혼합 부호화에 의한 압축률 개선에 관한 고찰II

차인숙, 박지환
*부경대학교 전산정보학과
**부경대학교 컴퓨터멀티미디어공학전공

A Study on Improving Compression Ratio Using Hybrid Coding

In-Sook Cha, Ji-Hwan Park
*Dept. of Computer & Information Science, PuKyoung Nat'l University
**Div. of Computer & Multimedia Engineering, PuKyoung Nat'l University

요 약

국제 전신전화자문위원회인 ITU-T에서는 일반 공중전화망을 이용하는 데이터 통신을 위하여 표준권고안 중 V시리즈의 하나인 V.42bis라는 표준을 제정하여 권고하고 있다. 이 표준은 자동재전송 방식(ARQ)으로 오류를 제어하고 있는 V.42 표준에 새로이 LZW압축기법을 추가한 것으로 모뎀 내에 오류제어와 데이터 압축방식을 함께 채용함으로써 데이터전송에 있어서 신뢰성과 효율성을 제공하도록 한 것이다. 이 논문에서는 V.42bis방식의 압축 효율성에 대하여 고찰하고, 적응형 산술 부호(Arithmetic Code)와 1중 마르코프 산술부호로 각각 혼합 부호화하여 압축률을 향상시키는 기법에 대하여 기술한다.

1. 서론

데이터 압축기술의 발달은 파일 저장이나 분산화일 시스템, 데이터 통신, 음성 및 화상통신, 컴퓨터 네트워크 등의 분야에서 중요한 역할을 담당하고 있다. 이 중에서도 빠른 속도로 데이터를 전송하기 위해서는 데이터 압축기법이 필수적이다.

데이터 압축[1]이란 코드화(Coding)의 일종이다 코드화란 임의의 데이터를 특정한 목적에 맞게 변경하는 엔코딩(Encoding)과 이 변경된 데이터를 다시 원래대로 복원하는 디코딩(Decoding)을 모두 말하는 것이며, 이러한 데이터 압축을 수행하는 실제의 소프트웨어는 최근 눈부시게 보급되어 왔다. UNIX시스템에서 최초로 출현한 압축소프트웨어는 compact이다. 이것은 적응형 산술 부호(Arithmetic Code)에 기초한 압축 소프트웨어로서 당시로는 충분히 실용적인 압축률을 달성할 수 있는데 반하여 실행속도가 느린 결점이 있었다. 이것의 해결책으로 UNIX시스템에서 출현한 것이 Compress이다.

Compress는 Ziv-Lempel-Welch(LZW) 부호[2]에 기초한 소프트웨어로서 compact보다 뛰어난 압축률과 고속성을 가진 것으로 현재 널리 이용되고 있다. 개인용 컴퓨터를 위하여 MS-DOS사에서 최초로 보급된 압축 소프트웨어가 ARC(PKARC)이다. ARC는 LZW 부호에 기초한 파일 압축기능 이외에도 몇 개의 파일을 하나의 파일로 묶는 아카이버(archiver) 기능을 가진 소프트웨어로서 MS-DOS 시스템에서는 한 때 사실상의 표준이 되기도 하였다.

데이터 압축을 할 때 어느 알고리즘이 가장 좋은가에 대한 해답은 상황에 따라 선택적이라는 것이다. 예를 들어, 컴퓨터에서 CPU나 메모리의 용량에 한계가 있는 경우, 사용 가능한 알고리즘은 간단하고 고속이어야 하며 대폭적인 압축을 바랄 수는 없다. 그러나, 어느 정도의 처리시간이나 메모리 사용이 허용된다면 보다 뛰어난 압축이 가능하게 되며, 이 때 LZ(Ziv-Lempel)부호 등이 그 조건을 만족하는 압축 알고리즘이 된다.

본 논문에서는 CCITT가 권고한 데이터 통신을 위한 데이터 압축 방식인 일반 가입자 전화선에 대하여 데이터의 비동기식 전송을 위한 국제 표준 권고안인 ITU-T V.42bis의 방식[3,4]을 고찰하고, 압축률을 향상시키기 위하여 비압축 부분에 대하여 2단계 부호화하는 혼합 부호화 방법을 제안한다. 논문의 내용은 다음과 같다. 먼저, 2장에서는 V.42bis 압축 알고리즘의 기본이 되는 LZW부호화 방식과 2단계 압축에 사용되는 적응형 산술부호화(Arithmetic Code)[8][9] 방식을 소개하고, 3장에서는 압축률을 향상시키기 위한 혼합 부호화 방식을 제안, 4장에서는 제안 방식과 기존 방식 사이의 성능을 비교하고, 5장에서 결론을 맺는다.

2. 무손실 압축

데이터 압축 알고리즘은 정적 부호화(static coding)와 동적 부호화(dynamic coding)로 분류할 수 있다.

정적 부호화는 우선 정보원의 모델을 작성한 후, 얻어진 모델을 기초로 입력 기호열을 부호화해 가는 방식으로 Huffman 부호[5]와 Huffman부호의 문제점을 해결하여 부호화와 복호화가 산술 연산(사칙 연산)에 의해 실행되는 적응형 산술 부호(Arithmetic Code)가 있다.[8][9].

동적 부호화는 입력되는 기호열의 통계적 성질에 관계없이 어떠한 정보원으로부터 생성된 기호열에 대해서도 그 기호열이 길어짐에 따라 달성할 수 있는 압축률이 최적이 되는 범용 데이터 부호화이다. 대표적인 동적 부호화 방식으로는 J.Ziv와 A.Lempel이 개발한 Ziv-Lempel부호가 있다[7].

2.1 LZW(Lempel-Ziv-Welch)부호

ITU-T의 V.42bis에 채용되어 있는 LZW 압축기법은 1978년 이스라엘 Lempel과 Ziv가 처음으로 제안하고, 1985년 현재 유니시스사의 전신인 스페리사의 Welch가 수정 구현한 압축기법이다. 이 압축기법에서 쓰인 LZW부호는 유니버샬 부호이며, 부호화할 입력 문자열(string)을 사전에 등록되어 있는 부분 문자열과 일치하는 단어를 찾아 일치하는 최대 길이로 분해(parsing)한 후, 분해된 문자열에 해당하는 인덱스를 부호화하여 전송하는 방법이다. 이 때 사전에 없는 단어들은 새롭게 사전에 추가되면서 사전이 적응적[6,7]으로 갱신된다. 또한 입력 데이터의 길이를 가변으로 하고 출력부호의 길이를 고정시킨 기법으로서 데이터 압축률이 높으며, 내부 연산량이 작기 때문에 압축수행속도 측면에서는 현재까지 가장 빠른 것으로 평가되고 있으므로 ITU-T의 표준권고를 기점으로 정보통신의 요소기술로 널리 이용되고 있다.

■ LZW 부호의 구성 알고리즘

① 한 기호로 이루어진 단어를 모두 사전에 먼저 등록하여 초기화한다.
② 입력 기호열에 대하여 최장일치 부분열을 등록된 사전에서 찾는다.
③ 최장일치 부분열의 참조번호를 사전의 크기에 대응하는 비트수로 부호화한다.
④ 새로운 절점 번호를 만들어 입력 기호열과 일치하지 않는 계열을 사전에 등록한다.
⑤ ②번부터 입력계열의 끝까지 반복 수행한다.

2.2 산술부호(Arithmetic Code)

메사추세츠 공과대학의 P. Elias에 의해 제안된 산술 부호(Arithmetic Code)는 기호열 길이에 비례하는 정도의 계산량으로 기호열 전체를 하나의 부호어로 하는 실용적인 부호화법이다. 여기서는 기호를 읽어들일때마다 각 기호의 발생 확률을 변경하면서 부호화를 수행하는 적응형 산술부호(Arithmetic Code)와 심볼 간의 상관관계를 조건 확률에 의하여를 적용하였는 1중 마르코드 산술부호화[1]를 적용하였다.

■ 산술부호의 구성 알고리즘

① 각 기호의 출현 빈도를 1로 한 빈도표를 만들고 문맥을 나타내는 변수의 초기치를 정하여 초기화한다.
② 기호열을 읽어 들여 빈도표를 이용한 산술부호에 의해 기호열을 부호화하고 기호열이 끝이면 부호화를 종료한다.
③ ②번부터 입력계열의 끝까지 반복 수행한다.

3. 제안 방식

V.42bis에 채용된 압축 방식은 LZW 부호화를 변형한 압축방법이다. LZW방식으로 압축하면서 주기적으로 압축률을 측정하여 문턱값과 비교한다. 압축률이 문턱값보다 클 경우는 압축형식으로, 작을 경우는 비압축형식으로 보내어진다. 제안 방식은 V.42bis에서 채용된 압축방법에서의 비압축형식으로 보내어지는 부분에 이미 소개한 적응형 산술 부호(Arithmetic Code)화 방식과 1중 마르코프 산술 부호를 적용하여 기존방법보다 압축률을 향상시키는 것을 목적으로 한다.

3.1 V.42bis의 압축방식

■ 부호화 알고리즘[10]

① 입력 데이터로 가능한 모든 문자를 사전에 등록시켜 초기화한다.

② 읽어들인 문자열과 사전에 등록된 문자열을 비교하여 일치되는 문자열의 위치 값을 찾는다.

③ 주기적으로 검사하여 부분 압축률이 문턱값보다 작으면 비압축형식(읽어들인 문자의 이진값)으로, 크면 압축형식(사전에서 찾은 문자열의 위치값)으로 부호화한다.

④ 부호화 되지 않은 입력 데이터를 사전에 등록시킨 후, ②번 과정부터 반복 수행한다.

3.2 제안 방식

■ 부호화 알고리즘

① 입력 데이터로 가능한 모든 문자를 사전에 등록시켜 초기화한다.

② 읽어들인 문자열과 사전에 등록된 문자열을 비교하여 일치되는 문자열의 위치 값을 찾는다.

③ 주기적으로 검사하여 부분 압축률이 문턱값보다 작으면 비압축형식(읽어들인 문자의 이진값)을 저장하고, 크면 압축형식(사전에서 찾은 문자열의 위치값)을 저장한다.

④ 부호화 되지 않은 입력 데이터를 사전에 등록시킨 후, ②번 과정부터 반복 수행한다.

⑤ 비압축형식으로 이루어진 파일을 읽어들여 적응형 산술 부호 압축방식으로 압축을 수행한다.

⑥ 압축방식과 비압축 방식을 혼합하여 전송파일을 저장 후 전송한다.

4. 모의실험 결과 및 분석

기존의 압축방식과 제안 방식의 압축률을 비교하기 위하여 표1의 4가지 데이터를 사용하였다.

사전에 등록 가능한 기호열의 수 M을 4,096~204,800까지 변화시키면서 압축률의 수렴속도 변화를 관찰한 결과, 입력열의 최초 부분에 대한 속도는 M에 관계없이 항상 일정하며, 그 수렴값이 M에 의존하였다. M이 늘어 날수록 압축률은 작아지나, 409,600까지 늘려도 압축률에는 거의 변화가 없었으므로 M=4,096으로 설정하였고, 어떤 형식(비압축/압축)으로 전송할 것인가를 판단하는 문턱값도 0.3~0.9로 설정하였으며, 입력 기호열의 수는 12~20개를 실험하여 가장 압축률이 좋은 0.9의 문턱값과 20개의 입력 기호열을 설정하였다. 아래의 표1은 문턱값과 입력 기호열의 개수에 변화를 주면서 V.42bis의 방식으로 실험한 결과이고 이 결과를 그림1로 표현하였다.

표1. V.42bis방식의 압축성능

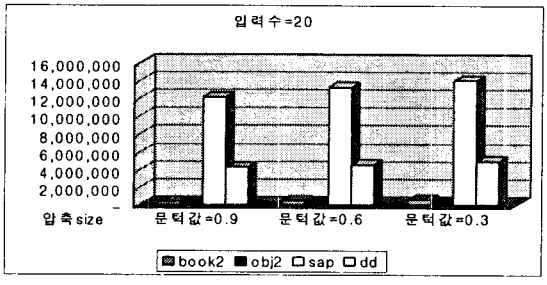
[단위 : byte]

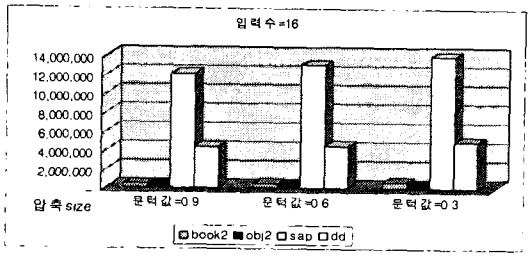
File	Size	입력수=20 문턱값=0.9	입력수=20 문턱값=0.6	입력수=20 문턱값=0.3
book2	626,490	400,215	484,405	630,362
obj2	246,814	241,785	245,128	247,886
sap.zip	14,009,709	12,144,907	13,238,847	14,063,866
dd.bmp	5,081,218	4,317,091	453,2174	495,3145

File	Size	입력수=16 문턱값=0.9	입력수=16 문턱값=0.6	입력수=16 문턱값=0.3
book2	626,490	413,231	466,060	630,430
obj2	246,814	242,478	245,100	247,981
sap.zip	14,009,709	12,150,377	13,110,778	13,979,042
dd.bmp	5,081,218	4,355,305	4,489,881	4,954,330

File	Size	입력수=12 문턱값=0.9	입력수=12 문턱값=06	입력수=12 문턱값=0.3
book2	626,490	429,053	531,548	632,290
obj2	246,814	243,211	246,558	248,783
sap.zip	14,009,709	12,127,728	13,304,590	14,083,900
dd.bmp	5,081,218	4,403,501	4,611,708	4,974,966

그림1. V.42bis방식의 기호의 수와 문턱값의 비교





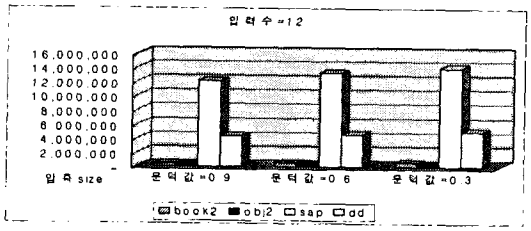


표1과 그림1의 결과에서 볼 때, 가장 좋은 압축률은 문턱값0.9와 입력기호열의 수 20이었다.

따라서, 가장 좋은 압축률을 달성할 수 있는 이런 값에 대하여 제안 방식을 적용하기 위해 비압축부분과 압축부분의 비율을 표2에 나타내었고, V.42bis방식과 비교하기 위하여 표3에 그 결과를 아래와 같이 정의되는 압축률로 나타내었다.

$$\text{압축률} = \frac{\text{압축된 파일의크기}}{\text{원 파일의크기}} * 100\ [\%]$$

표2. V.42bis방식에서의 비압축부분과 압축부분 비율표

File	입력수=20,문턱값=0.9	
	압축비율	비압축비율
book2	90.38%	9.62%
obj2	7.00%	93%
sap.zip	44.23%	55.77%
dd.bmp	35.08%	64.92%

표3. V.42bis방식과 제안방식의 압축률 비교표

File	V.42bis 방식	제안방식		
		허프만 부호적용	적응형 산술부호	1중 마르코트
book2	63.9%	61.0%	60.9%	43.2%
obj2	98.0%	78.3%	78.1%	77.6%
sap.zip	86.7%	79.5%	79.3%	78.4%
dd.bmp	85.0%	82.2%	81.9%	75.3%

표3의 결과 제안 방식이 V.42bis방식보다 높은 압축률을 달성할 수 있음을 알 수 있다.

5. 결론

본 논문에서는 기존의 V.42bis방식으로 압축하였을 때 발생하는 비압축 부분을 적응형 산술부호(Arithmetic Code)와 1중 마르코프 산술부호로 부호화하여 압축률을 개선하는 혼합 부호화에 대하여 고찰하였다. 그 결과, V.42bis보다 높은 압축률을 달성할 수 있었고, 기존에 적용한 허프만 부호[11]와 적응형 산술부호보다는 1중마르코프 산술부호를 적용하였을 때 더 압축률이 향상되었음을 알수 있었다.. 향후 수행속도를 개선하기 위한 고속 알고리즘의 개발이 요구된다.

[참고문헌]

[1] 植松友彦 著 朴志煥 譯, "데이터 압축 알고리즘 입문", 성안당 1995

[2] T.A. Welch, *"A Technique for High Performance Data Compression"*, IEEE Comput., vol.17, pp.8-19, Jun. 1984

[3] W.J. Betda, *"Data Communication "*, Prentice Hall, 1996

[4] T.C. Bell, J.h. Cleary, I.H. Witten, *"Text Compression"*, Prentice Hall, 1990

[5] D.A. Huffman, *"A Method for the Construction of Minimum Redundancy Codes"*, Proc. IRE, vol.40, no.9, pp.1098-1101, Sep. 1952

[6] J. A. Storer, *"Image and Text Compression"*, Kluwer Academic Publishers, 1992

[7] J. Ziv, A. Lempel, *"Compression of Individual Sequence via Variable-rate Coding"*, IEEE Trans. on Information Theory, vol. IT-24, pp.530-536, 1978

[8] I.H. Witten, R.M.Neal, J.G.Cleary, *"Arithmetic Coding for Data Compression"*, Communication of the ACM, vol.30, no.6, pp.520-539, Jun. 1987

[9] C.B. Jones, *"An Efficient Coding System for Long Source Sequences"*, IEEE Trans. on Inform. Theory, vol.IT-27 , no.3, pp.280-291 , May.1981

[10] 조성렬, "고속 일반 데이터 전송을 위한 데이터 압축 방식의 연구," 석사 논문, 서울대학교, 1997.

[11] 차인숙, "혼합 부호화에 의한 압축률 개선에 관한 연구" 한국멀티미디어 춘계학술 논문집, 2000.