

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl.⁷
H03M 7/46

(45) 공고일자 2005년05월25일
(11) 등록번호 10-0491333
(24) 등록일자 2005년05월16일

(21) 출원번호	10-2000-7015124	(65) 공개번호	10-2001-0053348
(22) 출원일자	2000년12월30일	(43) 공개일자	2001년06월25일
번역문 제출일자	2000년12월30일		
(86) 국제출원번호	PCT/US1999/015221	(87) 국제공개번호	WO 2000/02319
국제출원일자	1999년07월02일	국제공개일자	2000년01월13일

(81) 지정국

국내특허 : 아랍에미리트, 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그라나다, 그루지야, 가나, 감비아, 크로아티아, 헝가리, 인도네시아, 이스라엘, 인도, 아이슬란드, 일본, 케냐, 키르기즈스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 폴란드, 포르투갈, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 슬로베니아, 슬로바키아, 시에라리온, 타지키스탄, 투르크멘, 터키, 트리니다드토바고, 우크라이나, 우간다, 미국, 우즈베키스탄, 베트남, 세르비아 앤 몬테네그로, 남아프리카, 짐바브웨,

AP ARIPO특허 : 가나, 감비아, 케냐, 레소토, 말라위, 수단, 시에라리온, 스와질랜드, 우간다, 짐바브웨,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르기즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크멘,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀란드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디부아르, 카메룬, 가봉, 기니, 기니 비사우, 말리, 모리타니, 니제르, 세네갈, 차드, 토고,

(30) 우선권주장 09/109,475 1998년07월02일 미국(US)

(73) 특허권자 인텔 코오퍼레이션
미합중국 캘리포니아 산타클라라 미션 칼리지 블러바드 2200

(72) 발명자 아차리야틴쿠
미국아리조나85283템프사우스로버츠로드7292

카람리나제이.
미국아리조나85044피닉스이스트피에드몬트로드4619

마리노프란체스코마리아
이탈리아바리I-70126비아포스티글리오네26

(74) 대리인 특허법인 신성

심사관 : 나용수

(54) DWT-기반 기술에 의해 압축된 이미지를 코딩하기 위한실시간 알고리즘 및 구조

요약

고정된 길이 코드워드 및 플래그를 런-길이를 인코딩하기 위한 데이터 포맷 구조로 제공하도록 하는 방법이 기술되어 있다. 큰 런-길이는 데이터 셋 내의 두 비-제로 값 사이의 제로의 수를 표현하는 구조의 시퀀스를 사용한다. 인코딩될 데이터의 엔트로피 특성에 적합한 인코딩 방법을 결정하는 적응형 인코딩 기법이 기술되어 있다. 필요에 따라 고정된 길이의 사이즈를 변경할 수 있도록 하는 아키텍처가 기술되어 있다. 이 아키텍처는 각각 요구된 코드워드 사이즈에 따라 인코딩되는 하나의 카운터(410)에 연결된 다수의 로직 네트워크(422,424,426,432,434,436)를 포함한다.

대표도

도 4

색인어

이미지 압축, 데이터 압축, DWT, 양자화, 허프만 코딩, 런 길이

명세서

기술분야

본 발명은 일반적으로 데이터 및 이미지 압축에 관한 것으로서, 특히 본 발명은 적응형 양자화 및 이산 웨이블릿 변환이 이용되는 경우의 데이터 및 이미지 압축을 위한 2진 인코딩에 관한 것이다.

배경기술

이미지 및/또는 데이터 압축에 있어서, 인코딩으로서 알려진 프로세스를 통해, 외부에서 얻어지거나 입력되는 텍스트 또는 수치 데이터와 같은 일련의 값들이 2진 형태(1 또는 0)로 인코딩될 수 있다. 인코딩의 한가지 방법은 (ASCII 수치 지정과 같은) 텍스트를 위한 각각의 10진수 또는 10진 코드를 단순히 일정한 수의 비트(워드 크기)로 변환하는 것이다. 예를 들어, 수 23, 128 및 100은 시퀀스 00010101 1000000 01100100로서 2진수로 인코딩 되게 된다. 이러한 미처리(raw) 또는 순수한 2진 코드는 더 이상의 압축 기능을 제공하지 않는데, 그 이유는 이것이 단지 데이터를 취하여 그것을 2진수로 표현하는 것이기 때문이다. 이러한 인코딩은 제로의 수가 비-제로(non-zero)를 훨씬 능가하는 경우, 특히 이러한 제로 데이터 값이 연속되어, 2진수에서 제로의 큰 "런(run)"을 생성하는 경우에 효율적이지 못하다. 2진 변환 동안에 데이터를 압축하기 위해 특히 디지털 통신 분야에 몇 가지 방법이 개발되었다. 이미지 또는 데이터 압축을 위해 널리 사용되는 이러한 방법은 허프만 코딩(Huffman Coding) 및 런-길이 인코딩(Run-Length Encoding)이다.

고전적인 허프만 코딩은 L_i 비트의 코드워드에 의해 N 개의 가능한 데이터 값의 소정의 데이터 셋 S 내의 각각의 가능한 값 $y_i(i=1,...,N)$ 를 코딩하는 것으로 이루어진 가변 길이 코딩 기술이다. 허프만 코딩의 목적은 $\sum L_i P(y_i)$ 를 최소화하는 것이며, 여기서 $P(y_i)$ 는 인코딩될 데이터 셋 S 에서 발생하는 값 y_i 의 확률이다. 코드워드는 그들을 서로 구별하기 위해 선택된다. 코드워드는 가변 길이를 가지며, 예를 들어, 데이터 셋 $S=\{0,1,2,3,4\}$ 에 대해, 허프만 코딩은 맵핑 $\{0=0, 1=10, 2=110, 3=1110, 4=1111\}$ 을 이용할 수 있다. 만일 $P(0) \gg P(1) \gg P(2) \gg P(3) \gg P(4)$ 이면, 이 기술은 순수 고정 길이 2진 표기법 보다 더욱 효율적일 수 있다. 허프만 코딩은 주로 작은 수 N 을 갖거나 또는 작은 분산(variance)을 가진 데이터 셋 S 를 코딩하는 경우에 효율적이며, 그 이유는 L_i 의 크기가 셋 내의 값들의 수 N 의 증가에 따라 거의 선형으로 증가하기 때문이다. 이러한 이유로 인해, 수정 허프만 코딩으로 알려진 고전적인 허프만 코딩과 다른 기술이 개발되어, 그 데이터 셋 내의 보다 큰 수의 N 또는 보다 많은 분산을 가진 이미지에 이용된다.

수정 허프만 코딩에 있어서, 데이터 값 y_i 는 2개의 필드, 즉 레인지(Range)(2^{Range} 값을 포함하는 셋을 식별함)와 Range 내의 특정 값을 나타내는 포인터(Pointer)를 가진 데이터 구조에 의해 인코딩된다. 필드 Range는 허프만 코딩을 이용하여 코딩되고, Pointer는 Range 비트의 크기를 가진 2진수로서 표현된다. 예를 들어, 1 내지 n 값들의 Range 내에 포함된 값들은 다음과 같이 될 수 있다.

- * Range = 0: 값 {0} (필드 Pointer는 필요로 되지 않음)
- * Range = 1: 값 {-1,1} (필드 Pointer는 특정 값을 나타내기 위해 1 비트를 필요로 함)
- * Range = 2: 값 {-3,-2,2,3} (필드 Pointer는 2 비트를 필요로 함)
- * Range = n: 값 $\{-2^n+1, -2^n+2, ..., -2^{n-1}, 2^{n-1}, ..., 2^n-2, 2^n-1\}$ (필드 Pointer는 n 비트를 필요로 함)

예를 들어, 만일 인코딩할 데이터가 정수 값 $\{-3,-2,-1,0,1,2,3\}$ 이면, 그 각각에 대해 가능한 MHC는 다음의 표1에 도시된 바와 같이 될 수 있다.

표 1.

y_i	Range	Huf. Code for Range	Pointer having Range bits	Complete code (MHC)
-3	2	11	11	<u>11</u> 11
-2	2	11	10	<u>11</u> 10
-1	1	10	1	<u>10</u> 1
0	0	0	---	<u>0</u>
1	1	10	0	<u>10</u> 0
2	2	11	00	<u>11</u> 00
3	2	11	01	<u>11</u> 01

값 0은 하나의 워드 0으로 코딩된다(즉, Range의 허프만 코딩, 다른 비트는 없음). 값 3은 1101으로 코딩되며, 여기서 밑줄친 11은 Range 2의 허프만 코드이고, 그 다음에 오는 2(=Range) 비트, 01은 표1에서 Range=2를 가진 가능한 값들{-3, -2, 2, 3} 사이의 값 3을 지시한다. 만일 P(0)가 하이(high)이면, 전술한 방법은 각각의 값이 그 발생 확률에 관계없이 3 비트에 의해 코딩되게 되는 정상적인 고정 길이 2진 코딩 보다 더욱 효율적이며, 그 이유는 표1의 MHC에서 오직 하나의 비트만이 제로 값을 위해 이용되기 때문이다. MHC는 테이블 탐색 아키텍처(table look-up architecture)를 위해 실물 그대로 설계되며, 따라서 인코딩 및 디코딩 모두에 대해 더욱 효율적일 수 있다.

제로 런 길이 코딩(Zero Run Length Coding)(ZRLC)으로서 알려진 다른 기술은 매우 큰 수의 연속되는(또는 런) 제로 값을 포함하는 데이터 셋트를 인코딩하기 위한 표준 기술이다. ZRLC는 (허프만 코딩 또는 일부 다른 코딩을 이용하여) 제로가 아닌 값들만을 인코딩하고, 코더 및 디코더 모두에게 알려진 방식에 따라 2개의 연속되는 비-제로 값을 분할하는, 제로의 수를 명시하는 코드에 의해 이들 코드워드를 인터리빙하는(interleaving) 것으로 이루어진다.

종래의 ZRLC에서, 인코딩된 제로 런 데이터는 2개의 세그먼트, 즉 런 길이 및 비-제로 값을 이용하여 구성된다. 예를 들어, 데이터 스트림: {0000005000-6780000-1200000000000014500000000023}을 코딩하는 대신에, 단지 다음의 데이터가 코딩된다:

{[6,5][3, -6][0,78][4, -12][12,1][0,45][9,23]}

이 코드(여기서, _는 제로의 런 길이를 나타냄)는 6개의 제로 다음에 값5가 오고, 그 다음 3개의 제로 다음에 값-6이 오고, 그 다음 0개의 제로 다음에 값78이 오는 것 등을 나타낸다.

이러한 기술 중 하나가 이미지 압축에 이용되면, 인코딩 이전의 이미지 압축의 본질(nature)이 방법의 최상의 선택을 지시해야 한다. 허프만 코딩이 이용되어야 하는지 또는 런 길이 코딩이 이용되어야 하는지를 고려하기 위해, 이미지 또는 애플리케이션(예, 화상회의)의 본질이 분석되거나 고려될 필요가 있을 수 있다. 높은 엔트로피(entropy)를 가진 이미지에 대해, 런-길이 인코딩은 허프만 코딩만큼 높은 압축 비율을 제공하지 못할 수도 있으며, 그 역으로도 마찬가지이다. 2진 인코딩 프로세스를 데이터 셋트(또는 데이터 셋트의 서브셋트)의 특성에 일치시킬 수 없다는 것은 최적화되지 않은 전체 압축 비율을 초래할 수 있다.

이미징 기술분야에서, 이미지가 (디지털 카메라와 같은 장치에 의해) 포착되어, 아마도 "컬러 보간(color interpolated)"된(이미지 풀 컬러 해상도를 제공하기 위해 누락된 컬러 성분에 대한 판단이 이루어짐) 이후에, 종종 이미지는 (2진 인코딩 이전에) "압축"되어, 이미지를 표현하는데 필요한 총 비트 수를 감소시키게 된다. "1차적인(primary)" 이미지 압축 및 그 후속의 압축된 데이터의 2진 인코딩은 화상회의, 디지털 이미징 및 네트워크를 통한 비디오 스트리밍과 같은 멀티미디어 애플리케이션에서 핵심적인 역할을 한다. 이러한 애플리케이션을 위한 1차적인 이미지 압축 방식은 특정 애플리케이션에 대해 수용 가능한 이미지 품질을 유지하면서 이미지의 저장 또는 전송의 비트율(bit-rate)을 감소시키도록 설계되어야 한다.

한가지 일반적으로 사용되는 1차적인 이미지 압축 기술은 공지의 이산 코사인 변환(DCT)을 이용하여 입력 이미지의 픽셀을 변환하는 JPEG(Joint Photographic Experts Group)로서 알려져 있다. 이 결과로서 생성되는 변환된 픽셀 값은 압축을 실현하기 위해 값들의 보다 작은 셋트로 양자화 또는 맵핑된다. 압축 해제되는 압축 이미지의 품질은 변환된 픽셀의 양자화가 수행되는 방법에 크게 의존하게 된다. 압축 비율(원래의 미처리 이미지의 크기를 압축된 이미지와 비교한 것)도 역시 양자화에 영향을 받지만, 양자화 이후의 후속 2진 인코딩에 의해 더욱 영향을 받을 수 있다.

JPEG 압축 이미지 데이터의 후속 2진 인코딩은 런 길이 인코딩이 사용되는 경우에 "블로킹(blocking)"에 의해 제한을 받는다. JPEG에 있어서, 이미지는 8x8 또는 16x16 블록과 같은 픽셀 블록으로 분할된다. 이들 블록은 서로에 대해 독립적으로 처리되며, 따라서 가능한 최대 런-길이는 블록의 크기(64 또는 256)이다. 그러므로, 만일 런-길이 인코딩이 사용되면, 런-길이 값은 6-비트 또는 8-비트 폭이 된다. 그러므로, JPEG에 있어서 런-길이 인코딩은 런-길이를 포함하는 비트 수로 일정하게 유지될 수 있다. 블록-기반 코딩이 이용되지 않는 경우에, 각각의 "런(즉, 연속되는 제로의 수)을 코딩하기 위한 이러한 일정한 수의 비트는 심각한 한계가 될 수 있으며, 그 이유는 하나의 런에 대한 가장 긴 값이 전체 이미지의 크기에 달려 있기 때문이다.

높은 압축 비율 및 수용가능한 압축해제 이미지 품질을 실현하는 다른 1차적인 이미지 압축 방식은 JPEG 블록과 달리 일정하지 않고 그 크기가 가변적인 이미지 "서브-밴드" 또는 이미지 주파수 영역을 생성할 수 있으며, 그 이유는 이러한 방식이 이미지를 블록으로 분할하지 않기 때문이다. 이산 웨이블릿 변환(DWT)에 기반한 이러한 한가지 1차적인 이미지 압축 방식이 "The Compression of Color Images Based on a 2-Dimensional DWT"란 제목의 관련 미국특허출원(이후 DWT 특허로 언급됨)에 기재되어 있다. 이러한 DWT-기반 방식에 있어서, 각각의 서브-밴드 및 채널(컬러 평면 또는 컬러 평면의 차)은 특히 높은 엔트로피를 가진 서브-밴드에서 런-길이 인코딩보다는 오히려 허프만 코딩의 이용을 정당화하는 특성을 가질 수 있다.

만일 이미지가 압축되고, 그 다음에 디지털 카메라 또는 다른 이미징 장치 상에서 인코딩될 예정이면, JPEG를 위한 통상적인 런-길이 인코딩은 적당하지 않다. 그러므로, 디코딩이 더욱 실시간으로 이루어질 수 있도록 고정 길이 구조를 유지하면서 가변적이고 예외적으로 큰 런-길이 값이 인코딩될 수 있도록 허용하는 런길이 인코딩 방식이 요구된다. 또한, 1차적인 이미지 압축이 여러 스테이지(예를 들어, 각각의 서브-밴드에 대해 하나씩의 스테이지)에서 데이터를 생성하는 경우, 각 스테이지의 데이터가 다른 것에 비해 한가지 형태의 인코딩을 이용하는 것에 유리한 특성을 가진 경우에, 각 스테이지가 가능한 가장 효율적인 인코딩을 이용하여 처리될 수 있도록 적응형 인코딩 프로세스를 제공할 필요가 있다. 이러한 메커니즘은 인코딩 동안에 얻어지는 압축을 최대화하며, 따라서 이미지 또는 다른 데이터에 요구되는 저장/전송 크기를 감소시키게 된다.

발명의 상세한 설명

발명의 요약

고정된 길이 코드워드 및 플래그를 런-길이를 인코딩하기 위한 데이터 포맷 구조로 제공하도록 하는 방법이 기술되어 있다. 큰 런-길이는 데이터 셋트 내의 두 비-제로 값 사이의 제로 수를 표현하는 구조의 시퀀스를 사용한다. 인코딩될 데이터의 엔트로피 특성에 적합한 인코딩 방법을 결정하는 적응형 인코딩 기법이 기술되어 있다. 필요에 따라 고정된 길이의 사이즈를 변경할 수 있도록 하는 아키텍처가 기술되어 있다. 이 아키텍처는 각각 요구된 코드워드 사이즈에 따라 인코딩되는 하나의 카운터에 연결된 다수의 로직 네트워크를 포함한다.

도면의 간단한 설명

본 발명의 방법 및 장치의 목적, 특징 및 장점은 다음의 설명으로부터 명백해질 것이다.

도1a 내지 도1e는 본 발명의 일실시예에 따른 수정된 제로 런 길이 인코딩을 위한 기본적인 데이터 구조를 도시한 도면.

도2는 본 발명의 일실시예에 따른 이미지 처리 데이터 흐름을 도시한 도면.

도3은 본 발명의 일실시예에 따른 압축 및 인코딩된 이미지의 복구를 도시한 도면.

도4는 본 발명의 일실시예에 따른 수정된 제로 런 길이 인코더를 도시한 도면.

도5는 본 발명의 일실시예에 따른 적응형 인코더의 블록도.

도6은 본 발명의 일실시예에 따른 이미지 처리 장치의 블록도.

도7은 본 발명의 일실시예의 시스템 도면.

실시예

이제, 도면을 참조하여 본 발명의 실시예에 관해 설명한다. 이들 실시예는 본 발명의 양태를 예시하기 위해 제공된 것이며, 본 발명의 범위를 제한하는 것으로 해석되어서는 안 된다. 실시예는 주로 블록도 또는 흐름도를 참조하여 설명된다. 흐름도에 관해서, 흐름도 내의 각 블록은 방법의 단계와 이들 방법 단계를 수행하기 위한 장치 구성요소를 모두 나타낸다. 구현예에 따라, 대응하는 장치 구성요소는 하드웨어, 소프트웨어 또는 이들의 조합으로 구성될 수 있다.

도1a 내지 도1e는 본 발명의 일실시예에 따른 수정된 제로 런 길이 인코딩을 위한 기본적인 데이터 구조를 도시한 도면이다.

JPEG 표준에 있어서는, 8x8 블로킹 방식으로 인해, ZRLC에서의 런의 크기가 63보다 높지 않은 값을 가질 수 있으며, 그러므로 6-비트 워드로 고정될 수 있다. 만일 2-DDWT에 의해 생성되는 완전한 서브-밴드를 고려하는 경우에, 런 값의 크기의 상한(upper limit)은 전체 이미지의 크기에 의존한다. 그러므로, 본 발명의 일실시예는 2개의 비-제로 값 사이의 제로 수로서 정의된 런의 크기를 인코딩하기 위한 유연성 있는(flexible) 구조를 제공한다. 구조 인코딩은 플래그 F와 M 비트의 워드 R로 이루어진다. 도1a는 인코딩된 데이터의 기본 포맷을 도시하고 있다. 이 포맷에서, 이 형태의 하나 또는 그 이상의 구조(structure)(F,R) 다음에 데이터 셋트 내의 비-제로 값의 MHC(Modified Huffman Code)가 뒤따른다. 각각의 ZRLC 구조(F,R)는 1-비트 F 값과 비트 $R_1 \dots R_M$ 을 갖는다. 그러므로, 도1a에서 제1 런 길이는 하나의 구조(F,R)에 의해 인코딩된다. 정의에 의해 런 길이는 비-제로 값을 만날 때 종료된다. 예시적인 비-제로 값은 2개의 비트 MHC_1 및 MHC_2 를 이용하여 인코딩된다. 비-제로 값 이후에, 길이가 $2^M - 2$ 를 초과하는 제2 제로 런이 2개의 (F,R) 구조를 이용하여 인코딩된다. 그 다음 비-제로 값은 MHC_1 (단지 1 비트)을 이용하여 인코딩되고, 그 이후에 제로들의 다른 런 길이가 단 하나의 (F,R) 구조를 이용하여 인코딩된다. 도1b에 도시된 (F,R)=(0,0)의 구조는 그 지점으로부터 검사된 밴드의 끝까지 단지 제로들만이 존재한다는 것을 명시하는 것이다. r개의 제로들의 런(여기서, $r < 2^M - 1$)을 나타내기 위해, (F,R)=(0,1+r)이 사용된다. 도1c에 도시된 (0,1)의 구조는 그 경로에서 다음에 오는 비-제로 값이 이전의 비-제로 값과 인접해 있다는 것, 즉 R-1개의 제로들의 런이 2개의 연속되는 비-제로 값 사이에서 발생한다는 것을 나타낸다. 따라서, 도1d에 도시된 바와 같이 4개의 제로들의 런이 구조(0,5)를 이용하여 인코딩된다.

만일 r (런의 총 크기)이 2^M-2 보다 길면, $\log_2(r-2)M$ 구조를 이용하며(여기서, y 는 y 보다 큰 가장 낮은 정수로 y 를 라운딩하는 것을 나타냄), 각각의 구조는 $F=0$ 인 마지막 구조를 제외하고, M 개의 비트를 저장하며 $F=1$ 을 갖는다. 2^M-2 보다 더 긴 r 이 도1e에 도시되어 있다. 런 길이 r 이 2^M-2 보다 더 크면, r 의 최상위 M 비트를 나타내기 위해 플래그 $F=0$ 가 사용된다. 도1e는 런 길이에 85개의 제로가 있다는 사실을 인코딩하기 위한 구조를 도시하고 있다. (F,R) 구조에서 워드 R 의 크기인 M 은 단지 4비트 길이이기 때문에, 단지 14의 런 길이만이 플래그 0에 의해 인코딩될 수 있다. 그러므로, 런 길이 도면의 최상위 부분을 플래그 $F=0$ 가 사용된다. 수 $86(1+r)$ 은 r 인 런 길이에 85개의 제로가 있다는 사실을 인코딩하게 된다. 86_{10} 은 2진수로 0101 1100이고, 이 런 길이의 제1 M 비트($M=4$)는 0101이다. 그러므로, 이 제1 부분은 $(F,R)=(0)$ 또는 0 0101로서 인코딩된다. 인코딩될 값의 제2 및 마지막 부분 1100은 그것이 이전에 인코딩된 값 "0101"에 속한다는 것을 나타내는 $F=1$ 의 플래그를 수반한다. 250의 런 길이는 $(F,R)=(0,F_{16});(1,B_{16})$ 에 의해 인코딩되며, 여기서 $F_{16}=1111$ 이고, $B_{16}=1011_b$ 이며, 그 이유는 $(1111\ 1011)_b=251_{10}$ 이고, 이것은 250의 런 길이인 r 보다 더 크기 때문이다. 이와 유사하게, $r=4091$ 의 런 길이는 구조 $(0,F_{16});(1,F_{16});(1,C_{16})$ 에 의해 인코딩될 수 있으며, 여기서 $(FFC)_{16}=4092_{10}$ 이다. 도1e에 도시된 바와 같이 출력의 순서는 좌측에서 우측으로 런 크기의 최하위 비트($F=1$ 의 구조를 가짐)로부터 최상위 비트($F=0$ 의 구조를 가짐)로 이루어진다. 예를 들어, 도1e는 최상위 비트가 런 길이 85를 인코딩하는데 있어 마지막으로(플래그 $F=0$ 를 가짐) 출력된다는 것을 보여준다. 마찬가지로, 런 길이 250은 $(1,B_{16})(0,F_{16})$ 의 순서로 출력되고, 런 길이 4091은 $(1,C_{16})(1,F_{16})(0,F_{16})$ 의 순서로 출력되게 된다. 마찬가지로, 출력 순서 $(1,R_0)(1,R_1)(1,R_{I-2})(0,R_{I-1})$ 로 된 어떤 수의 구조도 사용될 수 있으며, 여기서 $I \times M$ 개의 비트가 $r+1$ 을 인코딩하는데 필요한 총 비트수이다. 이러한 시나리오에서, $I-1$ 구조는 1의 플래그를 갖고, I 번째 구조는 0의 플래그를 갖는다. 본 발명의 일실시예에서 수정된 ZRLC의 한가지 장점은 런의 크기가 제한되지 않으며, 여전히 단지 M 개의 비트를 가진 R 의 고정 길이 구조(F,R)를 이용하여 인코딩 가능하다는 것이다.

각각의 런 다음에 오는 비-제로 값을 코딩하기 위해, 다음의 가능한 *Range* 값을 가진 MHC 기술을 선택한다.

* $0=\{1\}$

* $1=\{-1,2\}$

* $2=\{-3,-2,3,4\}$

* $n=\{-2^n+1,-2^n+2,\dots,-2^{n-1},2^{n-1}+1,\dots,2^n-1,2^n\}$

이러한 값에 대한 LUT의 예가 다음의 표2에 도시되어 있다. $y=0$ 의 경우는 이러한 MHC에 의해 포함되지 않으며, 그 이유는 ZRLC는 데이터 셋트에서 모든 제로 값의 발생을 처리하기 때문이다.

표 2.

Address (=y)	Non Zero Value	
	size	code
-3	11 (=3)	1111
-2	11 (=3)	1110
-1	10 (=2)	X101
1	00 (=0)	XXX0
2	10 (=2)	X100
3	11 (=3)	1100
4	11 (=3)	1101

이 예에서, 탐색 테이블(look-up table)(LUT)은 각각의 셀이 y (즉, 인코딩될 데이터)에 의해 어드레스되고 6 비트를 포함하도록 구성된다. 처음의 2 비트(필드 "size")는 판독되어, 단지 필드 "code"의 "size"+1 비트만이 고려되어 유효 정보로서 저장된다. 이들 "size"+1 비트 중에서, 최상위의 밑줄친 비트는 허프만 코딩을 이용하여 코딩된 필드 "range"이며(본 예에서 range 0,1,2는 각각 0,10,11로 코딩된다), 최하위 비트는 필드 "pointer"를 구성한다. 예를 들어, -2는 워드 "1110"로 코딩된다(즉, "range"=11이고, "pointer"=10이다).

허프만 코딩의 효율적인 구현은 본 발명의 일실시예에서 ZRLC 내의 ql-제로 값에 대한 MHC를 코딩하는 LUT에 대한 소정의 통계적인 분석을 필요로 하기 때문에, 예시적인 이미지의 분석에 기반하여 이러한 코드의 최적의 셋트를 결정할 수 있다. 부록 A-E는 DWT 특허에 사용된 방식의 이미지 압축을 받는 이들 이미지의 결과로서 각 채널의 각 서브-밴드에 대해 50개 이상의 상이한 이미지를 실험적으로 고려함으로써 이루어지는 이러한 코드를 보여주고 있다. 분석에 이용된 이미지를 변환하기 위해 사용되는 DWT 필터는 DWT 특허에 이용된 DWT와 동일한 9-7개의 바이오포스플라인 필터(biorthogonal splines filters)이다.

도2는 본 발명의 일실시예에 따른 이미지 처리 데이터 흐름을 도시하고 있다.

스틸 정지화상 또는 동화상 촬영과 같은 디지털 애플리케이션에서는, 디지털 카메라에 의해 포착되는 것과 같은 원래의 이미지가 압축해제 및/또는 디스플레이를 위해 전송되기 이전에 소정의 품질 레벨을 유지하면서 가능한 한 많이 크기가 압

축되는 것이 바람직하다. DWT 특허에 기재된 1차적인 이미지 압축 기술은 이미지 품질을 유지하기 위해 컬러 및 광에 대한 사람의 시각 체계의 응답을 적응방식으로 이용하도록 특별히 개발되었다. 이러한 압축 이후에, 본 발명의 일실시예에 따르면 데이터를 더욱 압축하기 위해 적응형 2진 인코딩이 적용된다.

DWT 특허에서 언급된 바와 같이, 디지털 카메라 또는 다른 유사한 장치에 의해 포착되는 미처리 이미지는 통상적으로 베이어 패턴(Bayer pattern)으로 표현되게 된다. 센서 어레이(200)는 각각의 위치에서 이미징되는 환경/장면으로부터 센서에 입사되는 광의 세기 값을 포함하는 픽셀 위치 셋트이다. 베이어 패턴에서, 미처리 이미지(200)와 같은 이미지의 각각의 픽셀 위치는 컬러 평면, 적색(R), 녹색(G) 또는 청색(B)과 관련되어 있다. 베이어 패턴은 모든 R과 B에 대해 2개의 관련된 값을 갖기 때문에, 녹색 컬러 평면은 2개의 평면 G1 및 G2로서 간주될 수 있다.

DWT 특허에 기재된 이미지 압축 기술에 따르면, 컬러 평면 R, G1, G2 및 B 사이의 상관성(correlation)은 4개의 채널이 생성되어 분리적으로 압축되도록 하기 위해 활용된다. G1 및 G2 관련 픽셀은 직접 압축 블록(212, 216)으로 전달되며 2개의 채널을 포함한다. R 및 B 픽셀은 덜 직접적으로 처리된다. R 픽셀 값은 그 서쪽의 이웃하는 G1 픽셀 값에 의해 감소된다(블록 205). 이러한 차(R-G1) 채널은 압축 블록(210)으로 전달된다. 마찬가지로, 각각의 B 관련 픽셀은 그 동쪽의 이웃하는 G2 관련 픽셀에 의해 감소된다(블록 206). 이 차(B-G2) 채널은 압축 블록(216)으로 전달된다.

이들 차 채널은 G1 및 G2 순수 컬러 채널과 함께 적절한 압축 스테이지(양자화 및 DWT)로 각각 전달된다.

4개의 채널 R-G1, G1, G2 및 B-G2는 각각 압축 블록(210, 212, 214, 216)으로 전달된다. 이들 각각의 압축 블록(210, 212, 214, 216)에서는, DWT 특허에 따라 2개의 프로세스가 이루어진다. 제1 프로세스는 2차원 이산 웨이블릿 변환(2-D DWT)이다. 2-D DWT는 DWT 특허에 도시되고 기재된 바와 같이 이미지의 "서브-밴드"를 생성한다. DWT 동안에 또는 그 이후에, 양자화로서 알려진 제2 프로세스가 수행된다.

양자화는 n 개의 가능한 값의 셋트를 m 개의 가능한 값의 셋트로 맵핑하는 절차이며, 여기서 $m < n$ 이다. 양자화에 의해, DWT 이미지 데이터 셋트에 대한 가능한 데이터 값의 총수가 감소된다. 적응형 블록(220, 222, 224, 226)의 인코딩은 압축 블록(210, 212, 214, 216)으로부터의 서브-밴드 데이터를 각각 배열(패킹)하여, 그것이 효율적인 2진 표기를 갖도록 한다. 본 발명의 일실시예에 따르면, 적응형 인코딩 블록(220, 222, 224, 226)은 압축 프로세스에 의해 생성되는 각각의 서브-밴드 및 채널 조합에 대한 최적의 인코딩 방법을 이용한다. 본 발명의 실시예에서 정의된 바와 같은 수정된 ZRLC는 잘 알려진 순수한 수정 허프만 코딩(MHC)과 함께 이용된다. 압축된 어떤 채널 및 서브-밴드가 인코딩 되는가에 따라, 이들 2개의 방법 중 하나가 이용되게 된다. 또한, 수정된 ZRLC는 구조(F, R) 내의 R에 대한 고정된 수의 비트 M 을 이용하기 때문에, 수 M 은 인코딩되는 서브-밴드 및 채널에 따라 변화될 수 있다. 예를 들어, 인코딩에 대한 높은 엔트로피를 가진 서브-밴드에서는 보다 작은 값의 M 이 선택될 수 있다. 다음의 표 3은 DWT 특허에 기재된 바와 같이, 9-7 바이오소고널 스플라인 DWT 필터를 이용하여 베이어 패턴(미처리) 이미지에 대해 수행되는 DWT와 지각적 무손실 스레스홀딩(perceptually lossless thresholding)을 이용하여 최종 DWT 계수를 양자화하는 경우에 소정의 서브-밴드 및 채널에 대해 실험적으로 도달한 최적의 인코딩 전략을 보여주고 있다.

표 3.

DWT 서브-밴드

Channels										
LH ₁ , HL ₁	HH ₁	LH ₂ , HL ₂	HH ₂	LH ₃ , HL ₃	HH ₃	LL ₁	LH ₄ , HL ₄	HH ₄	LL ₂	LL ₃
G, G,	ZRLC ₁	ZRLC ₂	MHC	ZRLC ₃	MHC	MHC	MHC	MHC	MHC	MHC
R-G	ZRLC ₁	NU	ZRLC ₂	ZRLC ₃	ZRLC ₄	MHC	MHC	MHC	MHC	MHC
B-G	ZRLC ₁	NU	ZRLC ₂	ZRLC ₃	ZRLC ₄	MHC	ZRLC ₅	ZRLC ₆	ZRLC ₇	MHC

(NU = not used)

전술한 표 3에서, MHC는 공지된 수정 허프만 코딩을 말하고, ZRLC_M은 도 1a 내지 도 1e 및 도 4에 도시된 바와 같은 수정된 ZRLC를 말하며, 여기서 M 은 런 길이를 인코딩하는 구조 내의 비트수이다. 모든 경우에, ZRLC_M은 MHC를 이용하여 비-제로 값을 인코딩한다. 서브-밴드 LH_K, HL_K, HH_K, LL_K는 DWT의 각 레벨 또는 반복인수 K 에서 생성되는 다양한 서브-밴드를 말한다(DWT 특허 참조).

각각의 서브-밴드 및 채널은 그 특성에 따라 인코딩된다. 예를 들어, 채널 G1 및 G2에 대한 HH₂ 서브-밴드는 $M=2$ 인 ZRLC를 이용하고, LH₂ 및 HL₂ 서브-밴드는 순수 MHC를 이용한다. 도 2는 DWT 특허에서 제공된 바와 같이 이용된 이미지 압축 기술의 예를 도시하고 있으며, 여기서 상이한 특성을 가진 스테이지 또는 서브-영역에서 데이터를 생성하는 이미지 압축 기술은 인코딩이 서로에 대해 최적화되도록 적응형 인코딩을 이용할 수 있다. 압축 및 인코딩된 이미지 데이터(240)는 소정의 매체에 저장되거나, 한 시스템으로부터 다른 시스템으로 전송되거나 또는 네트워크와 같은 통신 경로를 통해 분배될 수 있다. 또한, 압축 및 인코딩된 이미지 데이터는 수집되어 단일 프레임으로 전송될 필요가 없으며, 하나의 인코딩된 값마다 그 목적지로 스트리밍될 수 있다.

DWT 변환, 양자화 및 인코딩을 위해 사용되는 정확한 파라미터에 따라, 원래의 이미지의 크기를 압축된 이미지의 크기로 나눈 압축 비율이 변화되게 된다. 본 발명의 이 실시예는 DWT 특허에 언급된 압축 프로세스에 의해 생성되는 각각의 채널 및 서브-밴드에서 압축 비율을 최적화하도록 작용할 수 있는 적응형 인코딩 프로세스를 제공한다.

도 1a-1e, 도 4 및 도 5에 기재된 본 발명의 한가지 중요한 양태는 수정된 제로 런 길이 인코딩 방식이다. 수정된 ZRLC는 어떠한 이미지 또는 데이터 압축 프로세스에도 이용될 수 있으며, DWT 특허로부터의 압축 프로세스의 이용은 단지 예시적인 것이다. 마찬가지로, 서브-밴드 특성 및 채널 특성에 잘 반응하는 것으로 도시된 본 발명의 다른 실시예에 제공된 적응형 인코딩 프로세스도 역시 이미지 압축을 발생하는 다른 형태의 스테이지 또는 서브-영역에 잘 반응하도록 수정될 수 있다.

도3은 본 발명의 일실시예에 따른 압축 및 인코딩된 이미지의 복구(recovery)를 도시하고 있다.

디코딩 블록, 역양자화 블록 및 역DWT 블록이 압축 및 인코딩된 이미지 데이터(240)로부터 런 이미지(200)를 복구하려고 시도하는 프로세스를 구성한다(도2 참조).

압축 및 인코딩된 하나의 채널마다 그리고 하나의 서브-밴드마다 효율적으로 저장될 수 있다. 그러므로, 압축 및 인코딩된 채널 (R-G1), G1, G2 및 (B-G2)는 독립적으로 디코딩되고 및 압축해제 될 수 있다. 먼저, 각 채널 및 서브-밴드에 속한 데이터가 디코딩된다(블록 310, 312, 314, 316). 도2에 도시된 적응형 인코딩(블록 220, 222, 224, 226)은 소정의 서브-밴드 및 채널의 데이터에 적용된 인코딩의 형태에 적절한 디코딩이 이용될 것을 요구한다. 데이터의 각 채널 및 서브-밴드는 다른 서브-밴드와는 상이한 기술을 이용하여 인코딩되고, 이에 따라, 채널은 인코딩 기술에서의 차를 고려하여 디코딩될 필요가 있다. 그리고 나서, 디코딩된 데이터는 압축해제된다(블록 320, 322, 324, 326). 도2에 도시된 압축에 따라, 압축해제는 두 절차 - 디코딩된 데이터를 양자화하고, 역 DWT(IDWT)를 수행함 - 로 구성된다. 이 압축해제 단계는 DWT 특허에서 보다 자세히 논의된다.

압축해제가 완료되면, 미처리 이미지(200)의 근사치(approximation)가 픽셀마다 획득되어 진다. G1을 (R-G1) 복구 값에 다시 가산하고(블록 325), G2를 (B-G2) 복구 값에 가산함(블록 326)으로써, 원래의 미처리 데이터(200)로부터의 각 배이어 패턴 픽셀 값(R, G1, G2)이 근사하게 복구될 수 있다.

적응형 디코딩은 하드웨어, 소프트웨어, 또는 이들의 하나 또는 조합으로 구현될 수 있고, 인코딩 압축 프로세스의 기능을 수행하는 장치로부터 물리적으로 분리될 수 있다. 압축 및 재생으로 구성된 이미지 프로세싱을 위한 기본적인 데이터 흐름은 또한 압축 블록으로부터 요구된 재생 능력을 필요로 하는 요구된 시스템 또는 프로세스로의 중간 변환을 포함할 수 있다.

도4는 본 발명의 일실시예에 따른 수정된 제로 런 길이 인코더를 도시하고 있다.

워드 R에 대한 M비트의 수가 3인 수정된 제로 런 길이 인코더는, 도4에 도시된 바와 같이, 카운터(410) 및 로직 네트워크로 구현될 수 있다. 카운터(410)는 이진수로 런의 총 길이를 결정하는데 사용된다. 이 예에서 12비트로 도시된 카운터(410)는 시스템에서 요구된 어떤 수의 비트도 될 수 있다. 본 발명의 일실시예에 따라, k-비트 카운터는 최대 런 길이 값 $run_max = 2^k - 2$ (k는 run_max 가 최대 전체 서브-밴드 또는 이미지의 치수보다 작지 않도록 선택됨)을 인코딩할 수 있다. 데이터 스트림에서 만난 연속되는 각 "제로"는 카운터를 1만큼 증가시키고, 비-제로 값을 만나면, 카운터(410)는 리셋(000000000001로)되고, 다음의 제로 런을 기다린다. 000101000010 또는 322 제로의 런 길이(r)가 인코딩되는 상태에서의 카운터(410)가 도4에 도시되어 있다. 이 카운터(410)는 $1+r$ 또는 000101000011 값을 유지한다. 이 런 길이는 데이터 구조(F_i, R_i)로 나타낼 필요가 있고, R_i 의 길이 M이 3이기 때문에, 카운터(410) 내의 값은 4, R_0, R_1, R_2, R_3 의 3-비트 세그먼트로 간주된다. 또한, 이것은 모두 4개의 가용 데이터 구조가 출력될 필요가 없는 경우일 수 있다. 도4의 예에서의 경우에는, 000101000011 값은 출력될 필요가 없는 3개의 유도(leading) 제로를 갖는다. 따라서, $En_1=1$ 일 때에는 데이터 구조(F_i, R_i)가 출력되고, $En_1=0$ 일 때에는 그렇지 않도록 하기 위해, 각 데이터 구조(F_i, R_i)에 대한 인에이블 신호 En_i 가 생성된다. 도4는 비-제로 값(MHC에 의한 경우로 인코딩된)을 만나면 (F_0, R_0)가 항상 출력되도록 하기 위해, STOP 및 RESET이 표명되면 En_0 는 항상 인에이블됨을 보여준다.

(F_{i-1}, R_{i-1})가 (F_i, R_i)를 앞서는 것과 같은 순서로 데이터 구조가 출력된다. 이에 따라, 도4의 예에서는, 데이터 구조(F_0, R_0) 뒤에 (F_1, R_1)가 뒤따르고, 그 다음에 (F_2, R_2)가 뒤따른다. 본 발명의 일실시예에 따라, 데이터 구조($1, R_1$) 내의 각 필드 R_i 는 구조($0, R_i$)가 나타날 때까지 단일 런을 구성하도록 사용된다. 이러한 구조에 대한 플래그 $F=0$ 는, R_i 의 비트가 구조($1, R_i$)로 코딩된, 즉 $F=1$ 을 갖는 이전에 만난 워드 R_i 의 나머지 모든 비트에 의해 구성되는 최상위 2진수라는 것을 나타낸다. 구조에 대한 제로 플래그는 런 사이즈의 모든 비트를 만났음을 나타낸다. 따라서, 한 쌍의 4-입력 OR 게이트(422, 424)는 연속되는 구조의 OR 게이트 결과의 입력으로 제공된다. 따라서, OR 게이트(422)에 대한 하나의 입력은 OR 게이트(424)의 출력이 되고, OR 게이트(424)에 대한 하나의 입력은 3-입력 OR 게이트(426)의 출력이 된다. 나머지 OR 게이트(422)의 3개의 입력은 R_i 의 비트인 반면, OR 게이트(424)의 3개의 입력은 R_2 의 비트이다. R_3 의 3비트는 OR 게이트(426)의 입력을 형성한다. 이 캐스캐이딩(cascading) OR 게이트 구조는 런 길이를 표현하는데 필요한 많은 구조를 정확하게 출력 및 플래그하는 능력을 구현한다.

만일, 인에이블 신호 En_i 의 상태가 1이라면, 구조 (F_i, R_i)가 출력된다. 2-입력 AND 게이트는 En_0 를 제외한 각 인에이블 신호를 제어한다. En_0 는 "STOP & RESET" 신호(440)(즉, 비-제로 값을 만나면 1로 설정되고, 이에 따라 "STOP & RESET" 신호(440)가 표명된)에 의해 직접 제어된다. En_1 은 AND 게이트(432)의 출력이고, 반면 En_2 는 AND 게이트(434)의 출력이며, En_3 는 AND 게이트(436)의 출력이다. AND 게이트는 하나의 입력으로 M_3 신호(이것은 En_0)를 갖고, 다른 입력으로 다음의 구조에 속하는 플래그 F를 갖는다. 플래그 F가 제로일 때, 런 길이수의 끝이 나타난다. 예를 들어, 도4의 예에서, $F_2=0$ 이고, 따라서 En_3 가 제로로 설정되어, (F_3, R_3)이 출력될 필요가 없음을 나타낸다. 도시된 예에서, "STOP & RESET"이 표명되면, En_0, En_1, En_2 가 모두 1이고, 구조 (F_2, R_2), (F_1, R_1) 및 (F_0, R_0)을 출력할 필요성에 부합한다. 심지어 R_i 가 000이더라도, 구조(F_1, R_1)는, 이것이 런 길이 수(101000010(101000011로 인코딩되는)의 일부를 형성하기 때문에 출력된다. 반면에, (F_3, R_3)을 구성하는 3 개의 유도 제로는, 런 길이 값을 나타내는 것이 필수적이지 않기 때문에, 출력되지 않는다.

플래그 F_i 는 OR 게이트의 출력을 고려하여 도달된다. 플래그 F_3 은 필요에 의해 12-비트 카운터의 주어진 예를 위한 런 길이 값의 최상위 부분을 형성하기 때문에, 항상 제로와 같아야 한다. 런 길이가 2^M-2 와 같거나 또는 작은 경우, 본 발명의

일실시예에 따라, 단기 (F_0, R_0)가 출력될 필요가 있을 것이다. 이 경우에, 플래그 F_0 은, 도1d의 데이터 구조 정의에 따라, 0이 되어야 한다. 2^M-2 이하의 런 길이는 카운터가 카운터(410)의 R_1, R_2, R_3 부분에 비트가 채워지지 않았음을 나타낸다. 캐스캐이딩 OR 게이트 구조에 따라, OR 게이트(426, 424, 422)는 그의 출력에서 모두 0을 나타낼 것이다. 이에 따라, F_0 은 0으로 설정되어야 한다. OR 게이트(426, 424, 422)가 출력으로 0을 나타냄으로써, 런 길이가 2^M-1 보다 작은 경우, En_1, En_2, En_3 는 또한 제로이고, 이에 따라 구조(F_1, R_1)(F_2, R_2) 및 (F_3, R_3)는 디스에이블되어 출력되지 않는다. 모든 경우에서 $En_0=1$ 이기 때문에, 구조 (F_0, R_0)는 출력되고, 0내지 2^M-2 레인지 내의 어떠한 런 길이라도 충분히 나타낼 수 있다. ZRLC 인코더는 M의 선택 및 카운터의 요구된 총 레인지에 의존하여 수정될 수 있다.

도5는 본 발명의 일실시예에 따른 적응형 인코더의 블록도를 도시하고 있다.

DWT 특허에서 기술한 바와 같이, DWT의 각 레벨 k에서, 서브-밴드 LL_k (k=1에 대해, 시작 데이터 셋트는 이미지 자신임)로부터 시작되어, LL_k, LH_k, HL_k 및 HH_k 서브-밴드가 생성된다. DWT 특허에 따라, k+1레벨의 나머지 3개의 서브-밴드를 생성하기 위하여, LL_k 서브-밴드가 변환된다. 그리고 나서, 서브-밴드 LH_k, HL_k 및 HH_k 가 양자화 및 코딩된다. DWT 해상도의 이전 레벨이 획득될 때까지, DWT가 서브-밴드 LL 위에 "피라미드(pyramidal)" 방식으로 반복되면, 최종 레벨 LL 서브-밴드가 양자화 및 코딩된다. 나머지 LL_k 서브-밴드는 다음 레벨의 서브-밴드로 추가로 변환될 가치로 표현되기 때문에, 양자화 및 코딩을 필요로 하지 않는다.

최종 레벨 LL 서브-밴드 외의 서브-밴드의 인코딩은 다음과 같이 수행된다. 신호 Encoding_Strategy는 일부 바람직한 매칭(상기 표3에 도시된 바와 같은)에 따라, 입력되는 서브-밴드가 본 발명의 다양한 실시예에 따라 나타난 수정된 ZRLC를 이용하여 인코딩 되어야 할지, 아니면 순수한 MHC를 이용하여 인코딩 될지를 나타내게 된다. Encoding_Strategy는 두 로직 경로 중 하나로 순서 제어를 거치는 판단 블록(520)을 거치게 된다. 만일, Encoding_Strategy가 순수한 MHC를 가리키면, 이 판단 블록(520)은 MHC(530)에 대한 LUT(룩업 테이블)에 대해 순서 제어를 거친다. 인코딩될 데이터의 서브-밴드, 레벨 및/또는 채널에 기반하여, LUT(530) 중 하나가 선택되어, 서브-밴드 데이터가 MHC에 따라 인코딩될 것이다. LUT(530)은 임의의 입력값에 대한 맵핑 또는 허프만 코드를 포함하는 일련의 테이블이다. 임의의 입력 서브-밴드 데이터에 대해 LUT로부터 획득된 허프만 코드값은 인코딩된 데이터(570)가 된다.

만일, 판단 블록(520)을 통과한 Encoding_Strategy가 ZRLC를 사용하도록 지시하면, 순서 제어는 블록(525)으로 지나간다.

런 길이를 인코딩하는 수정된 ZRLCM(여기서, M은 구조(F,R)에서의 R에 대한 비트 수임)는 DWT 블록으로부터 서브-밴드 데이터 값 "y"를 제로와 비교함으로써 수행된다. 만일, 이 테스트가 옳으면, 런 길이가 인코더(560)의 카운터는 값 r을 런 길이로 증가시킬 것이다. 비-제로 y(서브-밴드값)가 나타나면, 이 값은 상기 비-제로 값에 대한 MHC 맵핑을 포함하는 ZRLC에 대한 LUT의 어드레스로 사용된다.

압축/인코딩된 데이터(570)를 형성하도록 출력되는 LUT(540)로부터 검색된 비-제로 값에 대한 MHC 이전에, 런 길이가 인코더(560) 내의 카운터가 정지되고, M+1 비트의 구조(F,R)가 압축/인코딩된 데이터(570)에 대한 인코딩된 런 길이가 되도록 출력된다. 그리고 나서, 런 길이가 인코더(560) 내의 카운터는 값1로 리셋되고, 이와 동시에, MHC가 런 길이가 인코더(560)에 의해 구조(F,R) 출력을 따르도록 하기 위해, LUT로부터 나타난 비-제로 값의 MHC는 압축/인코딩된 데이터(570)로 출력된다. 런 길이가 인코더(560)는 또한 3개의 다른 신호, 즉 Reset, M_2 및 M_3 신호에 의해 제어된다. Reset 신호는 신규 서브-밴드의 시작 데이터를 나타낸다. M_2 신호는 구조(F,R) 내의 워드 R의 사이즈가 2, 즉 M=2일 경우에 ZRLC가 수행되도록 지시한다. M_2 가 표명되면, M=2에 대응하는 로직 네트워크는 이 런 카운터로 동작하도록 인에이블 된다. M_3 가 표명되면, 도4에 도시된 바와 같은, M=3에 대응되는 로직 네트워크는 이 런 카운터로 동작하도록 인에이블된다. 이에 따라, 본 발명의 일실시예에서, 인코더(560)는, 모두 동일한 카운터를 여전히 이용하는 동안, 상이한 M 값에 대해 각각 적합한 다수의 로직 네트워크와 함께 장착될 수 있다.

마지막으로, 전체 서브-밴드가 스캔되었을 때, 만일, 이전의 y가 제로라면, 단지 M+1개의 제로가 터미널 워드 (F,R)=(0,0)를 나타내도록 출력된다.

서브-밴드 LFINAL의 코딩은 상이한 데이터 순서를 따른다. 높은 압축을 달성하기 위해, 양자화 및 에러 인코딩(블록 510)이 포함된 간단한 예고적(predictive) 코딩이 L1 서브-밴드에 적용된다. LL 서브-밴드에서 양자화된 계수를 코딩하는 대신에, 인접한 또는 연속한 값 사이의 에러 값이 엔트로피 코딩 구조에 의해 인코딩된다. 이 값들은 순수한 MHC를 이용하여 코딩되고, 이에 따라, MHC(530)에 대한 LUT를 이용한다. 수정된 ZRLC의 완전한 셋트는 원래의 미처리 이미지의 압축된 버전을 나타낸 MHC 인코딩된 데이터이다. DWT 특허에서 언급된 각 "채널"은, 부록A-E에 도시된 바와 같은 MHC 값을 포함하는 적합한 LUT(530 또는 540)를 액세스하여, 도5의 로직에 의해 개별적으로 고려되고 인코딩된다.

도6은 본 발명의 실시예에 따른 이미지 프로세싱 장치의 블록도를 도시하고 있다.

도6은 적응형 인코더를 포함하는 본 발명의 적어도 일실시예와 결합되는 이미징 장치의 내부 이미지 프로세싱 소자의 블록도를 도시하고 있다. 도6의 예시적 회로에서, 센서(600)는 장면/환경으로부터의 컬러/세기 값인 픽셀 요소를 생성한다. 센서(600)에 의해 생성된 n-비트 픽셀 값은 캡처 인터페이스(610)로 전송된다. 본 발명에 관련된 관점에서의 센서(600)는 통상적으로 영역 또는 위치 중 하나의 "장면(scene)"으로부터의 R,G,B 중 하나를 감지하게 된다. 이에 따라, 각 픽셀의 세기 값은 3개(또는 G1,G2가 개별적으로 고려된다면 4개인)의 컬러 평면 중 단지 하나에 연관되고, 베이퍼 패턴 미처리 이미지와 함께 형성된다. 캡처 인터페이스(610)는 센서에 의해 생성된 이미지를 분석하여, 개별 픽셀에 대한 세기 값을 할당한다. 전체 이미지에 대한 이러한 모든 픽셀의 셋트는 디지털 카메라 센서의 통상의 산업적 구현에 따라 베이퍼 패턴 내에 있다.

어떤 센서 장치에서는 센서 평면 내의 일부 픽셀 셀이 장면/환경에서의 알맞은 조명 조건(lightning condition)에 대해 응답하지 않는 것이 전형적이다. 그 결과, 이들 셀로부터 생성된 픽셀 값은 결점이 있을 수 있다. 이 픽셀을 "데드 픽셀(dead pixel)"이라 부른다. "픽셀 대체(pixel substitution)" 유닛(615)은 로우(row) 상에서 각 데드 픽셀을 바로 직전의 유효 픽셀로 대체한다. RAM(616)은 데드 픽셀의 로우 및 칼럼 인덱스로 구성되고, 이것은 센서에 의해 제공된다. 이 RAM(616)은 포착된 이미지에 대한 데드 픽셀의 위치를 식별하도록 돕는다. 압축 모듈(companding module)(625)은 센서로부터 포착된 n-비트(통상적으로 n=10) 세기의 원래의 각 픽셀은 m-비트 세기 값으로 변환하도록 설계되는데, 여기서 $m < n$ (통상적으로, $m=8$)이다. 센서(600) 및 캡처 인터페이스(610)가 픽셀 값당 m-비트를 제공한다면 압축 모듈(625)은 필요하지 않게 된다.

주요 압축 프로그램(primary compressor)(628)은 압축된 센서 이미지 데이터를 수신하여, DWT 특허에서 논의된 DWT 기반의 압축과 같은 이미지 압축을 수행한다. RAM(629)은 DWT 특허에서 기술된 압축 기술을 수행시에 요구되는 각 채널/서브-밴드에 대한 DWT 계수 및/또는 양자화 임계값을 저장하는데 사용될 수 있다. 주요 압축 프로그램(628)은 압축된 채널마다 그리고 서브-밴드마다의 출력을 적응형 인코더(630)에 제공하도록 설계될 수 있다. 적응형 인코더(630)는 도4 및 도5에 나타난 디자인과 유사하게 설계될 수 있다. 본 발명의 일실시예에 따라, 적응형 인코더(630)는, 본 발명의 다른 실시예에서 나타나는 순수한 MHC 및 수정된 ZRLC와 같은, 다양한 2진 인코딩 구조를 수행하도록 장착될 수 있다. RAM(631)은 적응형 인코더(630)에 의해 사용되는 MHC LUT 및 ZRLC LUT를 저장하도록 구성될 수 있다. 적응형 인코더(630)는 인코딩 및 압축된 데이터를 저장 어레이(640)에 제공한다. 적응형 인코딩의 결과, 저장 어레이(640)는 이미지 데이터에 대한 많은 가능한 2진 인코딩 기술 중의 하나만이 유일하게 적용되는 통상의 비-적응형 인코더의 사이즈보다는 작게 설계될 수 있다. 각 이미지(또는 서브-밴드와 같은 이미지 압축 부산물)는 다른 것에 비해 하나의 인코딩 구조를 지지하는 독특한 특성을 가지기 때문에, 적응형 인코딩은 압축되거나 또는 그렇지 않은 이미지 데이터의 인코딩에서 특히 유리하다.

RAM 테이블(616, 626, 629, 631)의 각각은, 필요시 그들의 데이터를 로딩하고, 이후에 수정하기 위해, 버스(660)와 직접 통신할 수 있다. 또한, 이들 RAM 테이블 및 다른 RAM 테이블은 필요에 따라 중간 결과 데이터를 저장하는데 사용될 수 있다. 저장 어레이(640)의 데이터가 도6의 이미징 장치에 대해 외부로 전달될 준비가 되면, 이것은 전달을 위해 버스(660) 상에 위치될 수 있다. 또한, 버스(660)는 요구에 따라 RAM 테이블(616, 626, 629, 631)의 갱신을 도모할 수 있다.

도7은 본 발명의 일실시예에 따른 시스템을 도시하고 있다.

도시된 컴퓨터 시스템(710)은, 카메라(730)에 연결된 PC와 같은, 일반적인 또는 특별한 목적의 컴퓨팅 또는 데이터 프로세싱 머신이 될 수 있다. 카메라(730)는 디지털 카메라, 디지털 비디오 카메라, 이미지 캡처 장치, 이미징 시스템, 또는 이들의 조합이 될 수 있고, 장면(740)의 이미지를 포착하도록 사용된다. 필수적으로, 포착된 이미지는 이미지 프로세싱 회로(732)에 의해 처리되어, 이미지 메모리 유닛(734)에 충분히 저장될 수 있게 되고, 이것은 RAM 또는 고정된 디스크와 같은 그 밖의 저장 장치일 수 있다. 컴퓨터 시스템(710)으로 연결된 이미지 메모리 유닛(734) 내에 포함된 이미지는, 본 발명의 일실시예에 따라, 압축 및 적당히 인코딩된 이미지로 직접 저장될 수 있다. 스틸 이미징을 수행할 수 있는 대부분의 디지털 카메라에서, 이미지는 먼저 저장되고, 나중에 다운로드된다. 이것은 카메라(730)가 추가적 지연없이 빠르게 다음의 대상/장면을 포착할 수 있게 한다. 여러 실시예에서의 본 발명은, 특히 포착된 8-비트 베이어 패턴으로부터 직접 변환된 압축 이미지를 제공할 때, 카메라(730)의 계산상의 요구 및 이와 관련된 비용을 줄이게 되어, 보다 싼 카메라를 제공할 수 있게 된다.

이미지 프로세싱 회로(732)는 본 발명의 하나 또는 그 이상의 실시예에 따라 압축 및 적응형 인코딩을 수행한다. 압축 및 인코딩된 이미지가 컴퓨터 시스템(710)으로 다운로드되면, 이것은 디코딩되고 나서, 프린터(미도시) 또는 모니터 장치(720)와 같은 일부 출력 장치에 렌더링된다. 이미지 디코딩은 Pentium™(인텔 코퍼레이션사의 제품) 프로세서와 같은 프로세서(712), 및 명령어 어드레스 및 결과 데이터를 저장/로딩하는 RAM과 같은 메모리(711)를 이용하여 수행될 수 있다.

컴퓨터 시스템(710)은 카메라(730)에 도5 및 도6에 도시된 수정된 ZRLC 및 MHC를 위해 필요한 LUT를 제공할 수 있다.

대안적 실시예에서, 전송된 적응형 인코딩 및 수정된 ZRLC 기술은 카메라(730)에 직접 실행되기 보다는 컴퓨터 시스템 상에서 실행되는 소프트웨어 애플리케이션으로 달성될 수 있다. 이러한 실시예에서, 이미지 프로세싱 장치는 단지 압축된 이미지만을 유리하게 저장할 수 있다. 카메라(730)로부터 다운로드된 후에 인코딩 및/또는 디코딩을 수행하는데 사용되는 애플리케이션은 C++와 같은 언어로 쓰여진 소스 코드로부터 실행 가능하게 컴파일된 것일 수 있다. 이미지를 스케일링하는데 필요한 명령어에 대응하는 상기 실행가능 파일의 명령어는 디스크(718) 또는 메모리(711)에 저장될 수 있다. 또한, 이러한 애플리케이션 소프트웨어는 다른 시스템에서 사용하기 위해 네트워크 또는 컴퓨터-판독가능 매체 상에서 분배될 수 있다.

장면(740)의 이미지와 같은 이미지가 카메라(730)에 의해 포착되면, 이것은 이미지 프로세싱 회로(732)로 전송된다. 이미지 프로세싱 회로(732)는 다른 기능들 사이에서 압축된 이미지의 적응형 인코딩을 수행하는 IC 및 그 밖의 소자로 구성된다. 이미지 메모리 유닛(734)은 압축 및 인코딩된 데이터를 저장할 수 있다. 모든 픽셀이 처리되어 저장되거나, 또는 렌더링을 위해 컴퓨터 시스템(710)에 전달되면, 카메라(730)는 다음의 이미지를 포착하는데 자유로워 진다. 유저 또는 애플리케이션이 이미지의 다운로드를 요구/요청하면, 이미지 메모리 유닛 내의 인코딩된 이미지 데이터는 이미지 메모리 유닛(734)으로부터 I/O 포트(717)로 전달된다. I/O 포트(717)는 데이터를 메모리(711), 또는 선택적으로 디스크(718)에 일시적으로 저장하기 위해, 도시된 버스-브릿지 계층 구조(I/O 버스(715)-브릿지(714)-시스템 버스(713))를 사용한다. 컴퓨터 시스템(710)은 프로세서(712) 및 메모리(711)에/로부터 정보 전달에 이용되는 시스템 버스(713) 및 I/O 버스(715)에 연결된 브릿지(714)를 구비한다. I/O 버스(715)는 디스플레이 어댑터(716), 디스크(718) 및 직렬 포트와 같은 I/O 포트(717)등의 여러 I/O 장치에 연결된다. 이러한 I/O 장치, 버스 및 브릿지의 많은 조합이 본 발명에서 이용될 수 있고, 도시된 조합은 하나의 가능한 조합만을 예시한 것이다.

본 발명의 다른 실시예에서, 수정된 ZRLC 인코딩 및 적응형 인코딩은 텍스트 파일의 압축과 같은 일반적인 비-이미지 데이터의 압축을 수행하는데 사용될 수 있다. 이러한 데이터는 디스크, 메모리(711) 또는 다른 저장 메카니즘에 저장될 수 있고, 적당히 인코딩된 값으로 압축될 수 있다. 이러한 데이터의 인코딩에서, 수정된 ZRLC를 이용하여 인코딩되고, MHC를 이용하여 인코딩된 데이터 스트림의 부분을 검출할 수 있다. 많은 수의 연속한 제로가 포함된 데이터에서는, 임의의 큰 런-길이 값의 표현이 가능한 수정된 ZRLC 구조가 특히 효과적일 수 있다. 높은 엔트로피를 보이는 데이터에서는, MHC가 보다 적합할 수 있다.

앞의 명세서에서, 본 발명은 특정한 예시적인 실시예를 참조하여 설명되었다. 그러나, 첨부된 청구항에서 기술되는 바와 같이, 본 발명의 보다 넓은 사상 및 범위에서 벗어나지 않는 한, 다양한 수정 및 변경이 가능하다는 것은 명백한 사실이다. 따라서, 본 명세서 및 도면은 제한적 관점이라기 보다는 하나의 예시로서 간주되어야 한다. 이에 따라, 본 발명의 범위는 첨부한 청구항에 의해서만 제한되어야 한다.

CHANNEL R					
Level 1, Band HL, LH		Level 1, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	111		
4	11110				
5	11111				
Level 2, Band HL, LH		Level 2, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	1110		
4	11110	4	11110		
5	111110	5	11111		
6	111111				
Level 3, Band HL, LH		Level 3, Band HH		Level 3, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	111111	6	01010
7	1111111			7	01011
Level 4, Band HL, LH		Level 4, Band HH		Level 4, Band LL	
Range	Code	Range	Code	Range	Code
0	01	0	0	0	101
1	10	1	10	1	010
2	11	2	110	2	11
3	001	3	1110	3	000
4	0000	4	11110	4	001
5	00010	5	111110	5	011
6	000110	6	1111110	6	1000
7	0001110	7	1111111	7	10010
8	0001111			8	10011

Appendix A: Huffman tables for the R channel.

CHANNELS G1 and G2

Level 1, Band HL, LH		Level 1, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	111		
4	11110				
5	11111				
Level 2, Band HL, LH		Level 2, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	1110		
4	11110	4	11110		
5	111110	5	11111		
6	111111				
Level 3, Band HL, LH		Level 3, Band HH		Level 3, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	1111110	6	01010
7	1111111	7	1111111	7	01011
Level 4, Band HL, LH		Level 4, Band HH		Level 4, Band LL	
Range	Code	Range	Code	Range	Code
0	01	0	0	0	101
1	10	1	10	1	010
2	11	2	110	2	11
3	001	3	1110	3	000
4	0000	4	11110	4	001
5	00010	5	111110	5	011
6	000110	6	1111110	6	1000
7	0001110	7	1111111	7	10010
8	0001111			8	10011

Appendix B: Huffman tables for the G1 and G2 channel.

CHANNEL B					
Level 1, Band HL, LH		Level 1, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	111		
4	11110				
5	11111				
Level 2, Band HL, LH		Level 2, Band HH			
Range	Code	Range	Code		
0	0	0	0		
1	10	1	10		
2	110	2	110		
3	1110	3	1110		
4	11110	4	1111		
5	111110				
6	111111				
Level 3, Band HL, LH		Level 3, Band HH		Level 3, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	111111	6	01010
7	1111111			7	01011
Level 4, Band HL, LH		Level 4, Band HH		Level 4, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	101
1	10	1	10	1	010
2	110	2	110	2	11
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	011
6	1111110	6	1111110	6	1000
7	1111111	7	1111111	7	10010
				8	10011

Appendix C: Huffman tables for the B channel.

CHANNEL [R-G1]

Level 1, Band HL, LH	
Range	Code
0	0
1	10
2	110
3	1110
4	1111

Level 2, Band HL, LH		Level 2, Band HH	
Range	Code	Range	Code
0	0	0	0
1	10	1	10
2	110	2	110
3	1110	3	1110
4	11110	4	11110
5	111110	5	11111
6	111111		

Level 3, Band HL, LH		Level 3, Band HH		Level 3, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	1111110	6	01010
7	11111110	7	1111111	7	010110
8	11111111			8	010111

Level 4, Band HL, LH		Level 4, Band HH		Level 4, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	1111110	6	01010
7	11111110	7	11111110	7	010110
8	11111111	8	11111111	8	010111

Appendix D: Huffman tables for the [R-G1] channel.

CHANNEL [B-G2]

Level 1, Band HL, LH	
Range	Code
0	0
1	10
2	110
3	1110
4	1111

Level 2, Band HL, LH		Level 2, Band HH	
Range	Code	Range	Code
0	0	0	0
1	10	1	10
2	110	2	110
3	1110	3	1110
4	11110	4	1111
5	11111		

Level 3, Band HL, LH		Level 3, Band HH		Level 3, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	11111	5	0100
6	1111110			6	01010
7	1111111			7	01011

Level 4, Band HL, LH		Level 4, Band HH		Level 4, Band LL	
Range	Code	Range	Code	Range	Code
0	0	0	0	0	011
1	10	1	10	1	11
2	110	2	110	2	10
3	1110	3	1110	3	000
4	11110	4	11110	4	001
5	111110	5	111110	5	0100
6	1111110	6	1111110	6	01010
7	1111111	7	1111111	7	01011

Appendix E: Huffman tables for the [B-G2] channel.

(57) 청구의 범위

청구항 1.

워드(R)를 저장하기 위해 사용되는 비트 수를 결정하는 단계 - 여기서, 상기 워드(R)는 플래그를 수반하고, 상기 플래그와 워드(R)는 하나의 구조로 이루어짐 -;

플래그 1을 갖는 구조에서의 대응 워드(R)가 런 길이의 일부분을 형성하며, 다음 구조에서 워드(R)가 인접해 있다는 것을 나타내는 플래그 1을 갖는 구조를 특정하고, 대응 워드(R)가 런 길이의 최상위 부분(the most significant portion)이라는 것을 나타내는 플래그 0을 가진 구조를 특정하는 단계; 및

상기 각 비트가 전부 사용되었더라도 상기 워드(R)의 상기 비트 수에 의해 허용가능한 것보다 더 큰 길이의 런을 인코딩하는 단계 - 여기서, 상기 인코딩은 다수의 상기 구조를 함께 연결함(stringing)으로써 달성됨 -

를 포함하는 방법.

청구항 2.

제 1 항에 있어서,

상기 런은 데이터 셋트 내의 2개의 비-제로(non-zero) 값 사이의 제로의 수를 나타내는

방법.

청구항 3.

제 2 항에 있어서,

상기 비-제로 값은 수정된 허프만 코딩(Modified Huffman Coding)을 이용하여 인코딩되는

방법.

청구항 4.

제 2 항에 있어서,

상기 데이터 셋트는 이산 웨이블릿 변환(Discrete Wavelet Transform; DWT)으로부터 얻어지는 서브-밴드 데이터인

방법.

청구항 5.

제 3 항에 있어서,

상기 비-제로 값은 탐색 테이블(look-up table)을 어드레스하기 위해 이용되며, 그로부터 상기 비-제로 값에 대응하는 허프만 코드를 유도하는

방법.

청구항 6.

제 4 항에 있어서,

상기 데이터의 각각의 서브-밴드는 상기 서브-밴드의 엔트로피 특성에 따르는 방식으로 인코딩되며, 상기 방법은 상기 제로 및 비-제로 값이 모두 수정된 허프만 인코딩을 이용하여 인코딩되는 방법을 포함하는

방법.

청구항 7.

제 6 항에 있어서,

상기 데이터 셋트는 상기 DWT가 이미지에 대해 수행될 때 상기 데이터 셋트의 컬러 특성에 관련된 채널에 의해 더 분류되는

방법.

청구항 8.

삭제

청구항 9.

제 1 항에 있어서,

플래그 1과 워드(R) 1을 갖는 구조는 플래그 0과 워드(R) 1을 갖는 상기 구조 다음에 오는 비-제로 값이 이전의 비-제로 값에 인접해 있다는 것을 나타내는

방법.

청구항 10.

삭제

청구항 11.

삭제

청구항 12.

삭제

청구항 13.

삭제

청구항 14.

삭제

청구항 15.

M 비트의 그룹으로 각각 분할된 N-비트 카운터 - 여기서, M은 데이터 값을 인코딩하기 위한 강화된 제로 런 길이 코딩 방식(enhanced Zero Run Length Coding scheme)에서 워드(R)의 길이를 포함함 - ; 및

상기 카운터에 연결된 로직 네트워크 - 여기서, 상기 로직 네트워크는 플래그를 세팅하고 상기 카운터로부터의 선택된 상기 M 비트 그룹의 출력을 인에이블시키도록 구성됨 -

를 포함하고,

여기서, 상기 로직 네트워크는,

상기 카운터의 M개의 최상위 비트를 입력으로서 수신하도록 구성된 제1 OR 게이트 - 여기서, 상기 제1 OR 게이트 출력은 상기 카운터의 두번째의 M개의 최상위 비트의 플래그를 세팅함 - ;

다음의 M개의 최상위 비트 그룹과 상기 제1 OR 게이트의 출력을 입력으로서 수신하도록 구성된 제2 OR 게이트 - 여기서, 상기 제2 OR 게이트 출력은 상기 카운터의 그 다음 M 비트의 플래그를 세팅함 - ; 및

상기 제2 OR 게이트에 의해 수신되는 것 다음에 오는 M개의 최상위 비트 그룹과 상기 제2 OR 게이트의 출력을 입력으로서 수신하도록 구성된 제3 OR 게이트 - 여기서, 상기 제3 OR 게이트 출력은 상기 카운터의 최상위 M 비트에 속한 구조의 플래그를 세팅함 - 를 포함하는

장치.

청구항 16.

삭제

청구항 17.

제 15 항에 있어서,

상기 인에이블 신호는 상기 카운터로의 정지 및 리셋 신호 입력과 선택된 플래그의 논리 조합으로서 제공되고, 상기 최상위 M 비트의 플래그는 항상 제로로 세팅되는

장치.

청구항 18.

제 15 항에 있어서,

M=3인

장치.

청구항 19.

제 15 항에 있어서,

상기 로직 네트워크는 상이한 값의 M에 대해 수정되는

장치.

청구항 20.

각각 워드(R)와 플래그로 이루어진 구조를 추출함으로써 데이터 스트림 내의 유사한 값의 런을 디코딩하는 단계

를 포함하고,

여기서, 상기 플래그가 1인 경우에는, 그 구조에서의 대응 워드(R)가 런 길이의 일부분을 형성하도록 결정되고, 그 다음 구조에서 상기 R 값이 현재 구조에서의 워드(R)에 의해 형성된 상기 런 길이의 일부분에 인접하게 되며,

상기 플래그가 0인 경우에는, 대응 워드(R)가 런 길이의 최상위 부분이 되는

방법.

청구항 21.

제 20 항에 있어서,

플래그 1과 워드(R) 1을 갖는 구조는 플래그 0과 워드(R) 1을 갖는 구조 다음에 오는 비-제로 값이 이전의 비-제로 값에 인접한다는 것을 나타내는

방법.

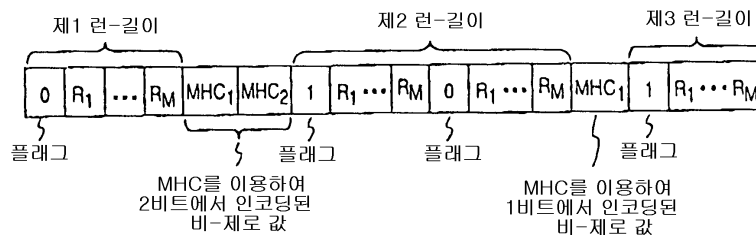
청구항 22.
삭제

청구항 23.
삭제

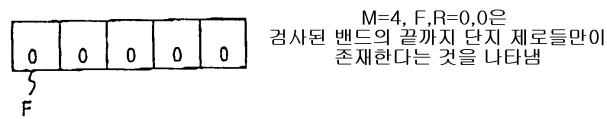
청구항 24.
삭제

도면

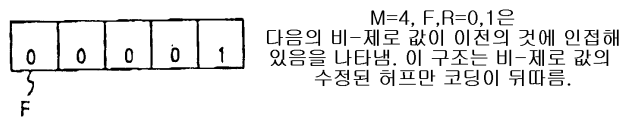
도면1a



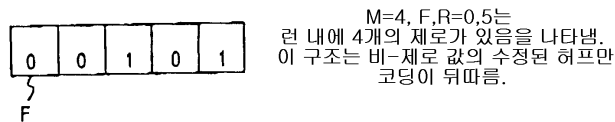
도면1b



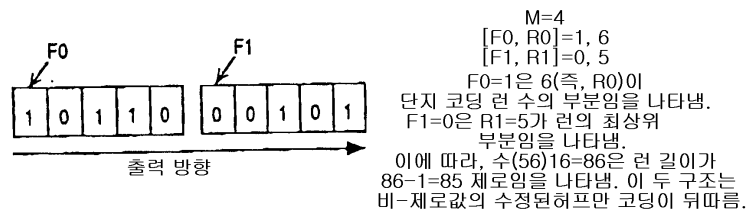
도면1c



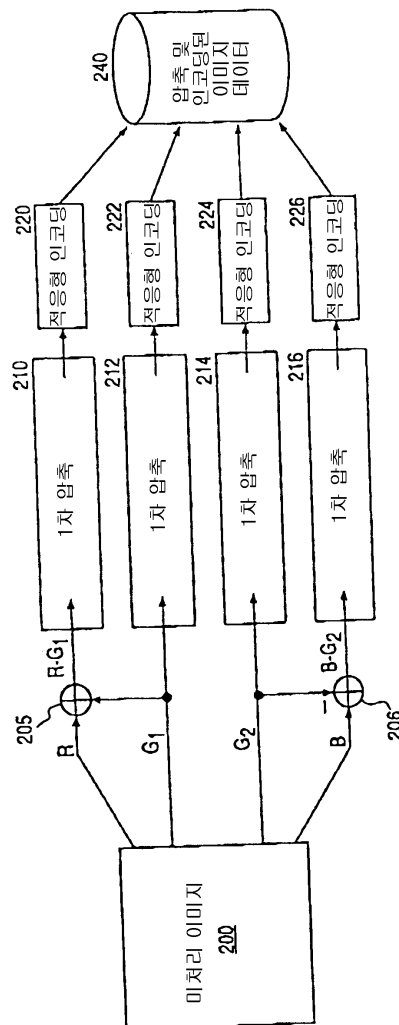
도면1d



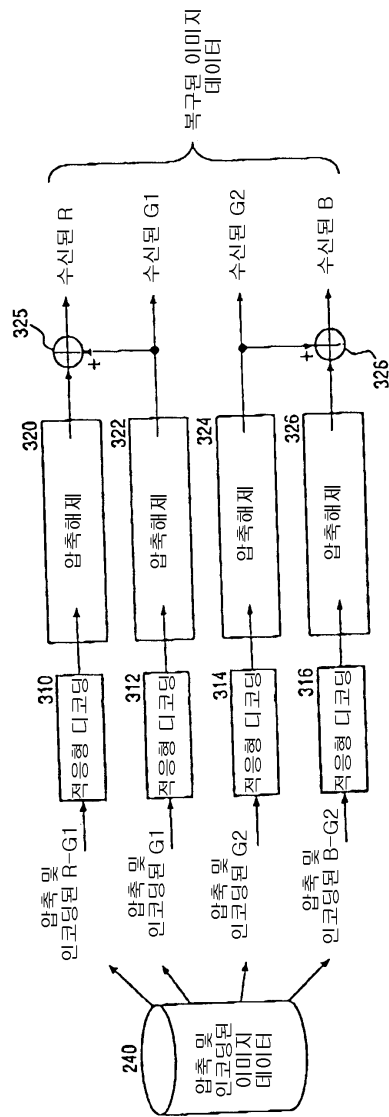
도면1e



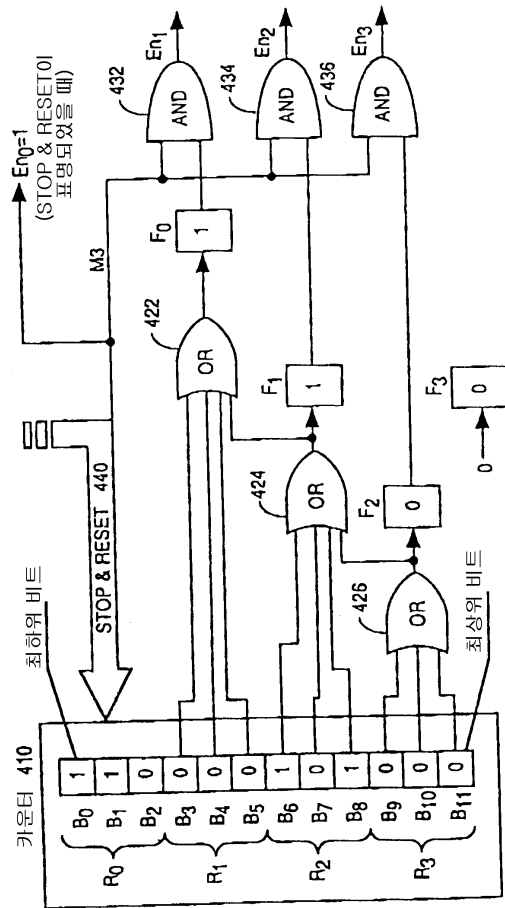
도면2



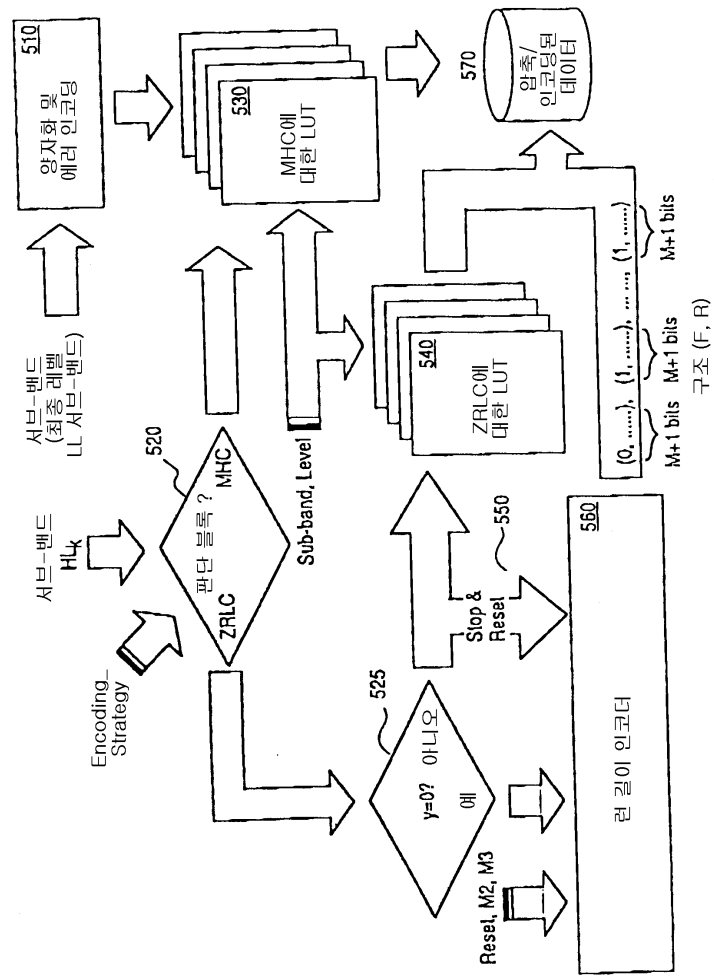
도면3



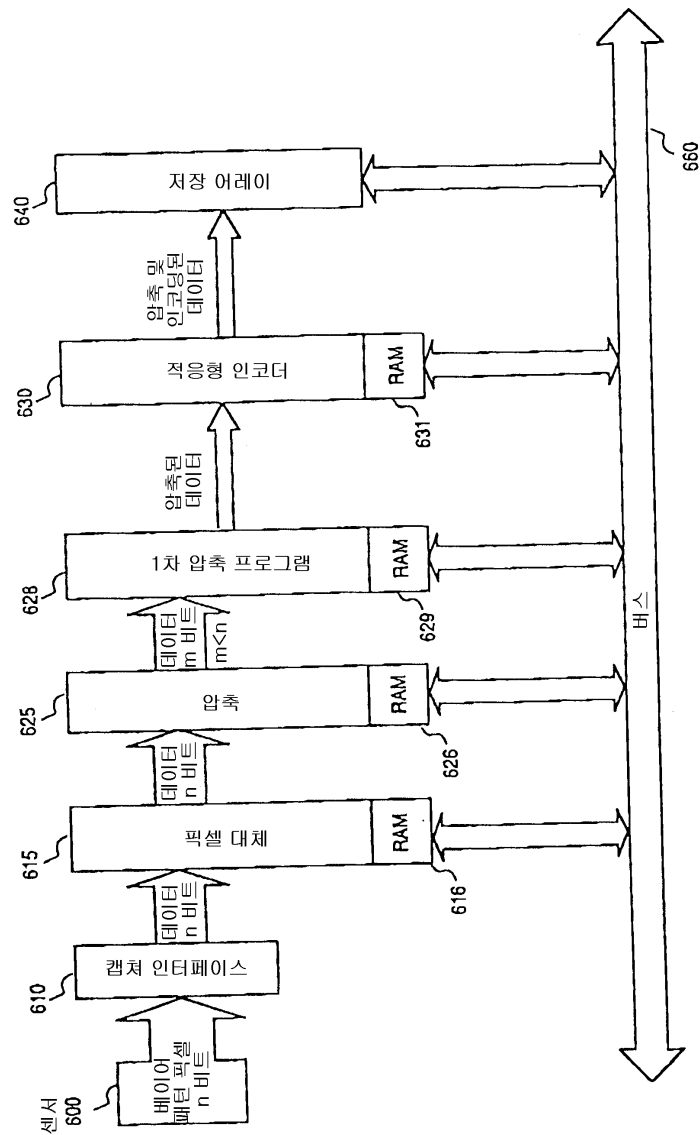
도면4



도면5



도면6



도면7

