(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.[6]
H04N 7/32

(45) 공고일자 2004년10월20일
(11) 등록번호 10-0437177
(24) 등록일자 2004년06월14일

(21) 출원번호 10-1995-0025805
(22) 출원일자 1995년08월18일

(65) 공개번호 10-1996-0009753
(43) 공개일자 1996년03월22일

(30) 우선권주장 9410158 1994년08월19일 프랑스(FR)

(73) 특허권자
똥송 멀티메디아 에스. 에이.
프랑스, 92400 꾸르베봐, 라 데팡스, 5, 플라스 데 보스게스 9

(72) 발명자
쟝-미셀바흐드
프랑스공화국에프-38920크로레스하뮤드라그네린루빅토르휴고259

앨라인피어슨
프랑스공화국에프-38430모이랜즈알레드사브와이에122

필리페폴
프랑스공화국에프-38320에이벤즈애브뉴드포이셋12

패티야싸
스위스연방씨에이취-1256트로이넥스체민드드리제6디

(74) 대리인
남상선

심사관 : 이진익

(54) 이동량추정장치

요약

본 개시는 블럭을 기초로한 이미지 압축 시스템에서 블럭 매칭에 의하여 이동량을 추정하기 위한 장치에 관한 것으로, 그 장치는 두 값들 사이의 에러함수의 N열 x M라인 계산수단, 추정될 이동량을 갖는 블럭을 나타내는 값들을 기억하기 위한 N x M 수단, 기준 윈도우의 블럭을 나타내는 값을 기억하기 위한 N x M수단, 계산수단에 개별적으로 링크될 두 개의 기억 수단, 계산의 N 요소의 열을 위한 에러함수의 값을 동시에 합계하는 가산수단(M)을 포함하는 것이 특징이다. 본 발명은 이미지 압축에 적용할 수 있다.

대표도

도 1

명세서

도면의 간단한 설명

제 1도 본 발명의 실시예에 따른 이동량 추정 프로세서의 개략적인 블럭도.

제 2a 및 2b도 본 실시예에서 사용된 비교 단계를 개략적으로 도시한 도.
제 3도 본 실시예에 따른 시스톨릭 네트워크의 개략적인 블럭도.
제 4도 제 3도의 시스톨릭 네트워크의 소위 'S' 셀의 개략적인 블럭도.
제 5도 비-비월 모드에서의 동작을 도시하기 위하여 사용되는 본 발명에 따른 간단한 시스톨릭 네트워크의 개략적인 블럭도.
제 6도 비월 모드에서의 동작을 도시하기 위하여 사용되는 본 발명에 따른 간단한 시스톨릭 네트워크의 개략적인 블럭도.
제 7도 본 발명의 제 1 변형의 코프로세서의 개략적인 블럭도.
제 8도 본 발명의 제 2 변형의 코프로세서의 개략적인 블럭도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 픽셀의 블럭에 근거한 이미지 압축 시스템에서 이동 벡터의 이동량 및 편향을 추정하기 위한 장치에 관한 것이다. 본 발명에 따른 장치는 특히 텔레비젼 이미지의 압축 및 MPEG('동화상 전문가 그룹')의 현재 또는 미래 압축 표준 에 근거한 시스템에 사용된다.
다수의 이미지 압축 시스템이 픽셀의 블럭을 처리한다. MPEG의 경우에 있어서, 이미지는 8 ×8 픽셀 블럭으로 분할되고, 네 개의 블럭은 16 x 16픽셀의 매크로블럭으로 어셈블된다. 블럭 또는 매크로블럭은 이미지들 사이 및 각 이미지내부에서의 용장도를 줄이기 위하여 다양한 가공을 한다. 이들 가공중 하나가 이동량의 추정이다. 이러한 처리과정은 과거 또는 미래의 이미지에 관하여 현재의 이미지의 변위를 추정하는 것으로 구성된다. 변위를 결정하는 하나 또는 그 이상의 이동량 벡터들이 규정되고; 이들 이동량 벡터들은 과거 또는 미래의 이미지에 근거하여 만들어질 현재 이미지의 예측을 가능하게 한다. 우리는 이러한 예측치와 현재 이미지 사이의 차이를 결정하고, 이러한 차이는 미래 이미지 압축을 결정한다. 현재, 이동량 벡터는 일어날 수 있는 전송 에러의 영향을 감소시킬 목적으로 소위 '중심(inter)' 또는 '쌍-방향(bi-directional)' 이미지용으로 사용되고 있지만, '내부(intra)' 이미지용으로 적용될 수도 있다.
이동량의 추정은 일반적으로 각 매크로블럭에 대하여 시행된다. 설명을 간략화하기 위하여 본 명세서의 남은 부분에서 우리는 단일 이동 벡터를 각 매크로블럭에 관련시킬 것이고, 이 벡터는 과거 이미지로 부터 현재 이미지의 매크로블럭을 결정한다. 과거 이미지를 대신해서, 우리는 코더내에 기억된 기준 이미지를 사용할 수 있고, 이 이미지는 디코더에 의해 디코드된 이미지에 대응한다(즉, 이미지는 압축으로 인한 정보의 손실을 고려한다).
기준 이미지에서 '최고의' 매크로블럭을 결정하기 위한 기술은 현재 매크로 블럭과 기준 이미지와 비슷한 크기의 블럭을 비교하도록 구성된다. 비교원리로써, 우리는 기준 이미지에서 매크로블럭의 각 가능한 위치를 계산하기 위한 에러함수(Err)를 규정하였다. (Err) 함수에 대하여 최저값을 갖는 위치가 변위벡터를 결정할 것이다. 이 벡터의 정확도는 가변적인데, 그것은 가능한 계산 용량 및 이동량 벡터를 코딩하기 위해 제한된 통과대역에 근거하기 때문이다. 픽셀의 부분 정밀도는 중심-픽셀 보간법(inter-pixcel interpolation)에 의해 성취될 수 있다. 만약 이동량 벡터들의 길이가 제한된다면, 적당한 크기의 검색 윈도우가 현재 매크로블럭에 의해 규정된 위치 주변에 위치하게 된다.
이동량 벡터의 계산은 강력한 계산 수단을 필요로 하고, 요구되는 정확도, 이미지의 해상도, 그들의 주파수, 등이 증가한다.
일반적으로 사용되는 에러함수는:

$$\mathrm{Err}(m,n) = \sum_{i}\sum_{j} D(a(i,j),\ b(i+m, j+n))$$

여기서, i 및 j는 매크로블럭내의 위치를 가리키고, m 및 n은 이동량 벡터의 좌표이며, a 및 b는 각각 현재 매크로블럭 및 윈도우 또는 검색 이미지의 변위된 매크로블럭의 픽셀의 휘도값이다.
함수 D는 예를들면 D(x,y) = ｜x - y｜이 될 수 있다.
MPEG 표준안을 따르는 이미지 압축에서의 이동량 벡터의 사용상의 추가 정보에 대해서는 참조를 활용하시오.['Information Technology, Generic coding of moving pictures and associated audio, Recommendation H.262 ISO/IEC 13818- 2(Video) Committee Draft of November 1993.']
본 발명의 목적은 특별한 장점을 갖는 구조를 구비한 '블럭 매칭 프로세서'를 제안하는 것이다.
본 발명의 목적은 블록매칭에 의하여 블럭에 근거한 이미지 압축 시스템에서의 이동량 추정을 위한 장치로, 그 내부에는:
- 두 값 사이의 에러함수(Err)를 계산하기 위한 N 열 x M 라인 수단,
- 추정될 이동량을 갖는 블럭을 나타내는 값을 기억하기 위한 (N x M) 수단,
- 기준 윈도우의 블럭을 나타내는 값을 기억하기 위한 (N x M) 수단, 두 개의 기억 수단은 상기 계산수단에 개별적으로 링크되고,

- 계산의 N 열 성분에 대한 에러함수값을 동시에 합산하는 M개의 가산수단을 포함한다.
특정 실시예에 있어서, 상기 장치는 직렬로 연결된 M개의 버퍼(B)로 구성되는 N 라인 네트워크를 포함하고, 이 네트워크에서 기준 윈도우의 대역은 열단위로 전파된다.
특정 실시예에 있어서, 각 버퍼의 출력은 에러함수를 계산하는 수단의 입력에 연결된다.
특정 실시예에 있어서, 계산수단은 버퍼에서 출력된 값과 추정될 이동량을 갖는 블럭을 나타내는 값으로 부터 에러함수를 계산한다.
특정 실시예에 있어서, 각 계산 수단의 계산 결과는 버퍼에 기억된다.
특정 실시예에 있어서, 주어진 열의 버퍼에 기억된 모든 결과들은 가산기에 의하여 합계된다.
특정 실시예에 있어서, 열에 대한 가산결과는 네트워크에서 기준 윈도우의 블럭의 전파를 고려한 지연과 함께 다음 열의 가산기에 공급된다.
특정 실시예에 있어서, 상기 지연은 가산기 사이에 직렬인 버퍼에 의하여 수행된다.
특정 실시예에 있어서, 추정될 이동량을 갖는 블럭을 나타내는 값을 기억하는 수단은 두 픽셀값을 기억하고, 두 값 중 하나는 에러함수 계산수단으로 멀티플렉싱된다.
특정 실시예에 있어서, 상기 두 픽셀값들은 주어진 블럭의 상위 및 하위 부-블럭을 나타내고, 네트워크 입력에 제공되는 그 값들은 기준 윈도우의 상위 부-대역 및 하위 부-대역을 나타내는 교번(alternating) 값이고, 이러한 종류의 동작은 '비월(interlaces)'로 알려져 있다.
특정 실시예에 있어서, 상기 장치는 이들 두 부-블럭에 대응하는 에러함수의 합계값을 가산하는 수단을 포함한다.
특정 실시예에 있어서, 비월 모드에서, 한 계산 주기의 보조 지연이 가산기들 사이에 삽입된다.
특정 실시예에 있어서, 에러함수는 거리의 함수이다.
특정 실시예에 있어서, 버퍼의 각 라인은 기준 윈도우의 라인을 기억하기 위한 수단에 의하여 공급된다.
특정 실시예에 있어서, 상기 기억 수단은 FIFO(선-입력-선-출력) 스택이다.
특정 실시예에서, 적어도 N+1개의 라인들의 메모리 수단이 존재하고, 여기서 상기 수단들 중 N개는 버퍼들의 N 라인들로 제공되고, 메모리의 보조 수단은 상기 N개의 라인들의 메모리 수단에 메모리된 기존 윈도우의 밴드와 바로 이어서 처리되는 밴드 사이에서 상이한 대응 라인들의 값으로 로딩된다.
특정 실시예에 있어서, 스위칭 회로는 라인 기억 수단과 버퍼 네트워크 사이에 포함되고, 상기 스위칭 수단은 라인의 기억 수단을 버퍼의 적당한 라인에 연결한다.
본 발명의 다른 목적은 상기 설명된 장치를 수행하는 이미지 압축 시스템의 이동량 벡터를 계산하기 위한 장치이고, 그 내부에는:
- 입력 수단,
- 출력 수단,
- 메모리와 관련된 제어수단,
- 현재 이미지 및 기준 이미지 부분을 기억하기 위한 수단,
- 상기 기억 수단의 어드레싱 수단,
- 현재 이미지 부분 및 기준 이미지 부분 사이의 에러함수의 계산수단,
- 및 상기 제어 수단과 접속하는 수단을 포함한다.
특정 실시예에 있어서, 입력 수단은 상기 이미지를 나타내는 데이타를 데이타 버스를 통해 기억수단으로 로드하는데 사용되는 입력 포트이다.
특정 실시예에 있어서, 상기 데이타는 재판독되고 에러함수의 계산수단에 공급된다.
특정 실시예에 있어서, 제어수단을 프로그램하는 수단이 포함된다.
특정 실시예에 있어서, 에러함수의 계산 수단은 N x M 셀의 시스톨릭(systolic) 네트워크를 포함한다.
특정 실시예에 있어서, 데이타는 적어도 N개의 메모리의 뱅크를 통하여 상기 시스톨릭 네트워크로 공급된다.
마지막으로, 본 발명의 다른 목적은 상기 장비들 중 적어도 하나를 포함하는 완전한 이미지 압축 시스템을 제공하는 것이다.
본 발명은 더 잘 이해될 것이고 다른 장점 및 특성들은 실시예로써만 선택된 제한없는 특정 실시예에 대한 다음 설명을 읽으므로써 분명하게 될 것이다.
본 발명에 따른 장치는 MPEG 표준: 프레임 또는 이미지를 위한 벡터, 과거 이미지에서 현재 이미지로 또는 미래 이미지에서 현재 이미지로의 벡터(각각 '순방향 벡터' 및 '역방향 벡터'로 알려져 있는)의 요구에 따라 다른 종류의 이동량 벡터로 계산될 수 있다.
제 1 도는 이동량 추정 프로세서(101)의 구성을 도시한다. 프로세서(101)는 RISC 제어기(102)(RISC-'감소된 명령 세트를 갖는 컴퓨터'), 시스톨릭 네트워크 코프로세서(103), 이미지를 로드하기 위한 세 개의 독립적인 입력 포트(104, 105, 106), 이동량 벡터를 공급하기 위한 출력 포트(107), 및 지역적인 RAM(108) 및 ROM(109)을 포함한다.
몇개의 독립적인 입력 포트의 존재는 동일한 수의 독립적인 비디오 소스들이 처리될 수 있도록 하여준다. 변형에 있어서, 각 입력 포트에는 이동량에 대한 계층 적 추정을 수행하기 위한 부-표본화 회로가 제공된다.
상기 언급된 이러한 상이한 부-어셈블리들은 그 중에서도 특히 외부 DRAM 메모리(110)에 연결된 데이타 버스(111)를 통해 통신한다. 메모리 운영 장치(112)는 지역 메모리들 사이의 데이타 이동을 제어한다. (기준 또는 현재의) 이미지에 대응하는 데이타는 동적 메모리(110)로 라우트된다. 그들은 재독출되고 적당한 시간에 코프로세서(103)에 제공된다. 코프로세서(103)에 의해 계산된 이동량 벡터는 버퍼 메모리(MV BUFF)(114)를 통하여 동적 메모리(110)으로 전송된다. 적당한 시간에, 이들 벡터들은 재독출되고 출력 포트(107)를 통하여 전송된다.

프로세서(101)는 또한 다운로딩 코드 및 계산 파라메터에 의하여 RISC 제어기(102)의 프로그래밍 및 구성을 제공하는 마이크로프로세서 인터페이스(115)를 구비한다. 동적 메모리(110)에 기억되어 있는 이동량 벡터는 또한 인터페이스(115)를 통해 접근할 수 있다.
장치 MAX-AVR(113)은 각 이미지에 대한 평균 이동량 벡터 및 최대 이동량 벡터를 계산한다.
본 실시예에서, RISC 제어기(102)는 62개의 레지스터와 64워드 데이타 메모리를 갖는 19비트 프로세서이다. 제어기(102)의 목적은 프로세서(101)의 다양한 장치들의 요청을 조절하고 적당한 시간에 그들을 동작하는 것이다.
RISC 제어기(102)는 지역 RAM(108) 및 지역 ROM(109)에 연결되어 있고, 후자는 일반적으로 사용되는 루틴을 포함한다.
본 명세서의 남은 부분에서, 우리는 단일 시스톨릭 네트워크의 동작을 먼저 서술할 것이다. 이것은 8개의 이러한 네트워크들을 관련시키는 코프로세서(103)의 동작을 보다 더 쉽게 이해될 수 있도록 하여줄 것이다.
제 2a도는 시스톨릭 네트워크에서의 정보의 순환을 도시한다. 네트워크의 기본적인 기능은 픽셀의 현재 블럭과 기준 윈도우의 내용을 비교하는 것이다. 현재 블럭은 시스톨릭 네트워크(201)에 기억된다. 기준 윈도우는 대역을 통해 네트워크로 전송된다. 에러 함수(Err)는 각 가능한 위치에 대해 계산된다. 구성요소(202)는 가장 작은 값을 갖는 에러 함수 및 이에따른 이동량 벡터를 결정한다.
제 2b도는 시스톨릭 네트워크로 전송될 대역을 결정하는 방법을 도시한다. 각 대역은 네트워크의 라인수(본 실시예에서는 4)에 대응하는 픽셀의 라인수를 갖는다. 제 1 대역은 기준 윈도우의 4개의 첫 번째 라인들을 포함한다. 제 1 대역이 완전히 처리되는 경우, 처리과정은 다음 대역으로 계속되고, 그것은 제 1 대역의 마지막 세 라인 및 제 1 대역에 바로 다음 라인을 포함한다. 다시 말하면, 각 대역은 이전 대역에 대하여 한 라인이 쉬프트된 것이다.
각 대역은 열 단위로 시스톨릭 네트워크에 제공된다. 따라서 각 대역과 동일한 라인 수를 갖는 현재 블럭은 각 대역의 동일한 크기의 모든 블럭과 비교된다. 결국, 현재 블럭은 동일한 크기의 모든 기준 윈도우의 블럭과 비교될 것이다.
우리는 두 대역이 픽셀의 한 라인만 다르다는 것에 주의하자. 이 용장은 메모리(110)로 부터 데이타의 로딩을 제한하기 위하여 나중에 사용되도록 놓여질 것이다.
제 3도는 4라인 5열의 시스톨릭 네트워크를 도시한다. 이 네트워크는 소위 'S'셀, 버퍼 'B', 가산기 'Add', 바이패스 셀(301) 및 최종 바이패스 셀(302)을 포함한다.
(S)셀은 ｜a - b｜계산을 수행하고, 여기서 a 및 b는 각각 현재 블럭의 필셀 및 기준 윈도우의 픽셀값이다. 주어진 라인의 각 (S)셀의 출력은 버퍼(B)를 통하여 다음 (S)셀의 입력에 연결된다. 버퍼(B)는 또한 라인의 첫번째(S)셀 앞에 존재한다. b 픽셀의 갑들은 매 사이클동안 주어진 라인상의 한 버퍼로 부터 다음 버퍼로 이동한다. 상기 버퍼는 (S)셀이 그들의 계산을 수행하는 동안 b 값을 유지한다.
각 (S)셀은 또한 계산 ｜a - b｜의 결과를 제공하는 출력(C)를 갖는다. 출력(C)는 또한 다른 버퍼(B)(주의: 제 3도에서, 비록 그들이 동일한 기능을 갖지는 않지만 모든 버퍼가 (B)로 표시된다)에 연결된다. 주어진 열의 모든 그러한 버퍼는 동일한 가산기 Add에 연결된다. 따라서 열만큼 많은 가산기가 존재한다.
가산기의 출력은 직렬인 세 버퍼(B)에 의해 다음 가산기의 입력에 연결된다. 선택된 동작 방법에서, 이들 세 버퍼 중 마지막 것은 바이패스 셀(301)에 의하여 단락회로가 되거나 또는 단락회로가 되지 않는다.
최종 바이패스 셀(302)은 정상적인 비-비월 동작에서의 각 사이클에서 에러함수의 값, 또는 비월 모드에서 매 두 사이클마다의 에러함수값을 제공한다. 후자의 경우, 최종 바이패스 셀(302)에 구비된 가산기는 연속적인 두 에러함수값을 가산한다.
변경에서, 가산기 Add는 '3-입력 캐리 저장 가산기'의 형태이다.
제 4도는 (S)셀의 기능을 더 자세히 도시한다. (S)셀은 입력(401), 입력(401)에 직접 연결된 출력(402), 하나는 입력(401)에 연결되고 다른 하나는 선택 라인(405)을 통해 버퍼(406)의 내용 또는 버퍼(407)의 내용을 선택할 수 있는 멀티플렉서(404)의 출력에 연결된 두 입력을 갖는 계산 장치(403)를 포함한다. 버퍼(406 및 407)의 입력은 또한 셀의 입력(401)에 연결된다.
두 버퍼 각각은 현재 블럭의 팍셀값을 포함한다. 이들 값들은 적당한 시간에 입력(401)을 통해 로드된다. 버퍼의 제어회로는 원칙적인 것으로 자세히 설명되지 않을 것이다. 그 중에서도 특히 병렬인 두 버퍼의 존재는 단일 시스톨릭 네트워크에서의 두개의 현재 블럭들의 동시 저장을 가능하게 한다. 이러한 방법으로, 두 번의 비교가 기준 윈도우내의 주어진 위치에 대해 수행될 수 있고, 그것은 동적 메모리(110)와 코프로세서(103) 사이에 전송되는 데이타량을 감소한다.
각 (S)셀에 두 개의 버퍼가 존재하는 것은 또한 시스톨릭 네트워크의 라인 수와 비교하여 현재 블럭의 라인수가 두 배가 되는 소위 '비월'동작을 가능하게 한다. 이러한 동작은 이후에 논의될 것이다.
계산 장치는 그의 입력에 존재하는 두 값의 차이의 절대값을 제공한다.
본 실시예에서, 우리는 기준 윈도우의 핏셀값 b가 (S)셀들 사이의 버퍼(B)에 기억되고, 반면에 현재 블럭의 픽셀값은 (S)셀의 버퍼에 기억된다는 것에 주의한다.
우리는 또한 (S)셀에 의해 생성된 중간 결과들을 합계하는 함수가 전체적으로 (S)셀의 각 열의 바닥에 있는 가산기에 의해서 수행됨에 주의해야 한다.
시스톨릭 네트워크의 정상(비-비월)동작은 제 5도를 고려하여 설명될 것이 고, 그것은 단일 라인과 8개의 열을 갖는 시스톨릭 네트워크를 나타낸다. 많은 라인을 가진 네트워크의 동작은 아주 유사하고, 각 (S)셀의 출력에서의 결과(출력 (S))는 동시에 가산기에 나타난다. 정상 동작에서, 단지 2개의 버퍼(B)가 열의 끝에 있는 각 가산기 사이에 필요하다. 명료성을 위하여, 제 3 버퍼 및 바이패스 셀은 도면에 도시되지 않았다.
픽셀 값 a는 이미 네트워크에 기억되어 있다. b 값은 각 사이클에서 네트워크의 입력으로 나타난다. 표 1은 사이클 번호 대 출력(C)의 거듭제곱근풀이(evolution)를 나타낸다.

주기	C0	C1	C2	C3	C4	C5	C6	C7
0	_	_	_	_	_	_	_	_
1	**|b0-a0|**	_	_	_	_	_	_	_
2	|b1-a0|	|b0-a1|	_	_	_	_	_	_
3	|b2-a0|	**|b1-a1|**	|b0-a2|	_	_	_	_	_
4	|b3-a0|	|b2-a1|	|b1-a2|	|b0-a3|	_	_	_	_
5	|b4-a0|	|b3-a1|	**|b2-a2|**	|b1-a3|	|b0-a4|	_	_	_
6	|b5-a0|	|b4-a1|	|b3-a2|	|b2-a3|	|b1-a4|	|b0-a5|	_	_
7	|b6-a0|	|b5-a1|	|b4-a2|	**|b3-a3|**	|b2-a4|	|b1-a5|	|b0-a6|	_
8	|b7-a0|	|b6-a1|	|b5-a2|	|b4-a3|	|b3-a4|	|b2-a5|	|b1-a6|	|b0-a7|
9	|b8-a0|	|b7-a1|	|b6-a2|	|b5-a3|	**|b4-a4|**	|b3-a5|	|b2-a6|	|b1-a7|
10	|b9-a0|	|b8-a1|	|b7-a2|	|b6-a3|	|b5-a4|	|b4-a5|	|b3-a6|	|b2-a7|
11	|b10-a0|	|b9-a1|	|b8-a2|	|b7-a3|	|b6-a4|	**|b5-a5|**	|b4-a6|	|b3-a7|
12	|b11-a0|	|b10-a1|	|b9-a2|	|b8-a3|	|b7-a4|	|b6-a5|	|b5-a6|	|b4-a7|
13	|b12-a0|	|b11-a1|	|b10-a2|	|b9-a3|	|b8-a4|	|b7-a5|	**|b6-a6|**	|b5-a7|
14	|b13-a0|	|b12-a1|	|b11-a2|	|b10-a3|	|b9-a4|	|b8-a5|	|b7-a6|	|b6-a7|
15	|b14-a0|	|b13-a1|	|b12-a2|	|b11-a3|	|b10-a4|	|b9-a5|	|b8-a6|	**|b7-a7|**

표 1

사이클 0동안, b0값은 네트워크의 입력에 제 1 버퍼(B)(501)에 기억된다. 사 이클 1동안에만 결과가 출력(C0)에서 버퍼(502)내에 기억된 제 1 (S)셀에 의해 공급된다. 사이클 1동안, 값 b0는 또한 제 2 버퍼(B)(503)에 의해 기억된다. 네트워크의 동작의 실제 시간에, 우리는 기준 윈도우내의 현재 블럭(8 픽셀 라인)의 주어진 위치에 대응하는 모든 차이의 절대값의 합을 마지막 가산기의 출력에서 찾을 수 있다. 기준 윈도우의 제 1 라인에 대하여 가산기들에서 가산되어야만 하는 결과들은 표에서 진한 문자로 표시되었다.

합계 함수는 버퍼(504)에 의하여 가산기들 사이에 삽입된 지연에 영향을 준다. 두 사이클의 지연은 가산기의 결과들의 전파가 올바로 성취되기 위하여 삽입되어야만 한다는 것을 쉽게 알 수 있다. 이것은 각 가산기 사이에 두 개의 버퍼의 존재를 설명한다. 각 버퍼는 한 사이클의 지연을 삽입한다. 다시 말하면, 두 개의 제 1 가산기 사이의 두 버퍼(504 및 505)는 버퍼(501 및 502)에 의해서 삽입된 지연을 소거한다.

15번째 사이클의 마지막에, 현재 블럭의 제 1 가능한 위치에 대하여 함수 Err 값이 네트워크의 출력(506)에 나타난다.

이러한 동작 동안에, 각 (S)셀에 기억될 두 값 중 하나만 사용됨에 주의하자. 변형에서, 현재 블럭의 픽셀값의 단일 기억 버퍼가 각 (S)셀에 사용된다.

제 6도는 소위 시스톨릭 네트워크의 '비월' 동작을 도시한다. 이 방법은 시스톨릭 네트워크의 라인 수의 두 배의 라인 수를 포함하는 현재 블럭을 위한 에러 함수의 결정을 가능하게 한다. 이것은 각 (S)셀의 과외(extra) 버퍼 및 각 가산기 사이의 과외 버퍼를 가산함으로써 간단히 성취된다.

제 6도는 보조 버퍼가 각 가산기쌍 사이에 삽입된다는 사실을 제외하면 제 5도와 유사하다. 게다가, 각 (S)셀은 제 4도에 도시된 바와 같이 두개의 내부 버퍼를 포함한다. 짝수 사이클 동안, 각 (S)셀의 제 1 버퍼의 내용은 대응하는 계산 장치로 멀티플렉싱되고, 반면에 홀수 사이클 동안 제 2 버퍼의 내용이 사용된다.

2배 크기의 현재 블럭은 수직으로 중첩된 두 개의 부-블럭 a 및 a'로 분리된다. 제 1 부-블럭 a는 각 (S)셀의 제 1 버퍼에 기억되고, 반면에 제 2 부-블럭 a'은 각 (S)셀의 제 2 버퍼에 기억된다.

이상적인 방법으로, 기준 윈도우의 대역은 라인수의 2배이다. 그것은 상위 부-대역(b)와 하위 부-대역(b')을 포함한다. 짝수 사이클 동안, 부-대역(b)의 열이 시스톨릭 네트워크의 입력에 존재하는 반면, 홀수 사이클 동안은 b'의 열이 존재한다.

표 2는 각 사이클 이후에 출력(C)의 상태를 나타낸다. 앞서 설명한 바와 같이, 진한 글씨는 부-블럭에 대응하는 에러 함수를 구성하기 위하여 합산되어야만 하는 중간 결과를 가리킨다.

비-비월 동작과 비교하여, 두 개의 부-블럭과 두 개의 부-대역 사이의 이러한 변경은 주어진 위치에 대응하는 축적된 에러의 가산기 출력의 존재와 다음 가산기의 입력에 대응하는 중간 결과의 존재 사이에 한 사이클의 보조 지연을 삽입한다. 이러한 이유때문에, 제 3 버퍼가 각 가산기쌍 사이에 삽입된다.

제 6도의 네트워크의 최종 가산기(601)는 가산기(602)의 제 1 입력과 가산기(602)의 제 2 입력에 연결된 출력을 갖는 버퍼(603) 모두에 연결된다.

부-블럭에 대응하는 에러함수는 매 사이클에서(네트워크가 알맞게 초기화될 때마다) 가산기(601)의 출력에 공급된다. 표 2는 부-블럭(a')에 대응하는 에러함수값이 부-블럭(a)에 대응하는 주기 이후에 한 사이클이 지연되는 것을 나타낸다. 두 값들은 전체 블럭(a+a')에 대응하는 에러함수의 값을 얻기 위하여 가산되어야만 한다. 제 1 사이클 동안, a에 대응하는 값은 버퍼(603)에 기억된다. 다음 사이클에서, 블럭(a')에 대응하는 값은 가산기(601)의 출력에 존재한다. 가산기(602)는 그들의 합을 계산한다. 그에 따라 유효한 결과가 매 2 사이클마다 가산기(602)의 출력에 존재한다.

이러한 '비월' 동작 모드는 네트워크의 라인수 및 지역 메모리의 수를 감소(아래 표를 참조)시킬 수 있어 매우 흥미롭다.

변형에 있어서, 시스톨릭 네트워크는 네트워크의 라인수의 N배와 같은 라인수를 갖는 현재 블럭을 처리하기 위하여 설계되었고, 여기서 N 은 2 보다 큰 정수이다.

표 2

주기	C0	C1	C2	C3	C4	C5	C6	C7
0	_	_	_	_	_	_	_	_
1	**|b0-a0|**	_	_	_	_	_	_	_
2	|b0'-a0'|	|b0-a1|	_	_	_	_	_	_
3	|b1-a0|	|b0'-a1'|	|b0-a2|	_	_	_	_	_
4	|b1'-a0'|	**|b1-a1|**	|b0'-a2'|	|b0-a3|	_	_	_	_
5	|b2-a0|	|b1'-a1'|	|b1-a2|	|b0'-a3'|	|b0-a4|	_	_	_
6	|b2'-a0'|	|b2-a1|	|b1'-a2'|	|b1-a3|	|b0'-a4'|	|b0-a5|	_	_
7	|b3-a0|	|b2'-a1'|	**|b2-a2|**	|b1'-a3'|	|b1-a4|	|b0'-a5'|	|b0-a6|	_
8	|b3'-a0'|	|b3-a1|	|b2'-a2'|	|b2-a3|	|b1'-a4'|	|b1-a5|	|b0'-a6'|	|b0-a7|
9	|b4-a0|	|b3'-a1'|	|b3-a2|	|b2'-a3'|	|b2-a4|	|b1'-a5'|	|b1-a6|	|b0'-a7'|
10	|b4'-a0'|	|b4-a1|	|b3'-a2'|	**|b3-a3|**	|b2'-a4'|	|b2-a5|	|b1'-a6'|	|b1-a7|
11	|b5-a0|	|b4'-a1'|	|b4-a2|	|b3'-a3'|	|b3-a4|	|b2'-a5'|	|b2-a6|	|b1'-a7'|
12	|b5'-a0'|	|b5-a1|	|b4'-a2'|	|b4-a3|	|b3'-a4'|	|b3-a5|	|b2'-a6'|	|b2-a7|
13	|b6-a0|	|b5'-a1'|	|b5-a2|	|b4'-a3'|	**|b4-a4|**	|b3'-a5'|	|b3-a6|	|b2'-a7'|
14	|b6'-a0'|	|b6-a1|	|b5'-a2'|	|b5-a3|	|b4'-a4'|	|b4-a5|	|b3'-a6'|	|b3-a7|
15	|b7-a0|	|b6'-a1'|	|b6-a2|	|b5'-a3'|	|b5-a4|	|b4'-a5'|	|b4-a6|	|b3'-a7'|
16	|b7'-a0'|	|b7-a1|	|b6'-a2'|	|b6-a3|	|b5'-a4'|	**|b5-a5|**	|b4'-a6'|	|b4-a7|

제 7도는 실시예의 제 1 변형에서의 코프로세서를 구조적으로 도시한다. 이러한 변형에서, 상기 코프로세서는 FIFO 스택(701 내지 710)을 포함하며, 데이타 버스(11)상에 병렬로 연결된다. 스택당 데이타 워드의 수는 검색 윈도우의 픽셀의 열의 수를 결정한다. 스택(701 내지 710)의 출력은 10개의 입력 및 8 개의 출력(714 내지 721)을 갖는 스위칭 회로(712)에 연결된다.

회로(712)는 10개의 FIFO 스택(이하 FIFO로 언급됨) 중 임의의 출력이 8개의 회로 출력 중 임의의 출력에 접속되도록 하여준다. 각 FIFO는 검색 윈도우의 완전한 라인을 포함한다. 주어진 시간에 출력에 연결되는 8개의 FIFO는 처리되는 검색 윈도우의 대역의 라인에 대응한다.

연결되지 않은 2개의 보조 FIFO는 메모리(10)에 저장된 데이타로 부터 데이타 버스를 통해 로드된다. 로드된 데이타는 현재 처리될 대역과 다른 검색 윈도우의 라인에 대응한다.

예를들어 검색 윈도우의 8개의 라인(L1 내지 L8)의 제 1 대역이 FIFO(701 내지 708)에 포함된다고 가정하자. 이들 8개의 라인을 처리하는 동안, 라인(L9)은 FIFO(709)에 로드된다. 제 1 대역의 처리가 완료되었을때, FIFO(701)에 기억된 라인(L1)은 필요없게 된다. 그에따라 제 2 대역은 라인(L2 내지 L9)을 포함한다. 제 2 대역의 처리는 유휴시간 없이 시작될 수 있는데 그 이유는 완전한 대역이 FIFO에 존재하기 때문이다. 스위칭 회로는 시스톨릭 네트워크(722 내지 725)에 제공되기 전에 기억된 라인의 바른 순서를 재생성해야만 한다.

대역이 8개의 라인을 갖는 이러한 경우, 9개의 FIFO는 동작을 설명하기에 충분하다. 제 10 FIFO는 예를들면 병렬인 4개 라인의 두 대역을 조정할 때 사용된다.

이러한 변형에서, 4개 라인 및 16열인 4개의 시스톨릭 네트워크(722 내지 725)는 각각 스위칭 회로의 출력에 연결된다. 규약에 의하여, 우리는 출력(714)은 8개 라인의 대역 중 상위 라인을 나타내는 반면, 출력(721)은 하위 라인을 나타낸다고 한다. 제 1 및 제 3 네트워크(각각 722 및 724)는 4개의 상위 출력(714 내지 717)에 연결되고, 반면 제 2 및 제 4 네트워크(각각 723 및 725)는 4개의 하위 네트워크(718 내지 721)에 연결된다.

상기의 장치는 다른 크기의 현재 블럭이 유동적으로 조절될 수 있도록 한다. 효과적으로, 우리는 다른 방법으로 4개의 네트워크;

- 8개의 4 ×16 네트워크, 각 4 × 16 네트워크는 2개의 현재 4 ×8 픽셀 블럭을 처리;
- 4개의 8 ×8 네트워크;
- 2개의 8 ×16 네트워크;
- 2개의 16 ×16 네트워크를 고려할 수 있다.

후자의 경우, 네트워크들은 비월 모드로 동작하고, 16개가 아니라 8개의 출력만 사용가능하다. 8개의 출력들은 기준 윈도우의 상위 대역 및 하위 대역의 8개 픽셀의 열을 번갈아 공급할 것이다.

물론 (다음 라인의 로딩을 위해 필요한 FIFO를 제외하고) 8개의 출력 대신 16개의 FIFO를 사용하는 것이 가능하다. 앞서 설명한 바와 같이, FIFO 스택의 용량은 기준 윈도우의 폭을 결정한다. 4 ×8 네트워크로 작업하는 경우, FIFO를 둘씩 묶거나 2배 길이의 픽셀의 라인과 동등한 각 FIFO 쌍에 저장하는 것이 가능하다.

제 7도는 또한 계산 결과들이 동작 모드에 따라 각 4 x 16 네트워크의 출력으로 묶이는 방법을 화살표 형식으로 설명한다.
제 8도는 실시예의 제 2 변형을 도시한다. 9개의 FIFO(기준(801)하에서 묶인)는 스위칭 회로(802)로 공급된다. 본 실시예에서 코프로세서는 8개의 4 ×8 시스톨릭 네트워크(803 내지 810)를 포함한다. 실제로, 제 8도에 연결된 2개의 4 × 8 네트워크는 제 7도의 4 ×16 네트워크와 동등이다. 제 8도는 8개의 4 ×8 현재 블럭이 병렬로 처리되는 경우 각 네트워크의 계산 결과의 라우팅을 더욱 분명하게 도시한다. 2개의 상위 네트워크(803 및 804)의 예를 살펴보자. 네트워크(803)의 출력에서 최종 결과(즉, 에러 함수 값)는 직접 네트워크(804)로 전송된다. 4 ×8픽셀의 현재 블럭 8개 또는 8 ×8픽셀의 현재 블럭 4개로 작업하는지에 따라서, 네트워크(803)의 출력에서의 결과는 계속적인 처리가 가능할수도 있고 그렇지 않을 수도 있다. 4 ×8 픽셀의 현재 블럭 8개로 작업하는 경우, 이 결과는 네트워크(803)에 의하여 처리된 현재 블럭의 이동량 벡터를 결정하기 위하여 필요하게 될 것이다.
이러한 변형에서, 셀(M)은 각 4 ×8 네트워크와 연결된다. 각 셀(M)은 네트워크의 한 출력에서 최소 에러함수값을 결정하고, 대응하는 이동 벡터의 좌표를 기억한다. 결과들은 특정 데이타 버스(811)를 통하여 전송된다.
네트워크(803, 804, 807, 및 808)의 출력은 각각 멀티플렉서의 입력에 연결되고, 그의 출력은 관련된 셀(M)의 입력에 연결된다. 이들 동일한 네트워크의 각 출력은 또한 그의 다른 입력상에서 바로 아래 네트워크의 출력을 수신한 2-입력 가산기의 입력에 연결된다. 네트워크(803, 804, 807, 및 808)에 대하여, 이들은 각각 네트워크(805, 806, 809, 및 810)의 출력이 된다.
프로세서가 8개의 현재 블럭상에서 작업할 때, 각 네트워크의 출력은 관련된 셀(M)에 직접 연결된다. 그 때 각 셀(M)은 그가 갖는 에러함수 및 이동량 벡터의 최소값을 결정한다. 이것은 8개의 개별적인 벡터를 생성한다 ·
프로세서가 8 ×8 픽셀의 4 블럭상에서 작업할 때, 멀티플렉서는 동일한 블럭상에서 작업한 2개의 중첩된 네트워크와 관련된 셀(M) 중 하나가 이들 두 네트워 크에 의하여 계산된 에러함수의 값의 합을 수신하기 위하여 스위치된다. 나중에는 이 합을 수신한 셀에 의하여 결정된 벡터만이 고려될 것이다.
멀티플렉서의 스위칭은 제 1도의 제어기(2)에 의해 제어된다.

(57) 청구의 범위

청구항 1.
블럭을 기초로 한 이미지 압축 시스템에서 블럭 비교에 의하여 이동량을 추정하는 장치에 있어서,
- 두 값들 사이의 에러함수를 계산하는 N×M 수단(S,403) ;
- 추정될 이동량을 갖는 블럭을 나타내는 값들을 저장하는 제 1 메모리 수단(406,407) ;
- 기준 윈도우의 값을 저장하는 제 2 메모리 수단(B) ;
- 에러 함수값을 계산하는 축적 수단(Add)을 포함하며, 상기 제 1 및 제 2 메모리 수단은 저장된 값을 상기 각각의 계산 수단(S,406)에 제공하도록 접속되며, 상기 제 1 메모리 수단은 두개의 픽셀값을 위한 메모리(406,407) 및 상기 에러 함 수를 계산하는 수단(403)으로 상기 값들 중 하나를 멀티플렉싱하는 멀티플렉서(404,405)를 포함하는 것을 특징으로 하는 장치.

청구항 2.
제 1항에 있어서, Nx의 계산 수단(S,403), N×M의 제 1 메모리 수단(406,407), N×M의 제 2 메모리 수단(B) 및 N라인의 계산 수단을 위해 에러 함수값들을 각각 축적하는 직렬 연결된 M 축적 수단을 더 포함하며, 상기 축적 수단의 출력은 메모리 수단에 의해 형성된 네트워크의 일 블록의 서치 윈도우의 전파에 관한 함수인 지연을 삽입하기 위해 다음 축적 수단의 입력에 접속되는 것을 특징으로 하는 장치.

청구항 3.
제 2항에 있어서, N×M의 제 2 메모리 수단(B)을 포함하며, 라인의 메모리 수단은 네트워크의 열들 사이의 값을 나타내는 서치 윈도우를 전송하기 위해 직렬 접속되는 것을 특징으로 하는 장치.

청구항 4.
제 3항에 있어서, 제 2 메모리 수단(B)의 출력은 에러 함수 계산 수단(S,403)의 입력에 접속되는 것을 특징으로 하는 장치.

청구항 5.
제 4항에 있어서, 각각의 에러 함수 계산 수단(S,403)은 계산 수단이 접속된 제 2 메모리 수단에 의해 출력값의 에러 함수와 추정될 이동량을 가진 블록을 나타내는 값을 계산하는 것을 특징으로 하는 장치.

청구항 6.
제 5항에 있어서, 상기 각각의 에러 함수 계산 수단(S,403)의 결과는 각각의 버퍼에 저장되는 것을 특징으로 하는 장치.

청구항 7.
제 1항 내지 제 6항 중 어느 한 항에 있어서, 상기 제 1 메모리 수단에 저장된 두개의 값(406,407)은 각각 단일 블록을 형성하는 낮은 서브 블록과 높은 서브 블록을 나타내며, 상기 제 2 메모리 수단의 네트워크에 제공된 값은 교번적으로(alternatively) 기준 윈도우의 낮은 서브 슬라이스 및 높은 서브 슬라이스(비월모드)를 나타내는 것을 특징으로 하는 장치.

청구항 8.

제 7항에 있어서, 두개의 서브 블록에 대한 에러 함수의 계산값의 합산 수단(302)을 더 포함하는 것을 특징으로 하는 장치.

청구항 9.

제 7항에 있어서, 비월 모드에서 계산 사이클의 추가 지연이 각 쌍의 축적 수단들 사이에 삽입되는 것을 특징으로 하는 장치.

청구항 10.

제 1항 내지 제 6항 중 어느 한 항에 있어서, 제 1 메모리 수단에 의해 저장된 두개의 값(406,407)은 두개의 서로 다른 블록 즉 제 1 블록 및 제 2 블록을 나타내며, 제 2 메모리 수단의 네트워크에 제공된 값들은 교번적으로 제 1 및 제 2 블록에 연관된 기준 윈도우를 나타내는 것을 특징으로 하는 장치.

청구항 11.

제 2항 내지 제 6항 중 어느 한 항에 있어서, 버퍼의 각 라인은 제 3 메모리 수단(701에서 710, 801)에 의해 기준 윈도우의 라인에 공급되는 것을 특징으로 하는 장치.

청구항 12.

제 11항에 있어서, 상기 장치는 적어도 N+1개의 제 3 메모리 수단을 포함하며, 상기 적어도 N+1개의 제 3 메모리 수단 중 N개가 N라인의 제 2 메모리 수단으로 공급될 때, 공급되지 않는 제 3 메모리 수단은 현재 공급되는 서치 윈도우의 슬라이스와 다음에 바로 공급될 서치 윈도우의 슬라이스의 차이를 나타내는 값으로 로딩되는 것을 특징으로 하는 장치.

청구항 13.

제 11항에 있어서, 기준 윈도우 라인을 저장하는 제 3 메모리 수단(701에서 710, 801)과 기준 윈도우값을 에러 함수 계산 수단으로 전송하는 제 2 메모리 수단 사이의 스위칭을 위한 스위칭 회로(712)를 더 포함하며, 상기 스위칭 수단은 적절하게 제 3 메모리 수단을 제 2 메모리 수단의 라인에 접속시키는 것을 특징으로 하는 장치.

청구항 14.

제 1항 내지 제 6항 중 어느 한 항의 장치를 포함하는 것을 특징으로 하는 이미지 압축 시스템.

청구항 15.

비교를 실행하기 위해 셀 매트릭스를 포함하는 블록 비교에 의해 이동량을 추정하는 이미지 압축 시스템에 있어서, 상기 각각의 셀들은:

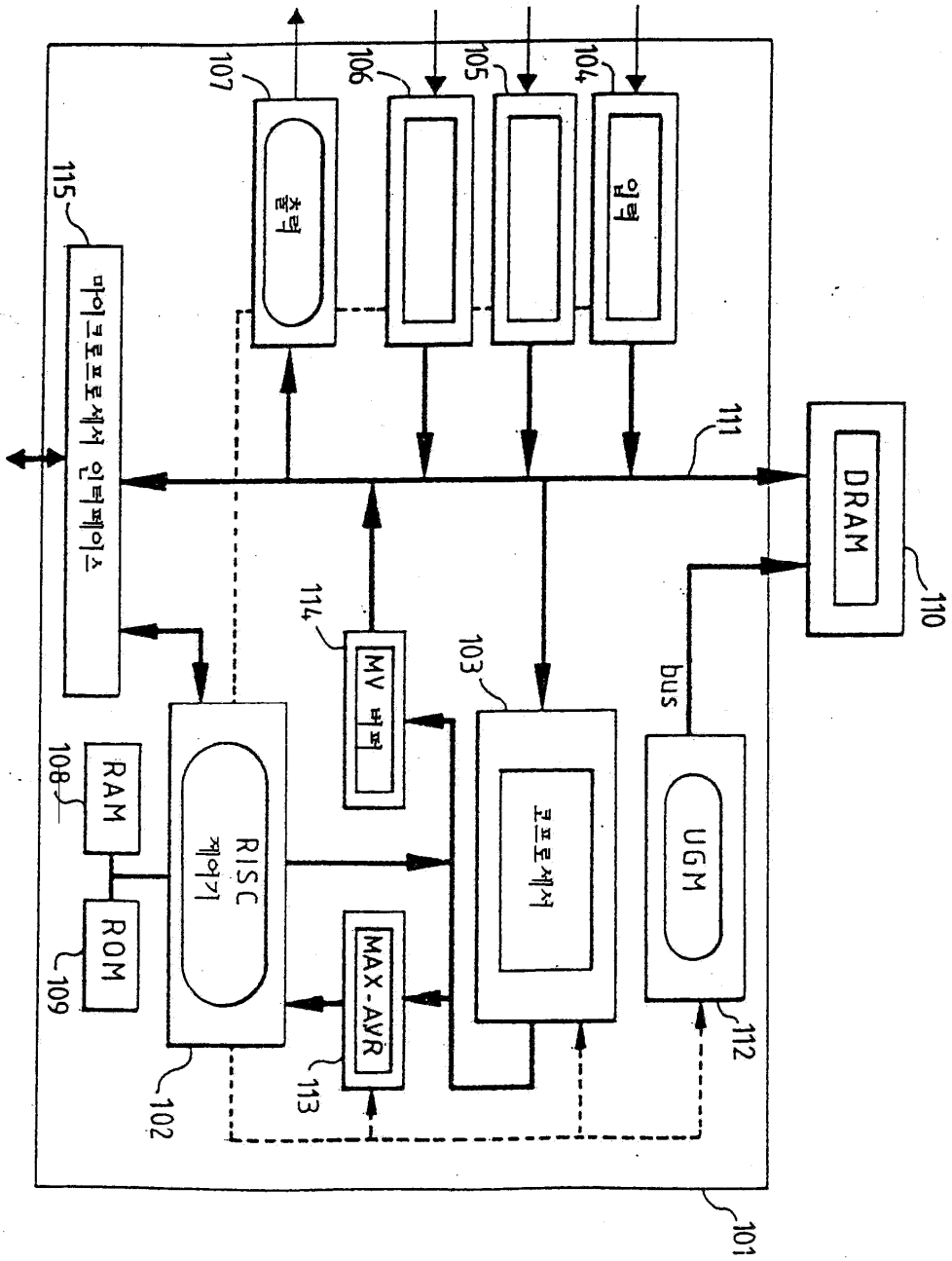
블록을 나타내는 값과 기준 윈도우를 나타내는 값 사이의 에러함수를 계산하는 수단(S,403);

이동량이 추정될 적어도 하나의 블록을 나타내는 다수의 픽셀값을 저장하는 메모리 수단(406.407) ; 및

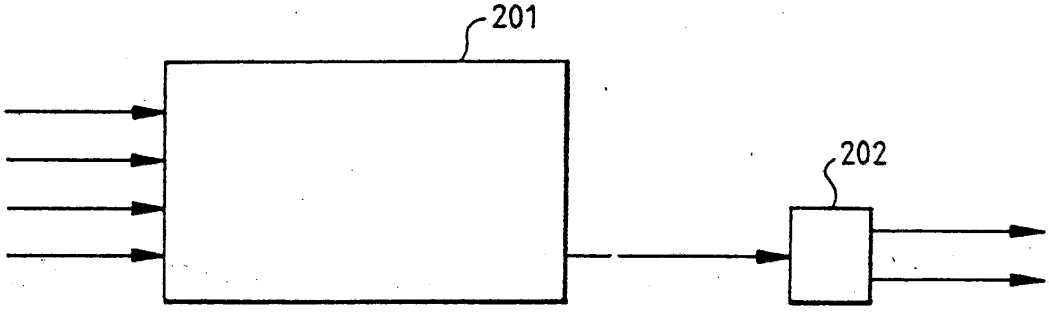
에러 함수 계산 수단(403)로 상기 다수의 값들 중 하나를 멀티플렉싱하는 상기 메모리 수단에 접속된 멀티플렉서(404,405)를 포함하는 것을 특징으로 하는 이미지 압축 시스템.

도면

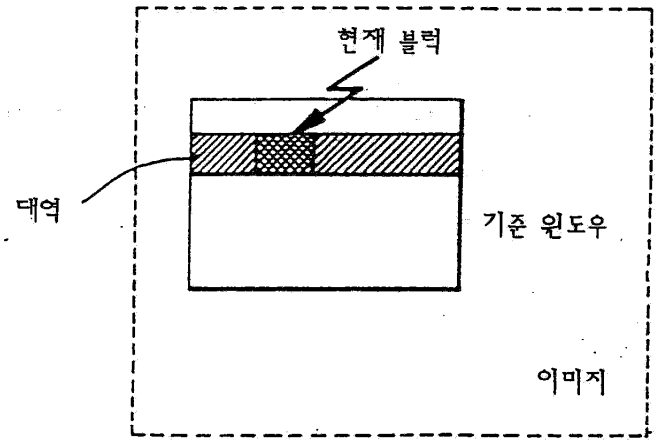
도면1


107
106
105
104
출력
입력
115
마이크로프로세서 인터페이스
111
DRAM
110
114
MV 버퍼
103
bus
108
RAM
RISC 제어기
프로세서
UGM
109
ROM
MAX-AVR
112
102
113
101

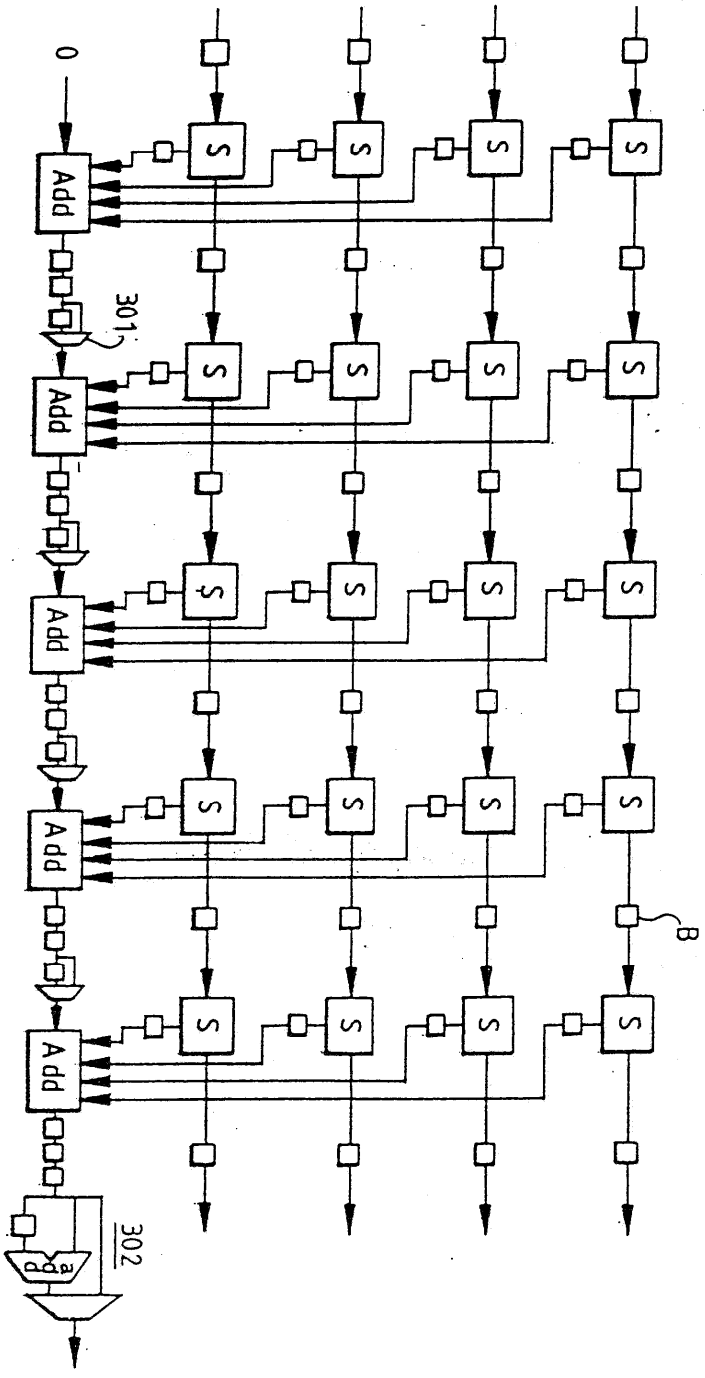
도면2a


201
202

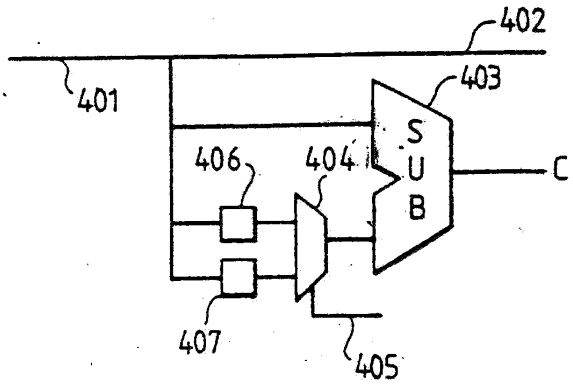
도면2b



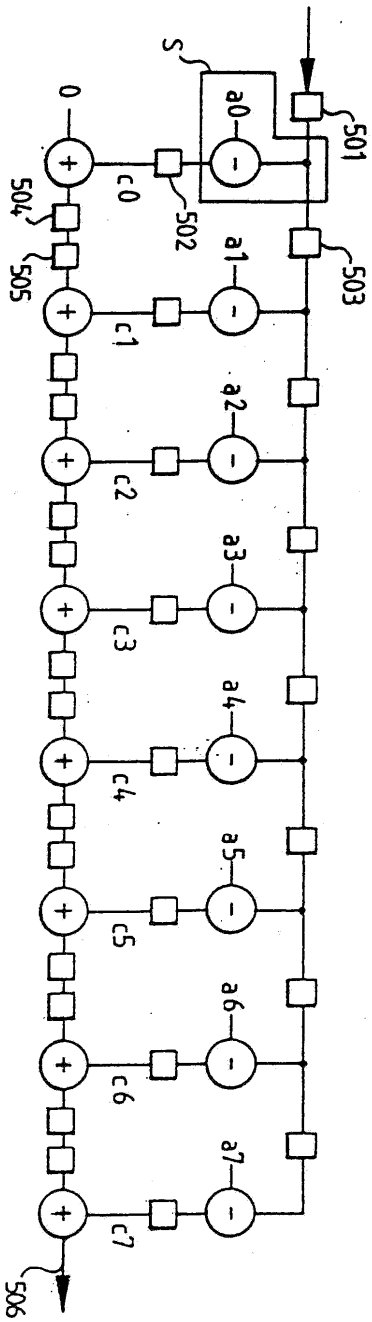
도면3



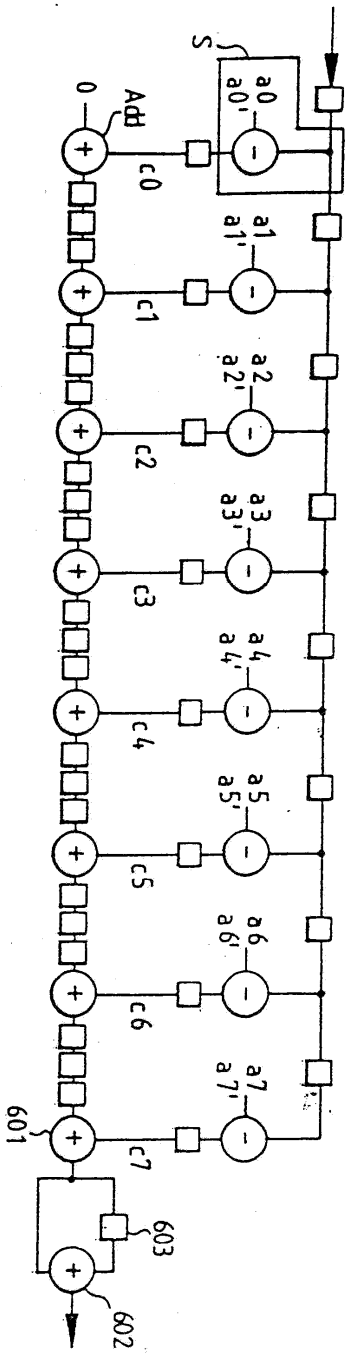
도면4


402
401
403
S
U
B
406
404
C
407
405

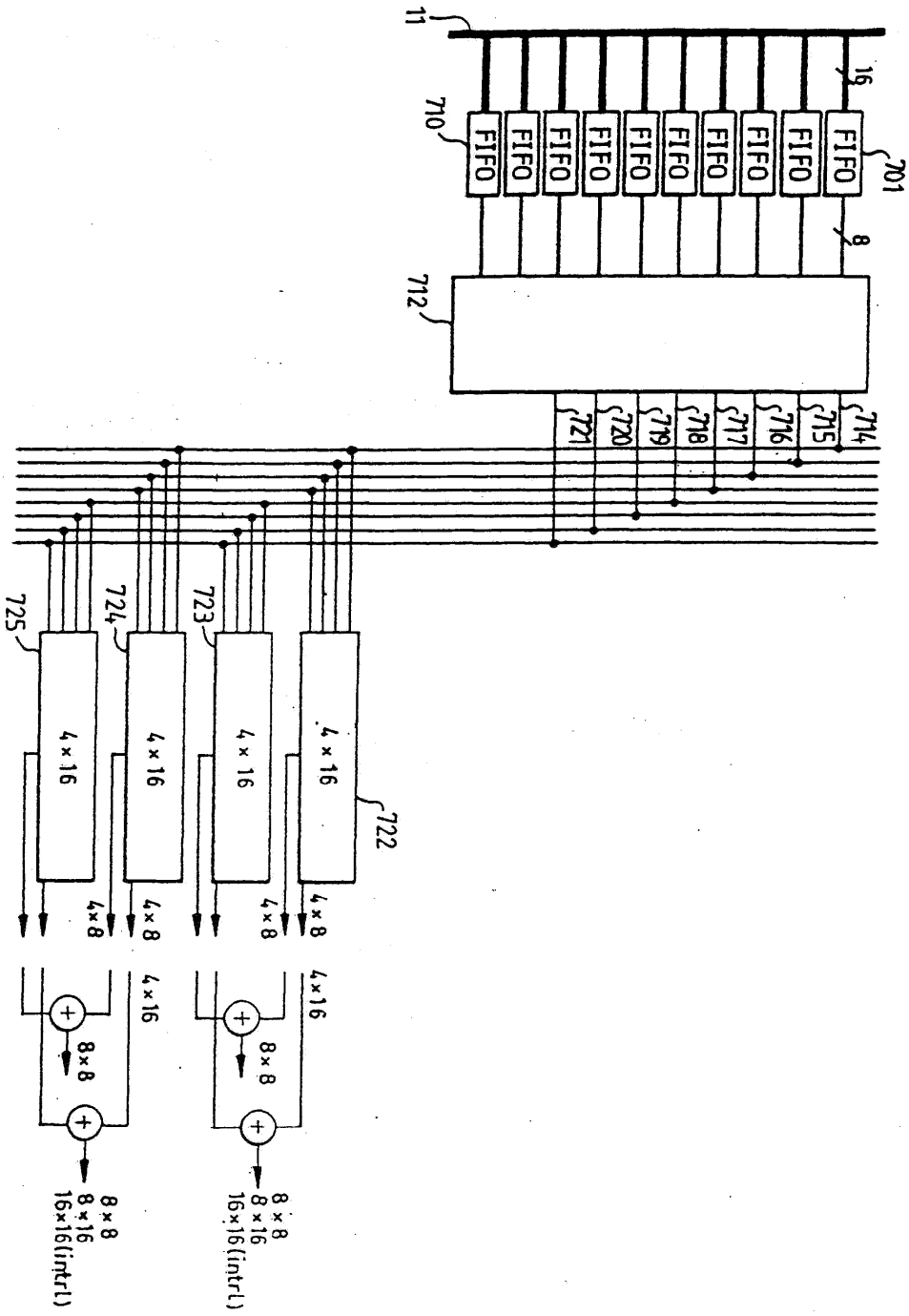
도면5


S
a0
501
0
c0
504
502
a1
503
505
c1
a2
c2
a3
c3
a4
c4
a5
c5
a6
c6
a7
c7
506

도면6


S
a0
a0'
0
Add
c0
a1
a1'
c1
a2
a2'
c2
a3
a3'
c3
a4
a4'
c4
a5
a5'
c5
a6
a6'
c6
a7
a7'
601
c7
603
602

도면7


11
16
710
FIFO
701
8
712
721
720
719
718
717
716
715
714
725
724
723
722
4×16
4×8
8×8
8×16
16×16(intrl)

도면8

