혼합 부호화에 의한 압축률 개선에 관한 연구

차인숙, 박지환
*부경대학교 전산정보학과
** 부경대학교 컴퓨터 멀티미디어공학전공

A Study on Compression Ratio Improving Using Hybrid Coding

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

요 약

최근 정보통신에서 데이터압축의 이용현황을 살펴볼 때, 국제 전신전화자문위원회인 ITU-T에서는 일반 공중전화망을 이용하는 데이터통신을 위하여 표준권고안 중 V시리즈의 하나인 V.42bis라는 표준을 제정, 권하고 있다. 이 표준은 자동재전송 방식으로 에러제어를 하고 있는 V.42라는 표준에 새로이 LZW 압축기법을 추가한 것인데, 모뎀 내에 에러제어와 데이터 압축방식을 모두 채용함으로써 데이터전송에 있어서 정확성과 경제성을 제공하도록 한 것이다.

이 논문에서는 V.42bis방식에 Huffman 부호화방식를 혼합하여 압축률을 향상시키는 기법을 제안한다.

1. 서론

최근들어 멀티미디어라는 용어가 유행하고 있으며, 실제로 멀티미디어 통신처리 기술은 빠른 속도로 발전해 나가고 있다. 앞으로의 정보화 사회를 주도하게 될 멀티미디어 통신기술은 많은 사람들에게 대량의 정보를 제공하는 주요 수단이 될 것이다. 그러므로, 대량의 정보를 빠른 속도로 교환하기 위해서, 고성능의 컴퓨터는 물론 통신망의 주어진 전송로를 최대한 활용하면서 고속으로 데이터를 전송하기 위해서는 데이터 압축기법이 필수적이다.

데이터 압축이란 컴퓨터에서 취급하는 각종 파일의 크기를 물리적으로 축소시키는 것으로, 압축을 이용하여 파일의 크기를 줄이면 하드디스크의 공간을 넓히거나 파일의 백업에 필요한 플로피 디스크의 매수를 줄일 수 있으며, 통신으로 파일을 송수신할 때 통신시간과 통신비용을 절약할 수 있다. 실제로 데이터 압축 기술은 파일 저장이나 분산화일 시스템, 데이터 통신, 음성 및 화상통신, 컴퓨터 네트워크 등의 분야에서 중요한 역할을 담당하고 있다. 이러한 데이터 압축을 수행하는 실제의 소프트웨어는 최근 눈부시게 보급되어 왔다. UNIX의 세계에서 최초로 출현한 압축소프트웨어는 compect이다. compect는 적응형 huffman부호에 기초한 압축 소프트웨어로서, 당시로는 충분히 실용적인 압축률을 달성할 수 있는데 반하여 실행속도가 느린 결점이 있었다. 이것의 해결책으로 UNIX세계에서 출현한 것이 compress이다.

Compress는 Ziv-Lempel-Welch(LZW) 부호에 기초한 소프트웨어로서 compect보다 뛰어나 압축률과 고속성을 가진 것으로 현재 널리 이용되고 있다. 개인용 컴퓨터에 대하여, MS-DOS사에서 최초로 보급된 압축 소프트웨어가 ARC(PKARC)이다. ARC는 LZW부호에 기초한 파일 압축기능 이외에도 몇 개의 파일을 하나의 파일로 묶는 아카이버(archiver) 기능을 가진 소프

트웨어로서 MS-DOS 세계에서는 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 압축알고리즘이 된다.

본 논문에서는 현재 LZW부호화를 기반으로 하는 일반 가입자 전화선에 대하여 데이터의 비동기식 전송을 위한 국제 표준 권고안인 CCITT V.42bis의 알고리즘을 개선한 혼합부호화 방법을 제안한다. 논문의 내용은 다음과 같다. 먼저, V.42bis에서 사용한 LZW 부호화 방식과 제안 알고리즘에 적용된 Huffman부호화 방식을 2장에서 소개하고, 3장에서는 압축률 향상을 위한 방법을 제안하며, 4장에서는 제안 방법과 기존 방법 사이의 성능을 비교하고, 5장에서 결론을 맺는다.

2. LZW부호와 허프만부호

데이터 압축 알고리즘은 정적부호화(static coding)와 동적부호화(dynamic coding)로 분류할 수 있다.

정적부호화는 우선 정보원의 모델을 작성한 후, 얻어진 모델을 기초로 입력 기호열을 부호화해 가는 방식으로, D. A. Huffman이 개발한 Huffman 부호가 가장 널리 쓰이고 있다.

동적부호화는 입력되는 기호열의 통계적 성질에 관계없이, 어떠한 정보원으로부터 생성된 기호열에 대해서도 그 기호열이 길어짐에 따라 달성할 수 있는 압축율이 최대인 만능 데이터 부호화이다. 대표적인 동적부호화 방식으로는 J.Ziv와 A.Lempel이 개발한 Ziv-Lempel부호가 있다.

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, 한 기호로 이루어진 단어를 모두 사전에 등록해 둠으로써 부호어 중에 포함되는 기호를 제거하며, 항상 포인터만으로 끝나 압축이 안되고 전송되는 부분이 없는 특징이 있다. 또한 입력 데이터의 길이를 가변으로 하고 출력부호의 길이를 고정시킨 기법으로서 데이터 압축률이 높으며, 내부 연산량이 작기 때문이 압축수행속도 측면에서는 현재까지 가장 빠른 것으로 평가되고 있으므로 ITU-T의 표준권고를 기점으로 정보통신에 널리 이용될 것으로 보인다.

■ LZW 부호의 구성 알고리즘

① 초기 설정에서, 한 기호로 이루어진 단어를 모두 사전에 먼저 등록한다.
② 입력 기호열에 대하여 최장일치계열을 등록된 사전에서 찾는다.
③ 최장일치계열의 참조번호를 사전의 크기에 대응하는 비트수로 부호화한다.
④ 새로운 절점 번호를 만들어 입력 기호열과 일치하지 않는 계열을 사전에 등록한다.
⑤ ②번부터 기호열의 끝까지 반복 수행한다.

2.2 허프만 부호(Huffman Code)

허프만 압축기법은 1952년 허프만에 의해 제안된 압축 방식으로 오늘날에도 널리 이용되고 있다.

이 기법은 정보원 데이터 내의 각 문자에 대한 발생빈도를 조사해 자주 나타나는 문자에는 보다 짧은 부호어를, 그리고 잘 나타나지 않는 문자에는 긴 부호어를 할당함으로써, 압축 후의 부호어 길이를 원래 정보원 길이보다 더욱 축소시킬 수 있도록 통계적 특성을 이용한 압축기법이다.

다음은 허프만 부호의 구성 알고리즘이다.

■ 허프만 부호의 구성 알고리즘

① 각 기호에 대응하는 잎을 만들고, 발생확률을 기록한다.
② 발생확률이 가장 낮은 2개의 잎을 1개의 새로운 잎으로 결합하여 한쪽에는 "0"을 다른 한쪽에는 "1"을 할당하고 두, 발생확률의 합을 새로운 잎에 기록한다.
③ 만들어진 잎의 확률값과 나머지 잎의 확률값을 크기순으로 정열한다.
④ 잎이 1개가 될 때까지, 즉 발생확률의 합이 1이 될 때까지 ②번을 반복한다.
⑤ 뿌리에서 각 기호의 잎으로 연결되는 가지에 붙여지는 "0"과 "1"의 계열이 그 기호의 부호어가 된다.

3. 제안 알고리즘

V.42bis의 압축 방법에 쓰이는 LZW 부호화를 변형한 압축방법이다. LZW방식으로 압축하면서 주기적으로 압축률을 측정하여 문턱값과 비교한다. 압축률이 문턱값보다 작을 경우 비압축형식으로 보내어진다. 제안 알고리즘은 비압축형식으로 보내어지는 부분에 이미 소개한 Huffman부호화 방식을 적용하여 기존방법보다 압축률을 향상시키는 것을 목적으로 한다.

3.1 V.42bis의 압축방식

■ 부호화 알고리즘

① 입력 데이터의 가능한 문자 모두를 사전에 등록시켜 사전을 초기화한다.

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

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

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

3.2 제안 알고리즘

■ 부호화 알고리즘

① 입력 데이터의 가능한 문자 모두를 사전에 등록시켜 사전을 초기화한다.

② 읽어들인 문자열과 사전에 등록된 문자열을 비교하여 사전에 등록된 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 FILE을 읽어들여 허프만 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 입력열의 수렴속도 변화를 관찰한 결과 입력열의 최초 부분에 대한 속도는 M에 관계없이 항상 일정하며, 그 수렴값이 M에 의존하였다. M이 늘수록 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 결과이다.

단위 :byte

File	FILESIZE	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	FILESIZE	입력수=16 문턱값=0.9	입력수=16 문턱값=0.6	입력수=16 문턱값=0.3
book2	626,490	413231	466,060	630,430
obj2	246,814	242478	245,100	247,981
sap.zip	14,009,709	12150377	13,110,778	13,979,042
dd.bmp	5,081,218	4,355,305	4,489,881	4,954,330

File	FILESIZE	입력수=20 문턱값=0.3	입력수=20 문턱값=0.3	입력수=20 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

표1. V.42bis방식의 입력계열의 수와 문턱값의 비교

표1의 결과에서 볼 때, 가장 좋은 압축율은 문턱값 0.9와 입력기호열의 수 20이었다.

따라서, 가장 좋은 압축율을 달성할 수 있는 이런 값에 대하여 제안 방식을 적용하였고, V.42bis방식과 비교를 위하여 표2에 그 결과를 제시하였다.

단위 :byte

File	FILESIZE	V.42bis	제안알고리즘
book2	626,490	400,215	382,407
obj2	246,814	241,785	193,169
sap.zip	14,009,709	12,144,907	11,136,376
dd.bmp	5,081,218	4,317,091	4,174,438

표2. V.42bis방식과 제안알고리즘 비교표

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

5. 결론

본 논문에서는 기존의 V.42bis방식에 Huffman부호화 과정을 혼용하여 압축하는 기법을 제안하였다. 그 결과는 V.42bis보다 높은 압축률을 제공한다.

향후 수행속도를 개선하기 위한 고속 알고리즘의 개발이 요구된다.

[참고문헌]

[1] W. J. Betda, *Data Communication,*} Prentice Hall, 1996

[2] Motorola Information System Group, "A white paper on synchronous data compression over wide area networks, *"Motorola, Inc,* May8 1996.

[3] J. E. Mcnamara, *Technical Aspects of Data Communication,* Digital Press, 1998.

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

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

[6] J. Ziv, A. Lempel, "Compression of individual sequence via variable-rate coding," *IEEE Trans. on Information Theory, vol.* IT-24, pp.530-536, 1978

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

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