

멀티미디어 압축 기술

박지환*

본고에서는 멀티미디어 응용을 위한 핵심 기술인 데이터 압축 기법에 관하여 기술한다. 먼저, PC통신과 기억장치의 저장공간을 줄여 효과적으로 이용할 수 있는 텍스트 압축^[1]에 관하여 알아보고, 아날로그 신호를 대상으로 하는 음성과 화상 및 동화상 압축을 위한 기본개념을 중심으로 기술한다. 나아가, 차세대 영상압축 기법으로 등장하게 될 프랙탈 압축에 대해서도 기술한다.

1. 시작하며

UNIX에 compress라는 압축 명령이 있다. 이 명령은 예를 들면

```
% compress myfile
```

을 수행하면 myfile이 압축되어 myfile.Z라는 새로운 파일이 만들어진다. 줄어드는 정도는 파일에 따라 다르지만 대략 절반 이하로 줄어든다. 이것을 다시 복원하려면

```
% uncompress myfile.Z
```

를 실행하면 된다. 이와는 별도로 UNIX에는 tar라는 아카이버(archiver)가 있다. 아카이버란 여러 개의 파일을 하나의 파일로 묶어 주는 소프트웨어이다. 아카이버에 의해 생성된 아카이브는 단지 파일을 결합하는 기능만이 아니라 각 파일의 이름, 갱신일시 등의 정보도 포함하고 있다. MS-DOS에서는 대개 아카이버가 압축 기능을 겸하고

있다. 예를 들면 lha라는 압축 아카이버는

```
C> lha a myarch *.*
```

를 수행하면 현재 디렉토리의 모든 파일을 압축 결합하여 myarch.lzh라는 새로운 아카이브를 생성한다.

역으로

```
C> lha x myarch
```

를 실행하면 원래의 파일들을 복원시킨다.

이 lha는 freeware(저작권은 있으나 자유로 복사하여도 무방한 소프트웨어)이다. 이에 비하여 미국의 압축 아카이버에는 shareware(계속 사용하려면 요금을 지불해야 하는 소프트웨어)가 많다. 이와 같은 압축 소프트웨어는 무수히 많다. 표 1에 주요 압축 소프트웨어와 대응하는 확장자를 나타낸다. 이외에도 MS-DOS의 확장자 .exe의 파일에는 실행하면 파일이 복원되는 자기 복원형 아카이브(self-extracting archive)가 있다. Mac의 확장자 .sea의 파일도 자기 복원형이다.

데이터 압축 기술은 GIF 및 JPG 등의 화상 파일과 폰트 파일에도 사용되고 있다. 최근의 모델은 FAX와 같이 데이터를 압축하여 전송한다. 또한, MS-DOS Ver.6.0에서는 dblspace라는 명령어를 사용하면 하드디스크 공간을 자동적으로 압축하여 마치 디스크 공간을 늘리는 것과 같다.

표2에는 UNIX 워크스테이션[Axil220: 50 MHz microSPARC: 59.1 MIPS]상에서 텍스

* 중신회원. 부경대학교 전자계산학과

트 압축 표준 데이터로 널리 사용되고 있는 Calgary Compression Corpus 중에서 대표적인 7개의 표본에 대해 압축을 수행한 결과를 나타낸다. 여기서, 시간은 UNIX의 time 명령에 의한 user time과 system call을 합한 값이며 단위는 [sec]이다. 또한, 압축율이란 압축계열의 길이를

원 입력계열의 길이로 나눈 백분율로서 그 값이 작을수록 많이 압축됨을 의미한다. 압축 유틸리티는 UNIX 시스템에서 널리 사용되어 온 허프만 부호에 의한 "pack"와 "zip", LZ77부호에 의한 "lha"와 "gzip" 및 LZW부호에 의한 "compress"를 사용하였다.

〈표1〉 주요 압축 소프트웨어와 압축 파일의 확장자

확장자	압축 소프트웨어	확장자	압축 소프트웨어
.lzh	LHA, LHarc	.arc	ARC, PKARC
.Z	compress	.pak	pack
.z	pack	.arj	ARJ
.gz	gzip	.sit	stuffit
.zip	PKZIP, zip	.pit	packit
.zoo	ZOO	.cpt	compact pro

〈표2〉 주요 압축 유틸리티의 압축율과 시간비교

utility-> data	compress	pack	lha	zip	gzip
bib (111,261)	0.4 41.8%(46,528)	0.2 65.5%(72,868)	5.5 36.6%(40,790)	1.4 31.6%(35,145)	1.9 31.5%(35,063)
book1 (768,771)	3.5 43.1%(332,056)	1.7 57.0%(438,487)	42.0 44.1%(339,126)	14.0 40.77%(313,460)	19.4 40.76%(313,376)
geo (102,400)	0.5 75.9%(77,777)	0.2 71.1%(72,836)	5.0 67.0%(68,624)	3.5 66.96%(68,575)	4.9 66.89%(68,493)
obj2 (246,814)	1.2 52.1%(128,659)	0.6 78.75%(194,378)	11.4 34.45%(85,032)	3.4 33.10%(81,714)	4.7 33.07%(81,631)
paper1 (53,161)	0.2 47.17%(25,077)	0.1 62.29%(33,457)	2.6 56.11%(19,279)	0.6 35.10%(18,662)	0.8 34.94%(18,577)
progc (39,611)	0.2 48.32%(19,143)	0.0 65.71%(26,030)	1.8 35.32%(13,993)	0.4 33.72%(13,359)	0.6 33.51%(13,275)
trans (93,695)	0.3 40.81%(38,240)	0.2 69.74%(65,343)	4.1 24.15%(22,630)	0.9 20.35%(19,069)	1.0 20.26%(18,985)
data (입력길이)	time[sec.] %(출력길이)				

2. 연 길이(run length) 압축

이 방법은 연속한 기호를 (기호, 개수)의 쌍으로 치환하는 가장 간단한 방법 중의 하나이다. 예를 들면, "abbbbc"를 연 길이 압축하면 a(b,4)c로 된다. 일반적인 문서 파일은 이것만으로는 거의 압축이 이루어지지 않지만 비트맵 폰트나 흑의

화소가 수십 개 연속한 후, 백의 화소가 수십 개 이어지는 만화와 같은 경우 효과가 크다. 또한, 데이터 베이스 파일은 NULL문자가 연속되는 경우가 많기 때문에 연(run)의 길이를 압축하는 경우가 있다.

윈도우즈의 BMP(BitMaP) 파일도 연 길이 압

축을 사용할 수 있으나, 실제로 압축을 사용하는 소프트웨어는 적은 듯하다. G3 팩시밀리의 기본 방식인 MH(Modified Huffman)법도 연 길이 압축을 사용한다. 주사선별로 연속한 백의 화소와 흑의 화소의 수를 교대로 출력하면서 픽셀의 수는 가변길이 부호로 나타낸다. 이 가변길이 부호는 8개의 표본 데이터로부터 국제전신연합(ITU-T)에서 정한 것이다.

2. LZ77기법

이스라엘의 Ziv와 Lempel의 1977년 논문^[2]에 의거한 방법으로 슬라이드 사전법 또는 LZ1법이라고도 불린다. LZ77 기법은 최초로 압축용으로 개발된 것이 아니라 계산론적 계산량에 의한 복잡도를 나타내기 위한 일례로 등장되었다. 따라서, 그 자체의 압축 성능은 실용적이지 못하기 때문에 수많은 개량이 이루어 졌다. 그중 가장 널리 이용되고 있는 LZSS 기법에 대하여 알아본다. 이 방법은 같은 문자열이 나타나면 그것을 (앞에 나타난 문자열의 시작 위치, 문자열의 길이)의 쌍으로 바꾼다. 예를 들면, “abbababaa”는 “ab(1,2)(2,3)(6,3)(10,1)로 되며, (x,y)는 x번째 문자에서 시작하여 y문자만큼 복사하라는 의미이다.

실제로 이것을 보내려면 문자 그대로인지 복사한 것인지를 나타내는 1비트의 플래그를 붙여서

<0,a> <0,6> <1,1,2> <1,2,3> <1,6,3> <1,10,1>의 형태로 나타낸다. ‘a’와 ‘b’ 등의 문자는 ASCII부호의 8비트 그대로 나타내고, 플래그 비트가 ‘1’일 때 <1,P,L>의 2번째 및 3번째 파라메터는 참조 문자열을 보관하는 버퍼 중의 일치하는 위치P와 문자열의 길이 L을 나타낸다.

실제적으로는 환형의 버퍼에 파일의 최근 4KB ~32KB분을 저장해 두고 그 중에서 가장 길게 일

치하는 문자열을 찾아 그 위치 P와 길이 L를 출력한다. 1988년에 T. Miki에 의해 구현된 LArc는 LZSS기법만을 이용하고 있으나, 그 후에 개선된 LHarc, LHA, ZOO, PKZIP, ARJ 및 gzip등은 LZSS에 허프만 부호 등을 조합하여 압축성을 개선하고 있다.

3. LZ78기법

파일 중에 나타난 문자열을 일정의 규칙에 의해 사전에 등록해 가면서 같은 문자열이 나타나면 사전 중의 번호를 출력하는 방법이다. 1978년에 Ziv와 Lempel이 고안한 것으로 동적 사전법 또는 LZ2법이라 불리기도 한다^[3]. 그러나, 실제로 사용되게 된 것은 1984년에 Welch에 의해 LZW법이 개발된 이후이다.

LZW법에서는 예를 들면 4096개의 요소를 갖는 사전을 준비한다. 사전의 0번에서 255번까지는 모든 1바이트 문자를 등록해 둔다. 그리고 나서 파일을 읽으면서 사전으로부터 가장 길게 일치하는 문자열을 찾아 그 등록번호를 출력한다. 이 최장일치계열에 새로운 1문자를 더한 문자열을 사전에 등록한다. 예를 들면 “ababababa”의 문자열 압축은 다음과 같다.

- (1) ‘a’의 등록번호 0을 출력한 후, ‘ab’를 사전의 2번째에 등록한다.
- (2) ‘b’의 등록번호 1을 출력한 후, ‘ba’를 사전의 3번째에 등록한다.
- (3) ‘ab’의 등록번호 2를 출력한 후, ‘aba’를 사전의 4번째에 등록한다.
- (4) ‘aba’의 등록번호 4를 출력한 후, ‘abab’를 사전의 5번째에 등록한다.

4096 크기의 사전이라면 등록번호를 출력하려면 12비트 필요하게 되지만, 초기부분에는 작은 번호밖에 사용되지 않기 때문에 9비트이면 충분하

다. 등록번호가 증가됨에 따라 10, 11, 12비트로 늘려 간다.

현재 고속 모뎀의 데이터 압축용으로 ITU-T 규격의 V.42bis에 LZW법이 사용되고 있으며, GIF(미국 CompuServe co Ltd.의 등록상표임) 형식의 화상 데이터도 LZW법의 일종으로 압축되고 있다. 또한, LZW법에 기초한 UNIX의 압축명령인 compress는 허프만 부호를 사용한 pack를 대신하여 gzip가 출현하기 이전까지 널리 사용되었다. ARC는 단순히 압축 도구로서 뿐만 아니라 복수의 파일을 1개로 묶어 주는 아카이버의 기능을 갖는 점에서 compress와 다르다. 그 후, PKWARE사가 SEA사의 ARC와 호환이면서 고속인 PKARC를 개발하여 널리 사용되었으나 1988년 특허권 소송에서 SEA사에 패소하여 사용권이 SEA사에 양도되게 되었다. 이 무렵 일본에서는 Lempel-Ziv부호에 대한 학계의 연구 결과가 봇물처럼 쏟아져 바로 프로그램 개발자들에게 큰 영향을 주게 되었다.

1988년 5월 1일에 NEC의 PC통신 서비스 「PC-VAN」에 H. Okumura에 의해 LZSS에 기초한 데이터 압축 프로그램이 소개되어, K. Miki씨가 파스칼과 어셈블리로 작성한 LArc가 작성되었다. 이것은 PKARC보다 압축률이 좋았기 때문에 PC통신에 널리 쓰이게 되었다. 그 후, H. Okumura에 의해 LZARI가 개발되었으며, A. Yoshizaki에 의해 LHarc(1988년 12월)로 개량되었다. 1989년 3월 LHarc는 K. Okubo에 의해 미국 PC통신 GENie에 소개되어 호평을 받아 급속도로 전세계에 보급되었으며, 1990년에는 PC Magazine Award of Technical Excellence를 수상했다. 이어서, Okumura, Yoshizaki등은 협력과 경쟁으로 LHarc를 개량하여 오늘날의 LHA에 이르게 되었다. LHarc에서 LHA로 개

명된 것은 압축 기능이 강화된 것 이외에 우여곡절이 있다. 미국의 SEA-PKWARE간의 저작권 싸움에서 이긴 SEA사의 ARC의 이름을 LHarc가 포함한다는 지적을 받아 LHarc Ver.2.0에서부터 LH로 개명되었으나, MS-DOS 5.0이 내부 명령어로 LH를 채용하였기 때문에 LHA로 재개명되게 되었다.

4. 허프만 부호(Huffman Code)

1952년에 D.A. Huffman에 의해 고안된 이 부호^[4]는 1948년 C.E. Shannon에 의해 체계화된 부호화 정리를 구체적으로 제시하고 있다. 그 기본 원리는 자주 나타나는 기호일수록 짧은 부호어(codeword)를 할당하고 드물게 나타나는 기호일수록 긴 부호어를 할당하여 평균적으로 부호어의 길이를 최소화하는 것이다. 그 방법은 대단히 간단하여

- (1) 출현빈도가 가장 낮은 2기호를 쌍으로 묶어 새로운 1기호로 취급한다.
- (2) 이때 새로운 기호의 출현빈도는 2기호의 합으로 한다.

라는 규칙을 마지막 1개의 기호로 될 때까지 반복하는 것이다. 허프만 부호는 UNIX의 옛 압축 소프트웨어인 pack에서 단독으로 사용되었으나, 현재는 compress, LHA 및 gzip 등에서 LZ77법과 조합하여 사용되고 있다.

Huffman 부호로 압축하려면 각 기호의 빈도 분포를 조사한 후 부호표를 작성하게 된다. 만약, 압축하고자 하는 파일이 영어 문장이라면 미리 많은 영어 문장으로부터 각 문자의 출현확률을 조사하여 부호표를 작성해 두는 것도 가능하다. 그러나, 범용의 압축 소프트웨어에서는 무엇을 압축할지 모르기 때문에 파일의 전체 또는 일부를 미리 읽어서 빈도분포를 조사하여 그것에 기초하여 부

호표를 작성한 후, 다시 읽으면서 압축한다. 이 경우 압축한 파일의 선두에 부호표 작성에 필요한 정보를 붙여 둘 필요가 있다.

LHA 및 gzip에서는 고속화를 위하여 부호어의 최대길이를 16비트로 제한하고 있다. 만약, 17비트 이상의 부호어가 나타나면 강제적으로 16비트로 고치고 대신에 15비트 이하의 부호어를 조금씩 길게 하는 기법을 사용하고 있다.

5. 동적 허프만 부호(Dynamic Huffman Code)

허프만 부호에는 다음과 같은 결점이 있다.

- (1) 빈도분포를 조사하기 위하여 파일을 미리 읽어야 한다.
- (2) 빈도분포가 파일의 도중에 변화하는 경우 적용이 어렵다.
- (3) 부호표 작성을 위한 빈도정보를 헤더에 붙여야 한다.

그래서, 허프만 트리의 형태를 동적으로 변화시키면서 압축을 수행하는 동적(적응형) 허프만 부호가 개발되었다^[5]. LHarc는 LZSS법에 동적 허프만 부호를 조합한 것이다. 그러나, 실제의 응용에서는 정적 허프만 부호가 훨씬 고속이면서 압축률도 좋은 것이 많기 때문에 LHA에서는 정적 허프만 부호를 사용하고 있다. MNP5 모델의 프로토콜에서도 동적 허프만 부호가 사용되고 있었으나 LZW류의 V.42 bis로 바뀌고 있다.

6. 산술부호(Arithmetic Code)

허프만 부호는 각 문자를 정수개의 비트로 이루어지는 부호어로 치환한다는 조건하에서 최적이다. 그러나, 예를 들어 0과 1이 랜덤하게 나열된 데이터를 압축하는 것을 생각해 보자. 0의 출현확률이 0.9이고, 1의 출현확률이 0.1이라 하자. 통상의 허프만 부호에서는 1비트 이하의 부호어는 없기 때문에 이 경우 압축이 되지 않는다. 그러나,

압축하고자 하는 파일에 이와 같은 0과 1이 100개 포함되어 있다고 하면 모든 비트가 0일 확률은 0.9×100 이고, 모든 비트가 1일 확률은 0.1×100 이 된다. 이와 같이 하여 모든 패턴의 확률을 계산할 수 있다. 이것에 가변길이 부호를 적용하면 압축이 가능할 것이다. 이것이 산술부호의 대략적인 원리이다^[6].

7. 선두이동법(Move To Front)

Move To Front(MTF)는 탐색의 기법으로 유한개의 기호로 이루어지는 사전에서 찾고자 하는 것이 나타나면 선두로 옮기는 기법이다. 따라서, 자주 나타나는 기호는 선두 부근에 밀집하고 드물게 나타나는 기호는 후미로 밀리게 된다. 그 결과, 고속의 탐색이 가능하고 사전상의 기호의 위치를 가변 길이로 나타내면 데이터 압축에 응용할 수 있다^[7].

예를 들면, "dddabda"라는 기호열을 압축해 보자. 각 기호를

$$a=0, b=1, c=2, d=3$$

의 수로 초기화한다. 선두 이동의 모습을 아래 표에 나타낸다.

부 호 표	입력기호	출력
0 a=0 b=1 c=2 d=3	d	3
1 d=0 a=1 b=2 c=3	d	0
2 d=0 a=1 b=2 c=3	d	0
3 d=0 a=1 b=2 c=3	b	2
4 b=0 d=1 a=2 c=3	d	1
5 d=0 b=1 a=2 c=3	d	0
6 d=0 b=1 a=2 c=3	a	2
7 a=2 d=0 b=2 c=3		

이와 같이 입력기호에 대한 출력은 작은 숫자, 특히 0이 많게 된다. 따라서, 가변길이 부호를

0 : 0 (1비트)

1 : 10 (2비트)

2 : 110 (3비트)

3 : 111 (4비트)

와 같이 부여하면 모든 기호에 2비트씩의 고정길이 부호를 사용하는 것보다 짧게 된다. 그러나, 선두 이동법은 같은 기호가 부분적으로 연속하는 데이터에는 유용하지만 일반적인 데이터에 그대로 적용하게 되면 큰 효과를 기대할 수 없다. 그래서, 1기호 단위로 처리하는 것보다 영어 단어 등과 같은 가변길이 기호열에 효과가 크게 나타난다. 또한, 다음 장에 기술하는 블록 정렬과 같은 전처리에 의해 부분적으로 기호를 밀집되게 처리한 후 적용하면 큰 압축효과를 얻을 수 있다.

8. 블록 정렬(Block sorting)

영어에서 'u' 라는 문자는 그다지 자주 나타나지 않지만 'q' 이후에는 거의 'u' 온다. 따라서, 바로 전의 기호에 의해 다음 기호를 예측하여 대응하는 부호어를 출력하면 효과적이다. 나아가, 예측에 사용하는 기호를 늘려 가면 그 효과는 더욱 커진다. 이와 같이 여러 기호의 상관을 이용하는 압축 기법은 텍스트 뿐만 아니라 음성 및 화상 데이터에 이르기까지 널리 사용되어 왔다. 그 중에 PPMC(Prediction by Partial Match, method C) 기법은 여러 압축 소프트웨어에 이용되고 있다.

PPMC 기법과 동등한 압축률을 달성하는 블록 정렬법이 캠브리지 대학의 Wheeler와 Burrows에 의해 연구 개발되어 새로운 압축 기법으로 주목되고 있다^[8]. 이 기법을 예를 들어 간단히 소개한다.

a, b, c, d의 4개 기호로 이루어지는 길이 N의 기호열 "acabdb"를 1기호씩 오른쪽으로 순회시키면 N개의 서로 다른 기호열이 생성된다. 이 기호열을 오름차순으로 정렬하면

0 : abdbac

1 : acabdb

2 : bacabd

3 : bdbaca

4 : cabdba

5 : dbacab

로 된다. 이 기호열의 마지막 기호들을 차례로 연결하면 "cbdaab"라는 새로운 기호열이 만들어진다. 이 기호열은 부분적으로 같은 기호가 연속되기 때문에 8장의 선두 이동법과 결합하게 되면 압축 효과가 더욱 커진다. 한편, 원 기호열을 복원하려면 정렬된 기호열 가운데 원 기호열의 위치정보 1도 함께 첨부되어야 한다.

9. 아날로그 신호압축의 기본

음악용의 CD(Compact Disk)는 좌우의 음을 각각 매초 44,100회의 비율로 16비트의 정수로 고쳐 기록한다. 즉, 16비트 2채널의 A/D(Analog to Digital) 변환하여 기록하는 것이다. CD데이터는 매초 1Mbit를 넘게 된다. 이것을 그대로 하드 디스크에 보관하거나 네트워크에 보내면 대단한 량이 된다.

TV의 화상은 수십만 개의 화소를 매초 수십 회 보내야 하기 때문에 데이터 량은 약 1억 비트/초가 된다. 또한 차세대 HDTV의 경우, 10배로 늘어난다. 이와 같은 대량의 데이터는 압축 없이는 디스크에 보관도 네트워크로 보내는 것도 어렵게 된다.

A/D변환에서는 횡축(시간축)도 종축도 실수치를 정수치로 바꾼다. 횡축의 정수화를 표본화, 종

축의 정수화를 양자화라 한다. 표본화의 과정에서 표본화 주파수의 절반(CD는 약 22KHZ)이상의 신호는 버려지게 된다. 또, 양자화 시에 ± 0.5 단위에 상당하는 양자화 잡음이 발생한다. 아날로그 신호를 디지털화 한 것은 A/D변환 시에 이만큼의 정보를 잃어버리게 되므로 압축에 의해 좀더 잃게 되어도 상관없는 경우가 많다. 이와 같이 정보의 손실을 수반하는 압축을 비가역(lossy) 압축이라 한다.

이것에 반하여 지금까지 서술하였던 압축법은 모두 「복원하면 완전히 원래대로 되돌아온다」는 조건을 만족한다. 이것을 무손실 압축(lossless compression) 또는 가역 압축이라고 한다. 가역 이든 비가역 이든 압축의 최종단계는 「자주 나타나는 문자일수록 짧은 비트열로 나타낸다」는 가변길이 부호를 이용한다. 이를 위해서는 빈번히 나타나는 문자(수치)와 그렇지 않는 것의 출현빈도의 차가 크게 되도록 전처리 할 필요가 있다. 시계열 데이터의 경우는 완전히 백색 잡음이 아닌 한 시간적으로 인접하는 수치의 차이는 그다지 크지 않을 것이다. 그래서, 차분

$$\Delta(n) = a(n) - a(n-1)$$

을 취한 후 가변길이 부호로 압축하면 좋다. 바꾸어 말하면, $a(n)$ 이 $a(n-1)$ 과 같다고 예측하여 그 오차를 출력하는 셈이다. 규칙성이 더 있는 데이터이면 $a(n-1)$ 만이 아니라 $a(n-2)$ 등도 사용하여 $a(n)$ 을 예측하면 좋다. 특히, 자기 회귀모델을 예측에 사용하는 선형예측부호(LPC : Linear Prediction Code)^[9]는 음성과 같이 거의 주기적인 데이터의 압축에 유효하다. 시간과 함께 성질이 변화하는 신호라면 가변길이 부호 및 자기 회귀모델의 계수를 적응적으로 변화시킨다. 2차원 화상압축의 경우도 $a(m, n)$ 을 예를 들면, $a(m, n-1) + a(m-1, n) - a(m-1, n-1)$

로 예측하여 오차만을 보내면 0에 가까운 값이 많아지기 때문에 가변길이 부호의 효과가 커진다.

10. 심리학과 압축

CD음에 대하여 여러 가지 다른 견해가 있다. CD음은 피치(pitch)가 이상하다는 주장이 모 일 간지에 게재되었으며, CD를 세제로 씻으면 음이 변화한다고 하는 이야기도 있다. 인간의 귀는 측정기와 다른 논리로 움직이기 때문에 정보량이 적을 아날로그 LP가 디지털 CD보다 좋은 음으로 들리는 것은 있을 수 있다. 그래서, 인간의 청각의 움직임을 심리학적으로 연구하면 손실 압축이지만 좋은 음으로 들릴 수 있는 것은 가능하다. 예를 들면, 인간의 감각은 대수적이기 때문에 절대치가 작은 입력은 조밀하게 양자화하고 큰 입력일수록 거칠게 양자화 한다. 이 원리는 전화기의 디지털 화에 이용되고 있다.

화상압축에서는 인간의 시력이 휘도 변화에 민감하고 색차 변화에 둔감한 것을 이용하여 색차에 대하여는 몇 개씩 속아 내어 표본화함이 일반적이다. 음성 압축에서는 귀의 강도가 주파수에 따라 변화하는 점과 작음 음이 큰 음에 묻혀 듣기 어렵게 되는 것을 이용한다. 예를 들면, MPEG·오디오에서는 가청 주파수를 32개의 서브 밴드로 분할하여 서브 밴드별로 인간의 귀의 특성을 감안하여 부호화 한다.

11. 화상의 LZW 압축

LZW^[10]법에 대해서는 앞에서 언급한 바와 같이 자주 나타나는 화소 패턴을 동적으로 사전에 등록하면서 같은 패턴이 나타나면 사전중의 번호를 출력한다. GIF형식의 화상 파일은 이 방법을 사용하고 있다. UNIX의 compress는 바이트 단위로 LZW 압축을 하지만 GIF는 화소 단위로 압

축한다. 가역 압축 중에는 비교적 고속이면서 간단한 선 그림이나 만화에 대해서 JPEG보다 잘 압축되는 경우가 많다. 또한, GIF(Graphics Interchange Format)는 CompuServe사가 저작권을 가지고 있으나 저작권을 표시하면 무료로 이용할 수 있다.

TIFF(Aldus사의 등록 상표)의 화상 파일은 LZW 및 JPEG 등의 압축 형식을 포함하고 있다. TIFF 5.0(1988년)에서는 LZW 압축이 기본 사양의 하나로 되어 있는데 Version 6.0에서는 확장사양으로 바뀌었다. 또, 사양서에는 LZW 압축을 사용하는 소프트웨어는 UNISYS의 허가를 얻어야 함을 기술하고 있다.

12. JBIG의 산술 압축

JBIG는 차세대 팩시밀리 압축법의 국제 표준화를 위한 전문가 그룹의 통칭 Joint Bi-level Image Experts Group으로부터 이름이 붙여진 방식으로 2치 화상을 위한 가역 부호화 방법이다. 어느 화소가 백인가 흑인가의 확률을 근처 화소 패턴으로부터 예측하여 적응 산술부호로 압축한다. 단, 산술 압축은 IBM이 기본 특허를 가지고 있으며 AT&T, 미쯔비시도 관련 특허를 가지고 있다. JBIG 사용자는 IBM에 1만 달러 지불해야만 한다.

JBIG에서는 칼러 화상은 색 평면별로 따로 압축한다. 농담이 있는 경우는 Gray부호로 변환하여 비트 평면별로 압축한다. 통상의 2진 부호에서는 예를 들면 127이 128로 되면 모든 비트 평면이 변화하지만 Gray부호를 사용하면 1개의 평면만이 변화하기 때문이다. 일반적으로 화소당 6비트 정도까지는 JPEG의 가역 모드보다 JBIG쪽이 잘 압축된다.

13. JPEG(Joint Photo-graphic Experts Group)

JPEG은 ISO(국제 표준화 기구)와 ITU 합동의 전문가 그룹으로 정지 화상을 압축하기 위한 국제 표준이다^[11]. JPEG에는 가역과 비가역의 두 방식이 있는데 가역 방식은 1/2정도밖에 압축이 안되며 IBM 특허의 산술 압축을 사용하기 때문에 그다지 사용되지 않는다.

JPEG이라 하면 일반적으로 비가역 방식을 가리킨다. 이것은 칼러 또는 농담 화상에 적합한 표준 압축법이다. 손으로 그린 선 그림이나 농담의 변화가 적은 만화류의 그림, 2치 화상 등은 JBIG이나 GIF가 더 잘 압축되는 경우가 많다. 자연 화상에 대해서는 JPEG가 압도적으로 좋으며, 24비트/화소의 자연 화상은 그대로 JPEG로 압축하는 쪽이 8비트/화소로 떨어뜨려 GIF로 압축하는 것보다 잘 줄어들며 화질도 양호하다.

JPEG의 비가역 방식은 먼저 화면을 8×8화소의 블록으로 분할하여 블록별로 이산 코사인 변환을 실시하여 그 결과를 양자화할 때 고주파 성분일수록 드물게 양자화하여 최후에 가변 길이로 부호화 한다. 칼러 화상은 인간의 눈의 특성을 고려하여 RGB의 3원색 정보를 휘도(명암) 정보와 2개의 색차 정보로 나누어 휘도는 조밀하게 색차는 드물게 한다.

14. 동화상 압축의 원리

TV화상과 같은 압축에는 화상의 프레임별 압축뿐만 아니라, 프레임 간의 상관을 이용한 압축이 사용된다. 정지화상과 같이 화면을 블록으로 나누어 블록별로 압축한다. 움직임이 적은 화면은 바로 직전의 프레임과 거의 차이가 없다. 따라서, 직전 프레임과의 차분을 정지화상으로 압축하여 보낸다. 움직임을 포함한 블록은 직전 블록을 움직임 방향으로 조금 이동시킨 것과의 차분을 보낸

다. 물론, 움직임 벡터도 함께 보내야 한다. 차분만을 보내는 경우는 중간 영상을 볼 수 없고 오차가 축적할 가능성도 있다. 그래서, 정지화상만의 압축 화상을 때때로 삽입한다.

동화상 압축의 표준에는 TV 전화용의 거친 화질의 H.261 및 MPEG이 있으며, 특히, MPEG에서는 과거·미래의 양측에서 중간의 화상을 예측하는 등의 여러 가지 공리가 되어 있다.

15. MPEG(Moving Picture Experts Group)

JPEG이 정지화상의 표준 압축인데 대하여 MPEG은 동화상·음성의 압축 표준으로 이른바 멀티 미디어 및 디지털 텔레비전을 지탱하는 기술이다^[12].

MPEG에는 CD-ROM을 매체로 한 1.5Mbps 정도까지의 MPEG-1과 현행 TV에서 HDTV까지의 넓은 영역을 포함하는 MPEG-2가 있으며, 64Kbps 정도의 MPEG-4는 현재 표준화 작업 중이다. MPEG-1은 CD 노래방 등에 사용되고 있는데 화질은 VHS 비디오 이하이다. MPEG-2는 현재 LSI화 경쟁이 한창이다. 참고로, HDTV는 일본의 경우 MUSE라는 아날로그 방식이 NHK를 중심으로 진행되고 있으나, 미국과 유럽에서는 MPEG-2에 의한 디지털 방식으로 결정되었다. 그러나, 일본에서도 2007년부터 디지털 방송으로 MPEG-2가 사용될 것 같다.

16. 프랙탈 압축

프랙탈 압축은 Georgia 공과대학의 수학자 Barnsley가 생각한 방법이다^[13]. 자연계에는 자기 자신의 Affine 변환을 자신의 일부분으로서 포함하는 프랙탈 도형이 많다. Affine 변환이란

$$x' = ax + by + c$$

$$y' = dx + ey + f$$

와 같은 정수항을 포함하는 1차 변환이다. 예를 들면 고사리의 잎은 자기 자신의 4종류의 Affine 변환을 포함한다. 이와 같은 Affine 변환의 집합을 IFS(Iterated Function System)라 한다. 임의의 도형은 적당한 IFS의 작업으로 근사 시킬 수 있다는 것이 콜라주의 정리이다.

IFS로부터 화상을 생성하는 것은 비교적 고속이지만, 주어진 화상으로부터 IFS를 구하는 것은 어렵다. 이것이 이루어지면 화상 대신에 IFS의 파라미터(a, b, c, d, e, f의 집합)를 이용하면 내지 1/1,000의 압축이 가능하다. 이 방법도 물론 비가역 압축이지만, 원화상보다 높은 해상도로 복원할 수도 있다. 그러나, 화상으로부터 IFS를 구하는 실용적인 알고리즘이 없다. 그래서, Barnsley의 대학원생 Jacquin은 불럭을 불럭으로 복사하는 대단히 한정된 Affine 변환의 집합에 한하여 탐색하는 방법을 생각했다. 그러나, 이 방법에서는 수십분의 1 정도밖에 줄어들지 않는다.

Jacquin의 방법은 벡터 양자화에 의한 압축과 비슷하나, 자기 자신의 일부를 사전 대신에 사용하는 점이 새롭다. Barnsley 등은 실제로 이 원리에 기초한 화상 압축용의 하드웨어 및 소프트웨어를 판매하는 Iterated System이라는 회사를 만들어 제품을 공급하기 시작했다. 또한, Iterated System사의 프랙탈 압축은 이미 Microsoft사의 Encarfa 및 Parson사의 Screen Saver 등에 이용되고 있으며, 동화상도 MPEG의 배의 속도로 복원할 수 있다고 한다.

17. 끝내며

문서, 음성 및 화상 데이터를 독립적으로 취급해 오던 기존의 연구 및 응용이 컴퓨터의 성능 향상과 정보통신망의 발달에 따라 통합적으로 다룰 수

있는 이른바 멀티미디어의 시대를 맞이하고 있다. 이와 같은 멀티미디어의 응용을 가능하게 하는 기반 기술 가운데 데이터 압축의 기술이 부각되게 되었다. 이는 데이터 속에 내재된 중복도를 제거함으로써 제한된 통신로를 통하여 주어진 시간에 보다 많은 데이터를 보낼 수 있게 하는 기술이다. 이 기술이 인간의 생활에 적용된 것은 18세기 모르스 부호를 사용한 고대사회로까지 거슬러 올라간다. 그러나, 학문적으로 취급되게 된 것은 1948년 C. E. Shannon이 정립한 정보이론으로부터이다. 또한, 그것이 이론의 굴레를 벗어나 실용적으로 응용되기 시작한 것은 1970대에 들어서이다. 특히, 문서 압축용으로 널리 사용되게 된 각종 압축 유틸리티들은 1976년에 등장한 Ziv-Lempel 부호에 기초한 것이 주류이다.

그 후, 화상 압축기법이 널리 연구 및 개발되면서 국제 표준화의 필요성이 대두되었으며, 1980년대 후반부터 활발히 이루어져 그 결실과 더불어 정보 시스템에 필수 불가결한 기반요소가 되었다. 그것을 계기로 이 분야에 대한 연구는 폭발적으로 증가하게 되었으며, 매스컴을 통해 우리는 쉽게 접할 수 있게 되었다. 한편, JPEG 및 MPEG의 국제 표준방식과는 별도로 화상 압축을 위하여 웨이브렛 변환, 프랙탈 압축 및 카오스 이론에 의한 새로운 방식에 대한 연구도 활발히 이루어지고 있는 상황이다.

참고문헌

1. 박 지 환, "문서 데이터 압축 알고리즘", 성안당, p.289, 1995
2. J. Ziv, A. Lempel, "A universal algorithm for sequential data compression", IEEE Trans. on Information Theory, IT-23(3), pp.337-343, 1977
3. J. Ziv, A. Lempel, "Compression of individual sequences via variable-rate coding", IEEE Trans. on Information Theory, IT-24(5), pp.530-536, 1978
4. D. A. Huffman, "A method for the construction on minimum-redundancy codes", P. roc. of IRE, 40, pp.1098-1101, 1952
5. D. E. Knuth, "Dynamic Huffman coding", Journal of Algorithms, 6, pp.163-180, 1985
6. I. H. Witten et al, "Arithmetic coding for data compression", Communication of the ACM,30(6), pp.520-540, 1987
7. J. L. Bentley et al, "A locally adaptive data compression scheme", Communication of the ACM, 29, pp.320-330, 1986
8. M. Burrows, D. J. Wheeler, "A block-sorting lossless data compression algorithm", SRC Research Report 124, 1994
9. S. Sugiyama, "Introduction to audio coding algorithm", Interface, pp.121-136 1994.8 (In Japanese)
10. T. A. Welch, "A technique for high-performance data compression", IEEE Computer, 7(6), pp.8-19, 1984
11. G. K. Wallace, "The JPEG still picture compression standard", Communication of the ACM, 34, pp.30-44, 1991
12. D. L. Gall, "MPEG: A video compression standard for multimedia applications", Communication of the ACM, 34(4), pp.46-58, 1987
13. M. Barnsley, "Fractals everywhere", Academic Press(2nd Ed), 1993



박 지 환 종신회원

- 1984년 경희대학교 전자공학과 졸업(공학사)
- 1987년 日本 國立電氣通信大學 情報工學科(工學修士)
- 1990년 日本 橫濱國立大學 電子情報學科(工學博士)
- 1990~1996년 부산수산대학교 전자계산학과 전강
 조교수, 부교수
- 1994~1995년 日本 東京大學生産技術研究所 客員研究員
- 1996~현재 日本 東京大學生産技術研究所 協力研究員
- 1998년 1월 日本 國立電氣通信大學 情報시스템학연구과
 방문연구
- 현재 부경대학교 전자계산학과 부교수
- 주관심분야 : 멀티미디어 압축, 암호화 응용, 오류제어부호 등

• ‘98 정기총회 및 춘계학술발표회 •

- 일 시 : 1998년 6월 5일 09:00~19:00
- 장 소 : 부산대학교 본관 대회의실
- 문의 및 접수처 : 한국멀티미디어학회 사무국
 - 부산시 사상구 모라 3동 75-8 (우)617-083
Tel: (051)302-9572 Fax: (051)303-5439
 - 부경대 전자계산학과
부산광역시 남구 대연3동 599-1 (우)608-737
Tel: (051)620-6395 Fax: (051)628-8147