

Binary Image 의 효율적인 데이터 압축 Algorithm 에
 관한 연구

강 호 갑
 성균관대학교

이 근 영
 전자공학과

An Efficient Data Compression Algorithm For Binary Image

Ho-Gab Kang
 Dept. of Electronics,

Keun-Young Lee
 Sung-Kyun-Kwan Univ.

Abstract

In this paper, an efficient data compression algorithm for binary image is proposed. This algorithm makes use of the fact that boundaries contain all the information about such images.

The compression efficiency is then further increased by efficient coding of Boundary Information Matrix.

The comparison of performance with modified Huffman coding was made by a computer simulation with some images. The results of simulation showed that the proposed algorithm was more efficient than modified Huffman code.

1. 서 론

Binary Image의 coding 방법으로는 크게 두가지로 나눌 수 있다. 첫째로는 오직 scanning line 각각에 대해서만 부호화를 하는 one-dimensional coding 방법으로 Run-length coding과 Huffman coding이 대표적인 부호화 방식이다. CCITT의 study group XIV는 전화회선을 이용한 facsimile의 문서 전송을 위하여 modified Huffman code를 권고하고 있다. 이 방식은 실험 영상의 확률 통계에 근거한 Run-length code로서 terminating code-words(TC)와 make-up codewords(MUC)로 구성된다.

두번째로는 two-dimensional coding 방법으로 두 개 이상의 scanning line에 대해서 pixel들간에 horizontal, vertical correlation을 제거하기 때문에 one-dimensional coding 보다 높은 data 압축율을 얻을 수 있다. 대표적인 방식으로서는 White Block Skipping(WBC) Coding, Chain coding, Modified READ coding 등이 있다.

본 논문에서는 2진 영상의 대부분 부호화 방식이 직접적, 간접적으로 정보를 포함하고 있는 물체의 Boundary를 이용해 부호화하고 있는 점을 이용하여, pixel간에 존재하는 correlation을 제거하였고 부호화 효율을 더욱 개선시키기

위하여 Chain code 개념을 도입, 혼합한 수정된 Run-length and Chain coding(RAC)을 사용하였다.

2. 본 론

2-1. Boundary Information의 추출

기존의 Run-length coding은 직접적 혹은 간접적으로 어떤 영상 물체의 boundary를 이용하여 부호화하는 방식이 대부분이었다. 이는 pixel들 간의 correlation을 제거함으로써 필요한 최소의 정보만을 추출하는 효과를 가져다 주기 때문에 많이 사용되어 왔다. 이 절에서는 2차원적인 배열의 pixel들 간에 존재하는 horizontal, vertical correlation을 제거하는 알고리즘에 관하여 소개하도록 한다.

(1) Horizontal correlation의 제거

(그림 1)과 같이 8 x 8 크기의 배열을 갖는 데이터를 실험 영상의 데이터라고 가정하자.

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

(그림 1)
 원화상의 데이터

원 영상의 horizontal correlation을 제거하기 위하여 수평 방향으로 scan하면서 다음 과정을 수행한다.

- 과정 1- Find l? → (Bi, j)
- 과정 2- If (Bi, j=시점) then (Bi, j=1)
- 과정 3- If (Bi, j=종점) then (Bi, j=1)
- 과정 4- If (Bi, j=고립점) then (Bi, j=-1)
- 과정 5- Otherwise (Bi, j=0)

이와 같은 과정을 거쳐 얻어진 (그림 2)와 같은 데이터를 Boundary Information matrix(BIM)(1)이라고 하자.

0	0	1	0	0	1	0	0
0	-1	0	0	0	0	-1	0
0	-1	0	0	0	1	1	0
0	-1	0	0	-1	0	-1	0
0	-1	0	-1	0	0	-1	0
0	1	1	0	0	0	-1	0
0	-1	0	0	0	0	-1	0
0	0	1	0	0	1	0	0

(그림 2) BIM(1)

0	0	2	0	0	2	0	0
0	3	0	0	0	0	4	0
0	0	0	0	0	2	2	0
0	0	0	0	4	0	3	0
0	3	0	4	0	0	0	0
0	2	2	0	0	0	0	0
0	4	0	0	0	0	3	0
0	0	2	0	0	2	0	0

(그림 3) BIM(3)

(2) Vertical Correlation의 제거

BIM(1)에 존재하는 vertical correlation을 제거하기 위하여 수직 방향으로 scan하면서 다음 과정을 수행한다.

- 과정 1- Find X? → (Bi, j) (X ≠ 0)
- 과정 2- If (X=고립점) and (X=1) then (Bi, j=2)
- 과정 3- If (X=고립점) and (X=-1) then (Bi, j=4)
- 과정 4- If (시점=중점=X) and (X=1) then (Bi, j=1)
- 과정 5- If (시점=중점=X) and (X=-1) then (Bi, j=3)
- 과정 6- Otherwise (Bi, j=0)

위의 과정을 통하여 얻어진 최종 데이터를 BIM(3)이라고 하며 (그림 3)에 표시하였다.

(3) BIM(1)과 BIM(3)의 element가 갖고 있는 의미

- [-1]: horizontal scan시 element[1]의 고립점
- [1]: " " " 시점과 중점
- [2]: vertical scan시 element[1]의 고립점
- [3]: " " " [-1]의 시점과 중점
- [4]: " " " 고립점

(4) BIM(3)에서 BIM(1)으로의 역변환

Vertical scan을 하면서

- 과정 1- If (Bi, j=2) then (Bi, j=1)
- 과정 2- If (Bi, j=4) then (Bi, j=-1)
- 과정 3- If (Bi, j=Bi, j+n=1) then (Bi, j+n=1) for n=0, 1, ..., n
- 과정 4- If (Bi, j=Bi, j+n=3) then (Bi, j+n=-1) for n=0, 1, ..., n

(5) BIM(1)에서 원 영상으로의 역변환

Horizontal scan을 하면서

- 과정 1- If (Bi, j=-1) then (Bi, j=1)
- 과정 2- If (Bi, j=Bi+n, j=1) then (Bi+n, j=1) for n=0, 1, ..., n

2-2. Run-length and Chain (RAC) Coding

BIM(3)를 구성하는 element중 정보를 포함하고 있는 element들의 집합 S는 다음과 같다.

$$S = \{ 1, 2, 3, 4 \}$$

이와 같은 element와 그 element가 위치한 좌표의 전송을 위하여 본 논문에서는 RAC 부호화 방법을 제시하였다.

RAC 부호화 방법은 element 간에 zero-run-length를 이용해서 전송하는 방법으로 일반적인 run-length의 압축

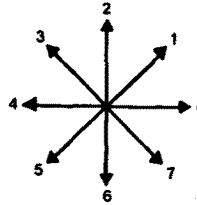
효율을 더욱 높이기 위하여 Chain code 개념을 도입하여 부호화 하는 방법이다.

(1) 수정된 Chain code

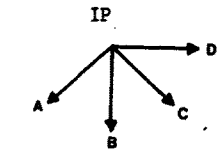
이 앞에서의 Boundary Information 추출에 의하여 화소들 간에 존재하는 horizontal-, vertical correlation을 제거할 수 있었지만 여전히 대각선 방향에 존재하는 correlation은 남아있다고 보아야 할 것이다. 또한 run-length code는 run의 길이가 클 수록, 또는 run의 변화가 적을 수록 효과적이기 때문에 본 논문에서는 Chain code의 개념을 도입하여 사선 방향의 element의 효율적인 처리와 run의 변화를 최소화 시켜 압축율을 높이도록 하였다.

일반적인 Chain code는 (그림 4)와 같이 H.Freeman이 제안한 8방향 Chain code를 사용하고 있다.

이 방법은 동일한 grey level의 영역을 검출하거나, 또는 문자와 같은 2진 영상의 경계를 부호화하는데 많이 사용되고 있다.



(그림 4) H.FREEMAN의 CHAIN CODE



(그림 5) 수정된 CHAIN CODE

(그림 4)와 같은 방법은 grey level과 시작점(IP; Initial Point; x축과 y축의 좌표로서 지정됨), 방향 정보인 chain vector들의 집합으로 이루어지며 Chain code의 수행을 위해서 IP Algorithm과 T Algorithm이 사용된다. IP Algorithm은 특정 광도의 시작점을 찾는 데 사용되며, T Algorithm은 동일한 요소(element) 들을 추적하는데 사용된다. 이때 code 열은 chain의 번호, 시작점의 x, y좌표, chain의 방향 vector를 순서적으로 나열함으로써 부호화를 한다.

본 논문에서는 이들 algorithm의 적용을 위해 수정된 Chain code를 사용하였다. 즉, IP Algorithm은 정보를 네포하고 있는 집합에서의 element 발견과 이들 element들 간의 run-length를 구하는데 사용하였으며, T Algorithm은 Chain vector들을 이용해서 element들을 추적하는데 사용하였다.

- 가정 1- Chain code는 8방향 chain code중 (그림 5)와 같이 5, 6, 7, 0의 네 방향 vector만을 사용하기로 하며 chain vector는 A, B, C, D로 한다.

- 가정 2- IP에서는 A, B, C, D의 4방향 모두에 대해서 T Algorithm을 적용하지만 P1(Chain code로서 연결되는 IP의 첫번째 level의 element)에서부터는 A, B, C, D의 방향중에서 우선 방향에 위치하는 element들만을 추적한다.

- 가정 3- IP에서 어느 한 방향으로의 추적이 끝나고 다음 방향으로의 추적을 시작할 경우에는 추적이 끝난 현재 방향의 Chain code word에 한 방향의 추적이 끝났음을 표시하는 code(one-search stop; OSE- "S")를 연결한다.

- 가정 4- IP에서의 네 방향 추적이 모두 끝난 경우, 즉 더 이상 chain vector가 연결되지 않을 경우에는 IP에서의 chain coding이 끝났음을 알리는 code(All-search end; ASE-"2")를 연결한다.

- 가정 5- chain vector의 우선 순위는 A,B,C,D의 순이다.

- 가정 6- 추적된 element들의 재추적을 피하기 위해 IP는 chain coding의 시점을 나타내는 "5"로 marking하고 Pi는 "0"으로 marking한다. 이렇게 함으로써 현재의 IP에서 다음 IP로의 coding을 할 때 재추적을 피할 수 있게 된다.

- 가정 7- 수정된 chain code를 효과적으로 사용하기 위해서 최소한 2개 이상의 Pi element들이 IP의 element와 연결되어 있어야 한다. 그렇지 않을 경우에는 chain code의 방향을 나타내는 chain vector와 ASE를 표시하는 code들의 인여 bit때문에 run-length coding을 취하는 것에 비해 비효율적이기 때문이다.

(table 1)은 어느 한 IP에 대하여 수정된 chain coding을 적용시켰을 때 얻어지는 code word열이다.

IP의 element	방향vector	Pi의 element	방향vector	..
Pi의 element	OSE	방향vector	Pi의 element	..
Pi의 element	ASE			

(TABLE 1)CHAIN CODE의 열

BIM(3)에 대해서 수정된 chain code를 적용했을 때 나타나는 BIM(4)를 (그림 6)에 나타내었다.

0	0	2	0	5	0	0
0	3	0	0	0	0	0
0	0	0	0	0	2	0
0	0	0	0	0	3	0
0	3	0	0	0	0	0
0	2	0	0	0	0	0
0	0	0	0	0	3	0
0	0	0	0	2	0	0

(그림 6)BIM(4)

(2) 수정된 Run-length coding

이 절에서는 BIM(4)의 효과적인 run-length coding을 위하여 수정된 run-length coding을 사용하였다. 일반적인 run-length coding 방법은 run의 길이가 길수록 효과적이는데 BIM(4)의 구조는 element가 드문 드문 있기 때문에 비효율적이다. 또한 각 line의 run-length coding이 끝났을 경우, 각 line의 coding이 끝났음을 알리는 EOL-code word를 첨부해야만 한다. 이러한 EOL-codeword를 첨부하는 이유는 run-length coding의 특성상 transmission error에 약하기 때문이다.

본 논문에서는 이러한 점을 개선시키기 위해 다음과 같이 BIM(4)의 matrix 배열을 축소시키는 algorithm을 적용시켰다.

- 과정 1- BIM(4)에 대하여 horizontal scan을 하면서 각 line에 적어도 하나 이상의 element가 존재할 경우에는 해당 line을 1로 표시한다. 만일 line의 끝까지 element가 존재하지 않을 경우에는 "0"으로 표시하고 해당 line을 제거한다. 이와 같은 과정을 horizontal scan하면서 각각의 line에 대하여 수행한다. 이때 각 line에 대한 code "0"

과 "1"의 열을 HNEL이라 하고 (그림 7)과 같이 NEL에 저장한다. 이와같은 과정을 거쳐 얻어진 데이터를 BIM(5)라 하자.

- 과정 2- BIM(5)에 대하여 vertical scan하면서 (과정 1)과 같은 과정을 거치도록 한다.

이때 각 line에 대한 code "0"과 "1"의 VNEL열을 NEL에 저장하고 축소된 matrix를 (그림 8)에 나타내었다.

NEL	1	2	3	4	5	6	7	8
HNEL	1	1	1	1	1	1	1	1
VNEL	0	1	1	0	0	1	1	0

(그림 7)NON-EMPTY LINE

0	2	5	0
3	0	0	0
0	0	0	2
0	0	0	3
3	0	0	0
2	0	0	0
0	0	0	3
0	0	2	0

(그림 8)BOUNDARY INFORMATION MATRIX(6)

이와 같은 과정을 거쳐 얻어진 BIM(6)은 BIM(4)에 비하여 상당한 배열의 삭제가 이루어졌음을 알 수 있다.

이것은 run-length coding을 사용했을 때 EOL의 code가 축소되는 효과를 가져다 준다.

BIM(6)의 element에 대하여 수정된 run-length coding을 수행한다. 이때 run-length를 표시하는 codeword는 (table 2)와 같다.

RUN-LENGTH	CODEWORD	BIT
0 - 3	1**	3
4 - 15	0**1**	6
16 - 63	0**0**1**	9
64 - 255	0**0**0**1**	12
256- 1023	0**0**0**0**1**	15

(TABLE 2)RUN-LENGTH CODEWORD

Run-length codeword뒤에는 BIM(6)의 element(1,2,3, 4,5)가 붙게되며,element "5"는 chain coding시의 IP이기 때문에 "5"의 code 뒤에는 chain code의 열이 나열되게 된다. 이때 BIM(6)와 chain code열의 element의 code는 Huffman coding을 사용해 최적의 bit 할당을 하며 chain vector A,B,C,D는 2bit로 할당하도록 한다.

이상에서 처리된 RAC의 code열은 (table 3)과 같은 순서열로 이루어 진다.

EOB	Run-length codeword	element 또는
chain code prefix	(chain code)	Run-length
codeword	element 또는 chain code prefix

(table 3) 전송되는 데이터 순서

3. 실험 결과 및 고찰

본 논문에서의 simulation을 위하여 (그림 9)와 같은 128 x 128 크기의 실험 영상을 사용하였다. 본 논문에서 제시한 algorithm의 data 압축 효과를 판정하기 위하여 CCITT에서 공표한 Modified Huffman coding을 사용하였으며 4개의 실험 영상에 대한 bpp 및 압축율을 (표 2)에 기록하였다. 또한 배열의 크기를 축소한 BIM(6)의 X축, Y축 크기를 (표 1)에 기록하였다.

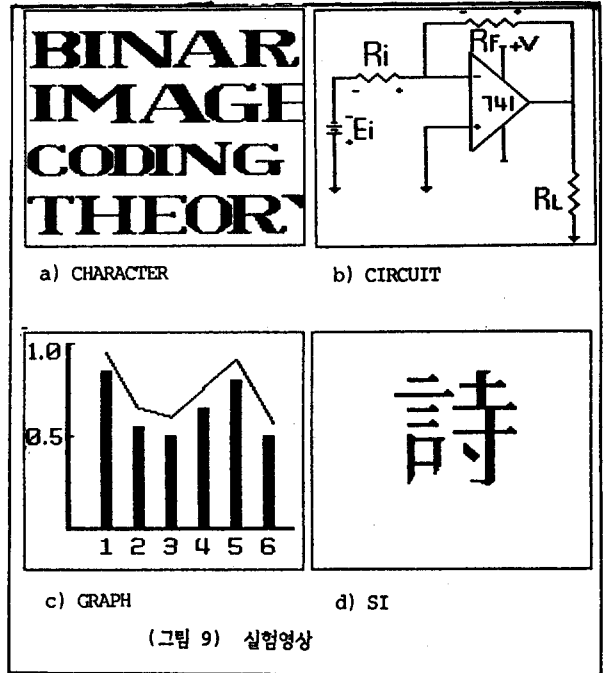
본 논문이 제시한 algorithm의 simulation 결과는 전체적으로 Modified Huffman coding에 비해서 상당한 성능 개선을 얻을 수 있었으며, 특히 사선의 경계를 갖거나 또는 직교적인 경계를 갖는 image에 대해서는 더욱 더 효과적이었음을 알 수 있었다.

본 논문에서 제시한 Algorithm은 Boundary Information 추출과 RAC coding을 통하여 원영상의 크기를 축소하였으며, 이로 인한 압축효과뿐만 아니라 Run-length coding시 필요한 EOL - code를 줄일수 있는 잇점을 지니고 있다. 본 논문의 (표2)에 나타난 BPP(Bit Per Pixel)의 측정에는 EOL-code를 고려하지 않았다.

앞으로 통계학적인 고찰이 본 알고리즘에 적용된다면 더 좋은 결과가 기대되어진다.

4. 참고 서적

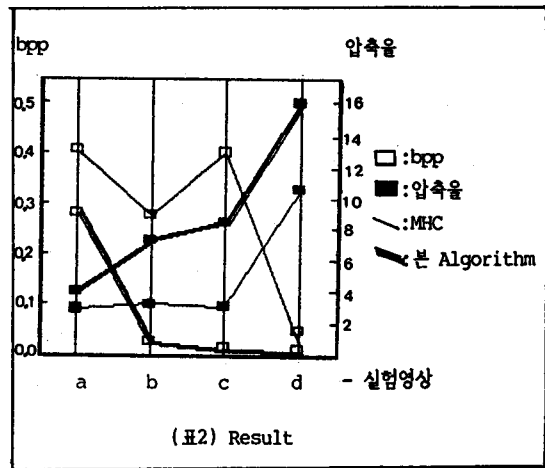
1. Iickho Song, "A New Noiseless Coding Techque for Binary Images", IEEE Tras. Acoustics, Speech and Signal Processing, Vol. ASSP-34, No.4, Aug., 1986.
2. Yasuhiko Yasuda, "Overview of Digital Facimile Coding Techniques in Japan", IEEE Proc. Vol. 68, No.7, July., 1980.
3. Roy Hunter and A. Harry Robinson, "International Digital Facsimile coding Standards", IEEE Proc., Vol. 68, No. 7, July, 1980.
4. Thomas S. Huang, "Coding of Two-Tone Images", IEEE Tras. Comm., Vol. COM-25, No.11, Nov., 1977.
5. Rafael C. Gonzalez & Paul Wintz, "Digital Image Processing", Addison-Wesley, 1977.
6. Richard W. Hamming, "Coding and Information Theory", Prentice-Hall, 1986.



(그림 9) 실험영상

	a	b	c	d
HNEL	64	58	36	37
VNEL	116	68	49	23

(표1) Horizontal-, vertical non-empty line 갯수



(표2) Result