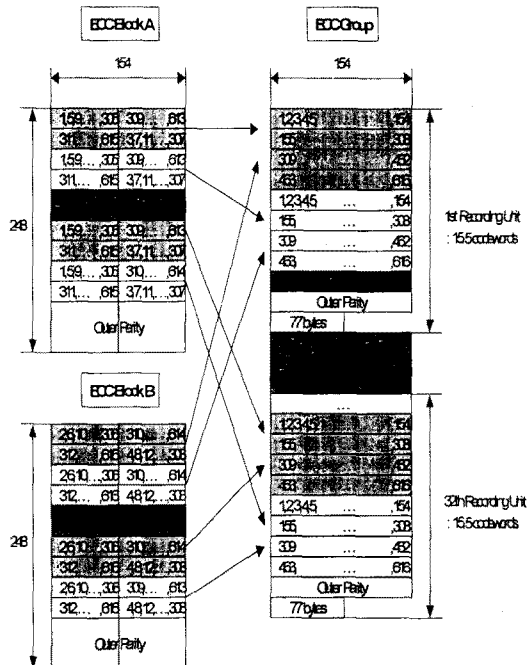


그림 3. 두 Block을 interleave한 ECC format

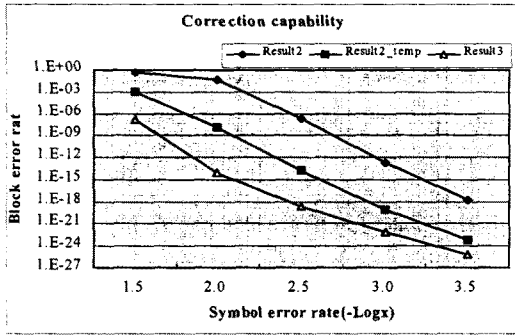


그림 4. 에러 정정 천이도

그림 4.에서 Result2는 PI→PO erasure correction, Result2_temp는 PI→PO error correction, Result3는 PI→PO→PI erasure correction을 했을 시의 시뮬레이션 결과이다.

(참 고 문 헌)

- [1] Kouhei Yamamoto, Masayuki Hattori, Tatsuya Narahara, "Error Modeling and Performance Analysis of Error-Correction Codes for the Digital Video Recording System"
- [2] Kunimaro TANAKA, Koji FUFITA, Tuyoshi SAKAMOTO, Ken NAGASHIMA, Yoshinori NEGISHI and Yutaka KURIHARA, "Performance Evaluation of Error Detection and Correcting Code for Optical Disk(Long Distance Code)"