

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁶
H04N 7/24

(11) 공개번호 특1998-081236
(43) 공개일자 1998년11월25일

(21) 출원번호	특1998-012597
(22) 출원일자	1998년04월09일
(30) 우선권 주장	8/832,681 1997년04월09일 미국(US)
(71) 출원인	휴렛-팩카드컴파니 바카엔쏘니제이
(72) 발명자	미국 캘리포니아주 94304 팔로 알토 하노버 스트리트 3000 러스트로버트에이
(74) 대리인	미국 아이다호주 83704 보이세 알라마 5061 김창세, 장성구

심사청구 : 없음

(54) 데이터 압축/복원 방법 및 압축 장치

요약

본 발명은 처리되는 데이터 세트의 크기에 최적화된 컨텍스트 모델(context model)을 선택함으로써, 보다 큰 데이터의 압축율에 대한 영향을 작게하면서 작은 데이터 세트들에 대해 향상된 압축율을 제공한다. 원래 데이터의 크기가 결정된다. 데이터의 크기를 기초로하여 컨텍스트 모델이 선택된다. 컨텍스트 모델을 사용하여 원래 데이터를 압축 데이터로 압축한다. 소정의 수의 컨텍스트 모델이 정의될 수 있으며, 적절한 값이 선택된다. 압축 데이터를 복원하기 위해서는, 원래 데이터의 크기를 기초로하여 컨텍스트 모델을 선택해야 한다. 정확한 컨텍스트 모델을 검출하는 두 가지의 방법이 기술된다. 우선, 원래 데이터의 크기는 압축 데이터의 일부이다. 대안적으로, 원래 데이터를 압축하는데 이용된 컨텍스트 모델을 식별하는 지시자(indicator)는 압축 데이터의 일부이다.

대표도

도1

명세서

도면의 간단한 설명

- 도 1은 산술 압축기의 블록도.
도 2는 사용된 확률표의 크기 증가를 도시한 도면.
도 3은 주어진 컨텍스트 모델을 사용하여 사용된 확률표의 크기를 조정하는 방법을 도시한 도면.
도 4는 바람직한 실시예의 논리 동작의 흐름도.
도 5는 본 발명에 따른 산술 압축기의 블록도.

도면의 주요 부분에 대한 부호의 설명

101 : 비트	1101 : 이전 행의 영상
1102 : 제어기	1107 : 영상/시프트 레지스터
1113 : 확률표	1115 : 컨텍스트 모델
1114 : 압축기	1116 : 산술 압축기

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 산술 부호화(arithmetic encoding) 방법 및 장치에 관한 것으로서, 특히, 처리되는 데이터의 양에 적응적인 산술 압축기/복원기(arithmetic compressor/decompressor)에 관한 것이다.

종래의 기술에는 디지털 데이터 신호의 스트림을 압축된 디지털 코드 신호로 부호화하고, 압축된 디지털 코드 신호를 원래 데이터로 다시 복호화하는 데이터 압축 시스템이 알려져있다. 데이터 압축은 주어진 포맷의 데이터를 원래의 공간보다 적은 공간을 필요로하는 대안적인 포맷으로 변환하는 소정의 처리를 의

미한다. 데이터 압축 시스템의 목적은 데이터를 유지하는데 필요한 저장 공간이나, 주어진 디지털 정보 부를 전송하는데 필요한 시간을 효율적으로 절약하는 것이다.

범용 디지털 데이터 압축 시스템을 실제로 이용하기 위해서는 소정의 기준을 만족해야 한다. 데이터 압축 시스템은 교환성(reciprocity)을 가져야 한다. 데이터 압축 시스템이 교환성을 갖기 위해서는 정보의 교체나 손실없이 압축 데이터를 자신의 원래의 형태로 재확장(re-expand) 또는 복호화될 수 있어야 한다. 복호화된 데이터와 원래 데이터는 동일해야 하며, 서로를 구별할 수 없어야 한다. 교환성은 사용된 정보 이론(information theory)에 잠음이 전혀 없음을 의미한다. 몇몇 응용에서는 교환성을 엄격하게 고수할 필요가 없다. 이것은 그래픽 데이터를 다루는 응용에서 그러하다. 인간의 시각은 잠음에 민감하지 않기 때문에, 압축 복원 처리를 하는 동안 약간의 정보 교체 또는 손실을 수용할 수 있다.

시스템은 데이터 압축 및 복원 시스템과 통신하고 있는 장치에 의해 제공되고, 수용되는 데이터율에 대하여 충분한 성능을 제공해야 한다. 데이터가 압축될 수 있는 비율은 압축 시스템의 입력 데이터의 처리율에 의해 결정되며, 전형적으로 초당 수백만 바이트(megabytes/sec)이다. 전형적으로 데이터율이 1 megabyte/sec를 초과하는 오늘날의 디스크, 테이프 및 통신 시스템에서 달성되는 데이터율을 유지하기 위해서는 충분한 성능을 갖고 있어야 한다. 따라서, 데이터 압축 및 복원 시스템은 전체 시스템에 나쁜 영향을 끼치지 않도록 충분한 데이터 대역폭을 가져야 한다. 전형적으로, 데이터 압축 및 복원 시스템의 성능은 데이터를 압축 및 복원하는데 필요한 연산과, 통계 데이터를 저장하고 데이터를 압축 및 복원 처리 하는데 이용되는 RAM(random access memory) 등과 같은 시스템 소자의 속도에 의해 제한을 받는다. 압축 장치의 성능은 압축기에서 요구되는 입력 문자당 프로세서 주기(processor cycles)의 수에 의해 결정된다. 주기의 수가 적을수록 성능이 향상된다.

데이터 압축 및 복원 시스템 설계시 또 다른 중요한 기준은 압축율(compression ratio)에 의해 결정되는 압축 효율이다. 압축율은 압축되지 않은 형태의 데이터 크기가 압축된 형태의 데이터 크기로 나누어진 비율이다. 데이터가 압축되려면, 데이터는 중복성을 갖고 있어야 한다. 압축 효율은 압축 처리시 얼마나 효율적으로 입력 데이터의 중복성을 이용했는지에 따라 결정된다. 전형적인 컴퓨터 저장 데이터에서, 중복성은 각각의 심볼, 예를 들면, 디지털(digits), 바이트(bytes) 또는 문자(characters)를 불규칙하게 사용하거나, 공통적인 워드(words), 여백 레코드 필드 등과 같은 심볼 시퀀스가 빈번하게 재현됨으로 인해 발생된다.

허프만 방법(Huffman method), 톤스탈 방법(Tunstall method) 및 렘펠-지브 방법(Lempel-Ziv method)과 같은 종래 기술의 범용 데이터 압축 절차는 잘 알려져 있다. 허프만 방법은 널리 알려져 사용되고 있으며, 이에 관한 내용은 D. A. Huffman에 의해 기술된 A Method For Construction Of Minimum Redundancy Codes란 명칭의 Proceedings IRE, 40, 10 pages 1098-1100(Sept. 1952)의 논문을 참조하면 알 수 있다. 톤스탈 알고리즘은 B. P. Tunstall에 의해 기술된 Synthesis of Noiseless Compression Codes란 명칭의 Georgia Institute of Technology(Sept. 1967)의 논문을 참조하면 알 수 있다. 렘펠-지브 절차는 J. Ziv 및 A. Lempel에 의해 기술된 A Universal Algorithm For Sequential Data Compression이라는 명칭의 IEEE Transactions on Information Theory, IT-23, 3, pages 337-343(May, 1977)의 문헌을 참조하면 알 수 있다.

1 차적인 범용 데이터 압축 절차들 중 하나는 허프만 방법이다. 간단히 설명하면, 허프만 절차는 전체 길이의 심볼 세그먼트를 가변 길이 워드(variable length words)로 매핑(map)한다. 허프만 데이터 압축 절차는 두 가지 제한을 갖고 있다. 첫째, 허프만 절차는 압축될 입력 데이터가 고정 길이의 심볼 세그먼트로 파싱(parsing)된다는 제한 조건에 따라 동작한다. 허프만 절차는 이들 제한 조건에 따라 최적의 압축율이 획득될 수 있도록 하지만, 제한 조건이 완화될 경우 다른 절차를 이용하여 보다 높은 압축율을 얻을 수도 있다. 둘째, 허프만 코딩은 소스 데이터의 통계적 특성을 모두 알고 있어야 한다. 허프만 절차는 각각의 고정 길이의 입력 세그먼트가 나타날 확률을 알고 있다는 가정하에서 동작한다. 실제로, 이와 같은 허프만 절차의 요건은 데이터를 처리하는 동안 필요한 통계를 축적하는 절차의 적응적인 버전(version)을 이용함으로써 충족될 수 있다. 그러나, 이것은 번거로운 일이고, 상당한 작업 메모리 공간이 필요하며, 적응 동안 최적화가 부차적으로 수행된다.

가변 길이의 심볼 세그먼트를 고정 길이 워드로 매핑하는 톤스탈 알고리즘은 입력 세그먼트 대신에 출력 세그먼트가 고정 길이로 된다는 제한 조건을 갖는 허프만 절차를 보완한 것이다. 허프만 절차와 마찬가지로, 톤스탈 절차는 소스 데이터의 확률에 대한 사전 지식이 필요하다. 이러한 사전 지식에 대한 요건은 데이터를 처리하는 동안 통계를 축적하는 적응적인 버전을 이용함으로써 어느 정도 충족될 수 있다.

렘펠-지브 절차는 가변 길이의 심볼 세그먼트를 가변 길이의 이전 워드로 매핑한다. 이 절차는 입력 또는 출력 세그먼트에 대한 제한 조건이 없을 때 점근적으로(asymptotically) 최적화 되는 방안이다. 이 절차에서 입력 데이터 스트링(data string)은 적응적으로 성장된 세그먼트로 파싱되며, 각각의 세그먼트는 입력 데이터로부터 하나의 새로운 심볼에 의해 부가된 입력 스트링의 이전 부분의 정확한 사본(copy)으로 구성된다. 생성된 사본은 가장 길 가능성이 있으며, 이전에 파싱된 소정의 세그먼트와 일치하기 위해 제한되지 않는다. 출력내의 세그먼트를 교체하는 코드 워드는 이전에 복사된 부분의 개시 포인터, 사본의 길이 및 새로운 심볼로 구성된 정보를 포함한다.

허프만 또는 쉐논-페노(Shannon-Fano) 코딩은 완벽한 데이터 압축 방법처럼 보일 수도 있다. 그러나, 그렇지 않다. 전술한 바와 같이, 이들 코딩 방법은 심볼의 확률이 $1/2$ 의 정수 제곱(integral power)일 경우에만 최적화된다.

산술 코딩 기법은 이러한 제한 조건을 갖지 않는다. 이 기법은 메시지를 하나의 단일 유닛으로서 처리함으로써 동일한 효과를 달성하며(허프만 코딩의 경우, 단일의 가능한 모든 메시지를 열거해야함), 이에 따라, 소정의 소스에 대한 압축 효과에 대해 이론적인 엔트로피 경계값(theoretical entropy bound)을 얻을 수 있다.

산술 코딩에서, 번호 라인을 따라 바람직하게 작고 적은 간격을 정의하기 위해 결정은 순차적으로 부호화된다. 산술 코딩에 대한 추가적인 정보는 G. G. Langdon, Jr.에 의해 기술된 An Introduction To Arithmetic Encoding이란 명칭의 IBM Journal of Research and Development, Vol. 28, n. 2, March 1984,

135-149의 문헌과, D. R. Helman, G. G. Langdon, Jr. 및 J. J. Rissanen에 의해 기술된 Arithmetic Compression Code Control Parameters Approximation이란 명칭의 Vol. 23, n. 11, April 1981, 5112-5114의 문헌과, Langdon, Jr. 등에 의해 기술된 Arithmetic Coding Encoder And Decoder System이란 명칭의 미국 특허 출원 제 4,905,297 호에 개시되어 있으며, 이는 본 명세서에서 참조로 인용된다.

위의 문헌에서 기술된 바와 같이, 산술 코딩에서 각각의 결정은 복수의 가능한 배타적 결과 또는 이벤트(events)를 갖는다. 각각의 결과 또는 이벤트는 데이터에서 심볼로 표현된다. 영상 환경에서, 예를 들어, 각각의 결정은 주어진 픽셀(pixel)이 검정색인지의 여부와 대응할 수 있다. 결정 결과는 픽셀이 검정색인 경우 Y(또는 YES) 심볼로 표현되고, 픽셀이 검정색이 아닌 경우 N(또는 NO) 심볼로 표현된다.

따라서, 복수의 결정들은 심볼 시퀀스, 예를 들면, YNNY...로 표현될 수 있다.

종래의 산술 코딩 기법에 따라, 확률 라인상에는 현재의 간격이 정의된다. 제 1 현재 간격은 0 내지 1이다. 현재의 간격은 다음의 결정에 대해 가능한 하나의 결과에 대응하는 세그먼트로 분할된다. 각각의 결정에 대해서는 단지 두 개의 가능한 결과만이 존재하며, 현재의 간격은 두 개의 세그먼트로 분할된다. 각각의 세그먼트의 길이는 세그먼트 자신과 관련된 확률을 기초로 한다. 각각의 확률은 고정된 상태로 유지하거나, 혹은 결정 데이터가 입력될 때 적응적일 수 있다.

압축될 빈도가 커짐에 따라 세그먼트 대 심볼의 상관 관계도 커지게 된다. 전술한 문헌(An Introduction To Arithmetic Encoding)에는 4개의 심볼에 대한 산술 코딩의 예가 있으며, 이 문헌에서 각각의 결정은 a 이벤트(50%의 확률), b 이벤트(25%의 확률), c 이벤트(12.5%의 확률) 또는 d 이벤트(12.5%의 확률)를 생성할 수 있다. 4개의 이벤트를 이진 형태로 표현하려면, 각각의 결정에 대해 2 비트가 필요하며, 각각의 이벤트는 00, 01, 10, 11로 표현된다. aab와 같은 3 개의 결정에서, 직접 부호되는 데이터는 6 비트인 00 00 01일 수 있다. 그러나, 상기 문헌의 137 페이지에 기술된 바와 같이, 산술 코딩 방식은 시퀀스 aab를 값 0.001로 표현할 수 있다. 정보를 6 비트가 아닌 3 비트로 표현할 수 있다. 이와 같은 비트의 보존(conservation)은 관련된 확률이 비교적 높은 연속적인 이벤트가 발생할 때 이루어진다.

발생 확률이 낮고 비교적 짧은 라인 세그먼트를 갖는 이벤트가 많이 발생하는 경우에는 보존이 나빠진다. 전술한 확률에 의하면, 이벤트의 시퀀스 dd는 11 11과 같은 부호화된 데이터로 표현될 수 있으나, 산술 코딩에 의해, dd 이벤트는 111111로 표현될 수 있다. 빈도가 커짐에 따라 발생될 이벤트에 실질적으로 대응하는 세그먼트가 커지게 되면, 발생 확률이 높은 심볼이 생성될 때 보존이 이루어짐에 따라 발생 확률이 낮은 심볼에 대해 필요한 추가적인 비트도 중요하다.

전체 압축 실행을 통해 산술 코딩은 데이터에 대해 적응적이며, 예전의 데이터를 결코 잊지 않는다. 이것은 일정하게 사전(dictionary)의 내용을 상실하는 다수의 LZ 기반 방안들과는 다르다. LZ는 사전을 재설정하여 다음 데이터 섹션에 적응되도록 한다. LZ 방안은 100KB의 데이터 뿐만 아니라 1KB의 데이터도 압축한다. 산술 코딩은 실행을 통해 그 확률을 계속해서 향상시키며, 이는 산술 코딩이 자신의 적응을 최적화할 기회를 갖지 못하기 때문에 1KB의 데이터상에서 잘 수행되지 않음을 의미한다. 그러나, 산술 코딩은 1KB를 LZ보다 잘 압축한다.

산술 코딩은 확률표(probability table)를 이용하여 영상에 대한 통계를 저장한다. 각각의 비트가 압축됨에 따라, 비트에 대한 처리 방법을 결정하기 위해 표를 액세스한다. 표가 클수록, 최종/최적의 상태로 더욱 늦게 이동한다. 그러나, 표가 클수록, 각각의 비트에 대해 더 많은 정보를 이용할 수 있기 때문에 더 큰 영상에 대해 압축율이 향상된다. 간단한 실험에 의하면 더 큰 표로부터의 이점에 대한 한계는 대략 10KB이며, 이러한 한계점 이후 더 큰 표는 압축율을 크게 향상시킨다.

프린터 동작시, 몇 가지 다른 형태의 영상이 생성된다. 어떤 영상은 크기가 100KB 내지 200KB이며, 어떤 영상은 400B 또는 그 이하의 작은 크기를 갖기도 한다. 이들 작은 영상은 폰트 캐시(font cache)에 의한 것이다. 각각의 독특한 문자가 처음에 생성되어 프린트가 시작되기 전에 폰트 캐시에 저장된다. 페이지에 따라, 폰트 캐시에 의해 사용되는 공간의 양은 페이지의 성공적인 표현을 위한 관심사이다.

LAN 환경에서 동일한 프린터가 사용된다면 폰트의 사용은 더욱 중요해진다. 몇몇 다른 사용자들은 그들 자신이 선호하는 폰트 및 포인트 크기로 프린트 작업을 전송할 수 있다. 각각의 새로운 작업시, 프린터는 요청된 문자 세트가 이전 작업으로부터의 폰트 캐시에 이미 존재하는지를 결정한다. 그것이 존재하지 않으면, 문자를 표현하는데 시간이 걸린다. 프린터의 메모리가 폰트 캐시 문자들로 채워짐에 따라, 그 문자들이 요구에 즉각적으로 응하지 못하는 시점이 오며, 그 문자들은 다른 문자를 위한 공간을 위해 폰트 캐시로부터 제거되어야 한다. 따라서, 더 긴 폰트 캐시 문자가 머물수록-특히, 몇몇 작업을 통해-더 빠른 사용자가 그들의 프린트 출력을 받게된다. 폰트 캐시를 생성하기 위해 상당히 오랜 시간을 기다리는 것은 일반적이지 않다. 이것은 전원을 켜 후에 처음 페이지를 프린트할 때 가장 많이 알 수 있다.

발명이 이루고자 하는 기술적 과제

본 발명은 원래 데이터를 압축 데이터로 압축하는 방법에 의해 수행된다. 우선, 원래 데이터의 크기가 결정된다. 결정된 크기를 기초로하여, 콘텍스트 모델이 선택된다. 원래 데이터는 콘텍스트 모델을 이용하여 압축 데이터로 압축된다. 소정의 수의 콘텍스트 모델이 정의될 수 있으며, 적절한 선택값이 선택된다.

압축 데이터를 복원하기 위해서는 원래 데이터의 크기를 기초로하여 콘텍스트 모델을 선택해야 한다. 정확한 콘텍스트 모델을 검출하는 두 가지 방법이 기술된다. 원래 데이터의 크기가 사전결정된 양보다 작으면 제 1 콘텍스트 모델이 선택되며, 원래 데이터의 크기는 압축 데이터의 일부이다. 원래 데이터의 크기가 사전결정된 양보다 크면 제 2 콘텍스트 모델이 선택된다. 대안적으로, 압축 데이터로부터 지시자(indicator)가 복원된다. 지시자는 원래 데이터 압축시 이용된 콘텍스트 모델을 식별한다.

또한, 데이터를 압축 또는 복원하기 위한 장치가 제공되며, 이 장치는 확률표, 제 1 콘텍스트 모델 및 제

2 컨텍스트 모델을 포함하는 산술 압축기로 형성된다. 산술 압축기에 접속된 시프트 레지스터가 제공된다. 시프트 레지스터는 데이터를 수신한다. 산술 압축기에는 제어기가 접속된다. 제어기는 데이터의 크기가 사전결정된 양보다 작은 경우 데이터 압축시 제 1 컨텍스트 모델을 이용하도록 산술 압축기에 신호를 제공하며, 산술 압축기는 대안적으로 데이터 압축시 제 2 컨텍스트 모델을 이용한다.

발명의 구성 및 작용

이하의 상세한 설명을 첨부된 도면과 함께 고려함으로써 본 발명을 보다 잘 이해할 수 있을 것이다.

본 발명은 본 명세서에서 기술된 특정한 실시예에 국한되지 않는다. 도 1을 참조하면, 본 발명의 바람직한 실시예의 하드웨어 구현시의 블록도가 도시되어 있다. 산술 압축기(1116)는 비트(101)를 압축하고자 하며, 영상/시프트 레지스터(1107)로부터의 데이터는 컨텍스트 모델(1115)로 전달된다. 컨텍스트 모델(1115)은 영상/시프트 레지스터(1107)로부터의 데이터를 확률표(1113)로 매핑한다. 압축기는 확률표(1113) 및 컨텍스트 모델(1115)에 따라 비트(101)를 압축한다. 그 후, 압축 데이터는 일반적으로 메모리 장치(도시되지 않음)에 기록된다. 비트(101)는 영상/시프트 레지스터(1107)로 시프트되며, 영상으로부터의 새로운 비트가 (101)로 시프트된다.

일반적으로, 복원은 압축과 같은 방식으로 수행된다. 그러나, 복원하는 동안, 산술 압축기는 확률표(1113) 및 컨텍스트 모델(1115)을 이용하여 압축 데이터에서 판독하여 복원한 후 비트(101)를 저장한다. 전술한 바와 같이, 일단 비트(101)가 복원되면 영상/시프트 레지스터(1107)의 데이터는 왼쪽으로 시프트된다. 영상/시프트 레지스터(1107)로부터 출력된 데이터는 일반적으로 메모리 장치에 저장된다.

압축기(1114)는 영상을 압축하거나 복원할 때, 확률표(1113)를 이용하여 영상의 통계를 저장한다. 각각의 비트가 압축/복원됨에 따라, 비트를 처리하는 방법을 결정하기 위해 확률표(1113)를 액세스한다. 확률표(1113)가 커질수록, 더 늦게 최종/최적의 상태에 도달한다. 그러나, 확률표(1113)가 커질수록 각각의 비트에 대해 보다 많은 정보를 이용할 수 있으므로, 더 큰 영상에 대해 압축율이 향상될 것이다.

컨텍스트 모델(1115)은 압축되는 비트 주변의 데이터의 비트를 관찰함으로써 인덱스(index)를 확률표(1113)로 나타낸다. 인덱싱에 대한 중요한 양상은 주소가 할당된 위치가 부호화/복호화되는 비트의 값과 관련된 유용한 정보를 포함하고 있다는 것이다. 특히, 확률표를 인덱스하는데 사용된 비트는 부호화/복호화될 수 있는 비트를 신뢰성있게 예측을 할 수 있도록 중요한 정보를 제공해야 한다. 예측에 대한 신뢰성이 높을수록, 영상 압축 능력이 향상된다.

본 발명의 바람직한 실시예는 처리되는 데이터의 크기에 최적화된 컨텍스트 모델을 선택한다. 도 3을 참조하면, 2 차원 컨텍스트 모델(300)이 더 큰 데이터 세트에 대해 확장된다. 각각의 픽셀은 확률표의 어드레스 라인에 접속되어 있다. 픽셀상의 번호는 소정의 어드레스 비트가 그 픽셀에 의해 제어되고 있음을 나타낸다. 도 3의 a를 참조하면, 100 바이트 미만의 데이터 세트에 대해, 어드레스 비트 9, 8, 7, 6, 1 및 0은 0으로되며, 이것은 도 2의 a에 도시된 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 100 바이트 데이터와 1 킬로바이트 데이터 사이의 데이터 세트들은, 도 3의 b에 도시된 바와 같이 어드레스 비트 9, 8 및 0이 0으로되며, 도 2의 b에 도시된 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 1 킬로바이트 데이터와 4 킬로바이트 데이터 사이의 데이터 세트들의 경우, 도 3의 c에 도시된 바와 같이 어드레스 비트 0이 0으로되며, 도 2의 c에 도시된 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 그 후, 더 큰 데이터 세트를 압축할 경우 도 3의 d의 컨텍스트 모델과, 도 2의 d에 도시된 바와 같은 전체 확률표를 이용한다.

도 1의 블록도는 제어기(1102)를 부가하여 변형함으로써 본 발명을 수행할 수 있다. 도 5에 도시된 바와 같이, 제어기(1102)의 역할 중 하나는 처리될 비트의 수를 결정하는 것이다. 제어기(1102)는 어느 비트를 마스크할 것인지에 대한 신호를 컨텍스트 모델(1115)에 출력한다. 제어기(1102)는 도시되지는 않았지만, 몇 가지 다른 역할을 수행한다.

도 4에는 바람직한 실시예에 대한 흐름도가 도시되어 있다. 데이터를 압축하기 전에 데이터의 크기가 결정된다(501). 데이터의 크기가 X보다 작으면(503), 소형 컨텍스트 모델이 선택되고(505), 데이터의 크기가 X보다 크고, Y보다 작으면(507), 중형 컨텍스트 모델이 선택되고(509), 데이터의 크기가 Y보다 크고, Z보다 작으면(511), 대형 컨텍스트 모델이 선택되고(513), 데이터의 크기가 Z보다 크면, 전체 컨텍스트 모델이 선택된다(515). 일단, 컨텍스트 모델이 선택되면, 선택된 컨텍스트 모델을 이용하여 전체 데이터를 압축한다(517). 당업자라면 본 발명의 동작 순서를 변경하여도 강조하고자 하는 사상을 마찬가지로 구현할 수 있음을 이해할 것이다. 또한, 위에서는 소형, 중형, 대형 및 전체 컨텍스트 모델에 대해 기술하였지만, 소정의 수의 컨텍스트 모델을 정의할 수 있으며, X, Y, Z, ...에 대한 적절한 값을 선택할 수 있다.

또한, 도 4의 흐름도는 데이터 복원시에도 이용할 수 있다. 그러나, 압축되지 않은 데이터의 크기는 압축 데이터의 크기에는 직접적인 관련이 없다. 따라서, 파일을 원래 크기로 복원하고 데이터를 압축하기 위해 컨텍스트 모델이 사용되는 복원기로 일컬어지는 몇몇 수단은 압축 데이터를 구비해야 한다. 바람직한 실시예에서, 원래 데이터의 높이 및 폭은 압축 데이터의 일부로서 저장되어야 한다. 다른 실시예에서는 압축되지 않은 파일의 크기 또는 이용된 컨텍스트 모델을 나타내는 플래그(flag)를 저장할 수도 있다. 일단, 적절한 컨텍스트 모델이 선택되면, 데이터가 복원된다.

본 발명은 보다 큰 영상의 압축율에 영향을 미치지 않으면서, 폰트 캐시 데이터(즉, 작은 파일들)에 대한 향상된 압축율을 제공한다. 라틴어 문자에 대한 특정한 경우, 폰트 캐시 문자의 압축은 40% 향상된다. 즉, 기존의 산술 코딩 압축 기법에 비해 문자가 40% 더 압축되어 폰트 캐시에 제공된다. 어드레스 비트를 마스크하는 복잡도는 매우 낮으며, 설계시 약간의 논리가 부가될 뿐이다.

비록, 본 발명의 바람직한 실시예가 예시되고, 그러한 형태가 기술되었지만, 당업자라면 본 발명의 사상을 벗어나거나, 첨부된 특허 청구범위의 영역을 벗어나지 않고서도 여러 가지 변형이 가능함을 알 수 있을 것이다.

발명의 효과

본 발명에 따른 데이터 압축/복원 방법 및 압축 장치에 따르면, 처리되는 데이터 세트의 크기에 최적화된 컨텍스트 모델을 선택함으로써, 보다 큰 데이터의 압축율의 영향을 작게하면서 작은 데이터 세트들에 대해 향상된 압축율을 제공한다.

(57) 청구의 범위

청구항 1

원래 데이터를 압축 데이터로 압축하는 방법에 있어서,
상기 원래 데이터의 크기를 결정하는 단계와,
상기 데이터의 크기를 기초로하여 컨텍스트 모델(context model)을 선택하는 단계와,
상기 컨텍스트 모델을 이용하여 상기 원래 데이터를 상기 압축 데이터로 압축하는 단계를 포함하는 데이터 압축 방법.

청구항 2

제 1 항에 있어서,
상기 컨텍스트 모델 선택 단계는 상기 데이터의 크기가 사전결정된 양보다 크면 제 1 컨텍스트 모델을 선택하고, 상기 데이터의 크기가 사전결정된 양보다 작으면 제 2 컨텍스트 모델을 선택하는 데이터 압축 방법.

청구항 3

제 1 항에 있어서,
상기 압축 데이터를 저장하는 단계를 더 포함하는 데이터 압축 방법.

청구항 4

압축 데이터를 원래 데이터로 복원하는 방법에 있어서,
상기 원래 데이터의 크기를 기초로하여 컨텍스트 모델을 선택하는 단계와,
상기 컨텍스트 모델을 이용하여 상기 압축 데이터를 상기 원래 데이터로 복원하는 단계를 포함하는 데이터 복원 방법.

청구항 5

제 4 항에 있어서,
상기 컨텍스트 모델 선택 단계는 상기 데이터의 크기가 사전결정된 양보다 크면 제 1 컨텍스트 모델을 선택하고, 상기 데이터의 크기가 사전결정된 양보다 작으면 제 2 컨텍스트 모델을 선택하는 데이터 복원 방법.

청구항 6

제 4 항에 있어서,
상기 컨텍스트 모델 선택 단계는 상기 압축 데이터가 상기 압축 데이터를 압축하기 위해 상기 제 1 컨텍스트 모델이 이용된 것으로 나타낼 때, 제 1 컨텍스트 모델을 선택하고, 상기 압축 데이터가 상기 압축 데이터를 압축하기 위해 상기 제 2 컨텍스트 모델이 이용된 것으로 나타낼 때, 제 2 컨텍스트 모델을 선택하는 데이터 복원 방법.

청구항 7

제 4 항에 있어서,
상기 컨텍스트 모델 선택 단계는,
상기 압축 데이터로부터 지시자(indicator)를 복원하는 단계와,
상기 식별자를 이용하여 선택된 컨텍스트 모델을 식별하는 단계를 더 포함하는 데이터 복원 방법.

청구항 8

데이터 압축 장치에 있어서,
산술 압축기(arithmetic compressor)를 포함하되,
상기 산술 압축기는,
확률표(probability table)와,

제 1 콘텍스트 모델과,

제 2 콘텍스트 모델과,

상기 데이터를 수신하도록 배열된, 상기 산술 압축기에 접속된 시프트 레지스터와,

상기 산술 압축기에 접속되고, 상기 데이터의 크기가 사전결정된 양보다 작을 때 상기 제 1 콘텍스트 모델을 이용하여 상기 데이터를 압축하고, 대안적으로는 상기 제 2 콘텍스트 모델을 이용하여 상기 데이터를 압축하도록 상기 산술 압축기에 신호를 출력하는 상기 데이터의 크기를 결정하는 수단

을 포함하는 데이터 압축 장치.

청구항 9

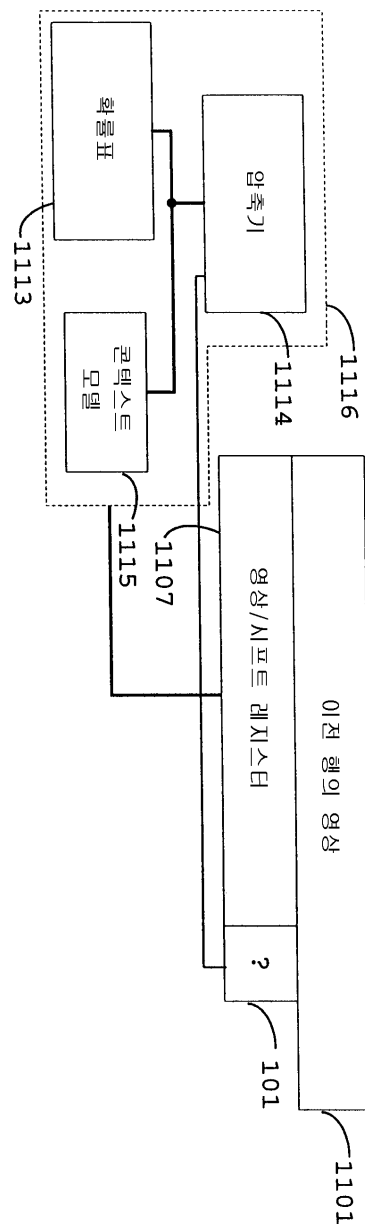
제 8 항에 있어서,

상기 제 1 콘텍스트 모델은 상기 사전결정된 양보다 작은 데이터의 양에 대해 최적화되고,

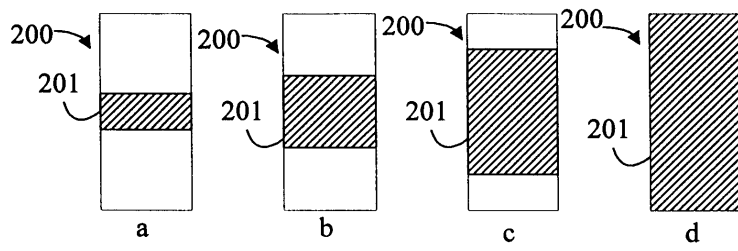
상기 제 2 콘텍스트 모델은 상기 사전결정된 양보다 큰 데이터의 양에 대해 최적화되는 데이터 압축 장치.

도면

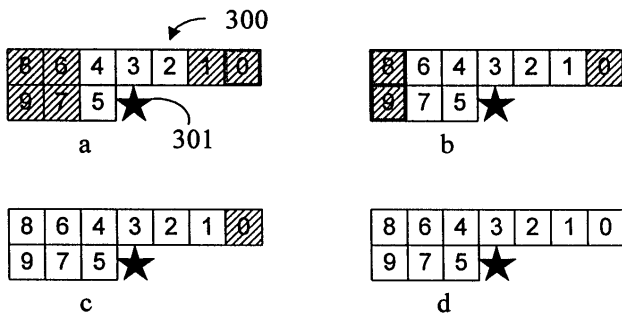
도면1



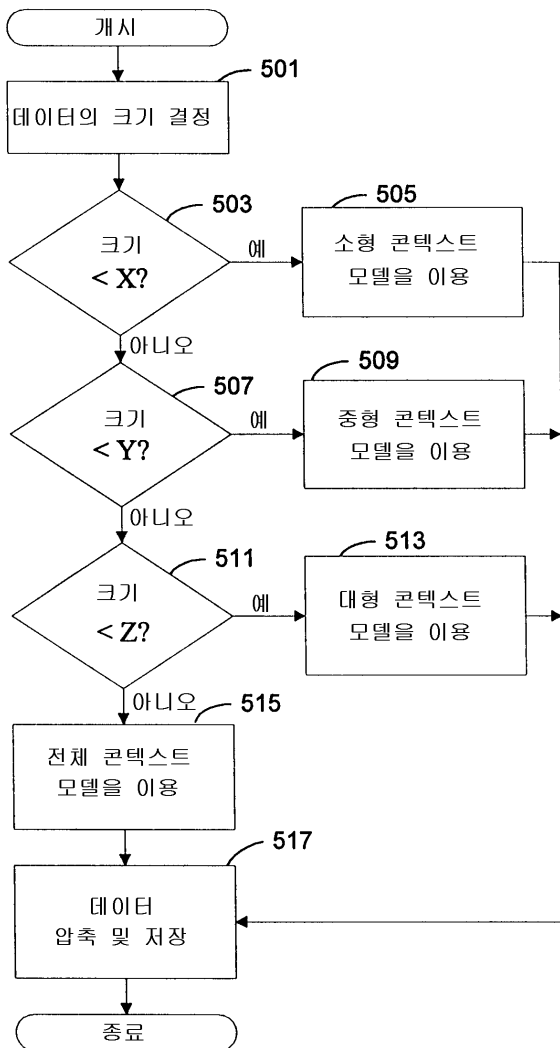
도면2



도면3



도면4



도면5

