(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년02월14일
(11) 등록번호 10-1013344
(24) 등록일자 2011년01월31일

(51) Int. Cl.
H04N 7/30 (2006.01) *H04N 1/41* (2006.01)
G06F 17/14 (2006.01) *H03M 7/30* (2006.01)
(21) 출원번호 10-2008-7031503
(22) 출원일자(국제출원일자) 2007년06월25일
심사청구일자 2008년12월26일
(85) 번역문제출일자 2008년12월26일
(65) 공개번호 10-2009-0027670
(43) 공개일자 2009년03월17일
(86) 국제출원번호 PCT/US2007/072039
(87) 국제공개번호 WO 2008/002881
국제공개일자 2008년01월03일
(30) 우선권주장
60/816,697 2006년06월26일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
KR1019920702100 A
KR1019930701074 A
JP평성05030470 A
전체 청구항 수 : 총 84 항

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
가루다드리 해리나스
미국 92129 캘리포니아주 샌디에고 오비도 스트리트 9435
레즈니크 유리이
미국 92122 캘리포니아주 샌디에고 코스타 베르데 불러바드 8775 넘버319
(74) 대리인
특허법인코리아나

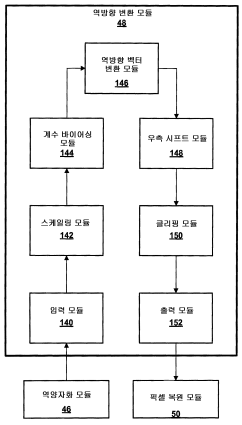
심사관 : 김영태

(54) 이산 코사인 변환의 계산 중의 에러 감소

(57) 요 약

고정 소수점 계산을 이용한 이산 코사인 변환의 계산 중에 라운딩 에러를 감소시키는 기술이 기재되어 있다. 이들 기술에 따르면, 이산 코사인 변환에서, 스케일링된 계수의 행렬은, 계수의 행렬에서의 계수와 스케일 인수를 승산함으로써 계산된다. 다음에, 중심점 바이어스 값 및 보완 바이어스 값이 스케일링된 계수의 행렬의 DC 계수에 가산된다. 다음에, 그 결과적인 스케일링된 계수의 행렬에 대해 역이산 코사인 변환이 적용된다. 그런 다음, 그 결과적인 행렬에서의 값이 우측-시프팅되어, 픽셀 성분값의 행렬을 도출한다. 본 명세서에 기재된 바와 같이, DC 계수에 대한 보완 바이어스 값의 가산은 이 우측-시프팅에 기인하는 라운딩 에러를 감소시킨다. 그 결과, 이들 기술을 이용하여 압축해제된 디지털 미디어 파일의 최종 버전은 디지털 미디어 파일의 오리지널 버전을 보다 근접하게 닮을 수도 있다.

대 표 도 - 도6



(30) 우선권주장
60/829,669 2006년10월16일 미국(US)
60/841,362 2006년08월30일 미국(US)
60/847,194 2006년09월25일 미국(US)
60/869,530 2006년12월11일 미국(US)

특허청구의 범위

청구항 1

소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 단계;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 단계;

상기 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 단계로서, 상기 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들인, 상기 출력 계수들의 행렬을 발생시키는 단계; 및

미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 단계를 포함하고,

상기 소스 계수들의 행렬의 일 계수에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 상기 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 2

제 1 항에 있어서,

상기 출력 계수들의 행렬에서의 계수들은 픽셀 성분값들이고,

상기 방법은,

상기 픽셀 성분값들을 포함하는 픽셀들의 블록을 구성하는 단계를 더 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 3

제 2 항에 있어서,

상기 미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 단계는, 상기 미디어 프리젠테이션 유닛으로 하여금 상기 픽셀들의 블록을 디스플레이하도록 하는 단계를 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 4

제 2 항에 있어서,

상기 픽셀들의 블록을 픽셀들의 다른 블록들과 결합함으로써, 제 1 비디오 프레임을 발생시키는 단계; 및

시간적 디코딩 (temporal decoding) 을 이용하여, 상기 제 1 비디오 프레임을 기준 비디오 프레임으로서 이용하는 하나 이상의 후속 비디오 프레임들을 발생시키는 단계를 더 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 5

제 1 항에 있어서,

상기 소스 계수들의 행렬의 일 계수는 상기 소스 계수들의 행렬의 DC 계수인, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 6

제 1 항에 있어서,

상기 소스 계수들의 행렬의 일 계수에 중심점 바이어스 값을 가산하는 단계를 더 포함하고,

상기 중심점 바이어스 값은 2^{P-1} 과 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 7

제 1 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 8

제 1 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 단계는, 상기 소스 계수들의 행렬의 일 계수에 보완 바이어스 값을 가산하는 단계를 포함하고,

상기 보완 바이어스 값은, 상기 소스 계수들의 행렬의 일 계수가 음수인 경우에는 -1 과 동등하고, 상기 소스 계수들의 행렬의 일 계수가 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 9

제 1 항에 있어서,

의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하는 단계를 더 포함하고,

상기 바이어싱된 계수들의 행렬을 발생시키는 단계는, 상기 소스 계수들의 행렬의 일 계수에 상기 선택된 값을 가산하는 단계를 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 10

제 1 항에 있어서,

양자화된 계수들의 행렬을 역양자화함으로써, 역양자화된 계수들의 행렬을 발생시키는 단계; 및

상기 변환된 계수들의 행렬을 발생시키는 단계 동안에 이용되는 고정 소수점 수들의 가수 비트들의 수만큼 상기 역양자화된 계수들의 행렬에서의 계수들을 좌측-시프팅시킴으로써, 상기 소스 계수들의 행렬을 발생시키는 단계를 더 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 11

제 10 항에 있어서,

상기 가수 비트들의 수는 3 이고,

상기 소스 계수들의 행렬이 주어지는 경우, 상기 출력 계수들의 행렬은 IEEE (Institute of Electrical and Electronics Engineers) 1180 표준의 정밀도 요건들을 만족시키는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 12

제 1 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 단계는, 승산 또는 제산 연산들을 이용하지 않으면서, 고정 소수점 연

산을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하는 단계를 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 13

제 1 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 단계 동안에 이용되는 고정 소수점 수들은, 16-비트 고정 소수점 수들인, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 14

제 1 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 단계는,

고정 소수점 연산을 이용하여 상기 바이어싱된 계수들의 행렬의 각 행 벡터에 대해 일련의 버터플라이 구조 연산들을 적용함으로써, 중간 계수들의 행렬을 발생시키는 단계; 및

고정 소수점 연산을 이용하여 상기 중간 계수들의 행렬의 각 열 벡터에 대해 상기 일련의 버터플라이 구조 연산들을 적용함으로써, 상기 변환된 계수들의 행렬을 발생시키는 단계를 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 15

제 14 항에 있어서,

상기 일련의 버터플라이 구조 연산들을 적용하는 것은,

$$u = (((x * C') + (y * S') + (1 << (k\text{-}1))) >> k;$$

$$v = ((x * S') + (y * -C') + (1 << (k\text{-}1))) >> k$$

의 형태의 버터플라이 구조 연산들을 적용하는 것을 포함하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 16

제 14 항에 있어서,

상기 버터플라이 구조 연산들 중 하나의 버터플라이 구조 연산에 의해 발생된 결과들과, 무한의 정밀도 연산을 이용하여 등가의 버터플라이 구조 연산에 의해 발생되는 결과들 사이의 차이들은, 0 을 중심으로 하며, 1 이하의 포지티브 크기 또는 네거티브 크기를 갖는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 17

제 16 항에 있어서,

상기 일련의 버터플라이 구조 연산들을 적용하는 것은,

$$u = ((x * C) >> k) - ((y * -S) >> k);$$

$$v = ((x * S) >> k) - ((y * C) >> k)$$

의 형태의 버터플라이 구조 연산들을 적용하는 것을 포함하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 18

소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 계수 바이어싱 모듈;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 역방향 변환 모듈;

상기 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 우측-시프트 모듈로서, 상기 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들인, 상기 우측-시프트 모듈; 및

미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 프리젠테이션 드라이버를 포함하고,

상기 소스 계수들의 행렬의 일 계수에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 상기 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 무선 통신 디바이스.

청구항 19

소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 계수 바이어싱 모듈;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 역방향 변환 모듈;

상기 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 우측-시프트 모듈로서, 상기 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들인, 상기 우측-시프트 모듈; 및

미디어 프리젠테이션 유닛으로 하여금, 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 프리젠테이션 드라이버를 포함하고,

상기 소스 계수들의 행렬의 일 계수에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 상기 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 20

제 19 항에 있어서,

상기 출력 계수들의 행렬에서의 계수들은 픽셀 성분값들이고,

상기 디바이스는,

상기 픽셀 성분값들을 포함하는 픽셀들의 블록을 구성하는 픽셀 복원 모듈을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 21

제 20 항에 있어서,

상기 프리젠테이션 드라이버는, 상기 미디어 프리젠테이션 유닛으로 하여금 상기 픽셀들의 블록을 디스플레이하도록 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 22

제 20 항에 있어서,

상기 픽셀들의 블록을 픽셀들의 다른 블록들과 결합함으로써, 제 1 비디오 프레임을 발생시키는 블록 결합기 모듈; 및

상기 제 1 비디오 프레임을 기준 비디오 프레임으로서 이용하는 하나 이상의 후속 비디오 프레임들을 발생시키는 예측 프레임 발생 모듈을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 23

제 19 항에 있어서,

상기 소스 계수들의 행렬의 일 계수는 상기 소스 계수들의 행렬의 DC 계수인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 24

제 19 항에 있어서,

상기 계수 바이어싱 모듈은, 상기 소스 계수들의 행렬의 일 계수에 중심점 바이어스 값을 가산함으로써 상기 바이어싱된 계수들의 행렬을 발생시키고,

상기 중심점 바이어스 값은 2^{P-1} 과 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 25

제 19 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 26

제 19 항에 있어서,

상기 계수 바이어싱 모듈은, 상기 소스 계수들의 행렬의 일 계수가 음수인 경우에는 -1 과 동등하며, 상기 소스 계수들의 행렬의 일 계수가 음수가 아닌 경우에는 0 과 동등한 보완 바이어스 값을 가산함으로써, 상기 바이어싱된 계수들의 행렬을 발생시키는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 27

제 19 항에 있어서,

상기 계수 바이어싱 모듈은, 의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하고, 상기 소스 계수들의 행렬의 일 계수에 상기 선택된 값을 보완 바이어스 값들로서 가산함으로써, 상기 바이어싱된 계수들의 행렬을 발생시키는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 28

제 19 항에 있어서,

양자화된 계수들의 행렬을 역양자화함으로써, 역양자화된 계수들의 행렬을 발생시키는 역양자화 모듈; 및

상기 변환된 계수들의 행렬을 발생시키는 동안에 이용되는 고정 소수점 수의 가수 비트들의 수만큼 상기 역양자화된 계수들의 행렬에서의 계수들을 좌측-시프팅시킴으로써, 상기 소스 계수들의 행렬을 발생시키는 스케일링 모듈을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 29

제 28 항에 있어서,

상기 가수 비트들의 수는 3 이고,

상기 소스 계수들의 행렬이 주어지는 경우, 상기 출력 계수들의 행렬은 IEEE (Institute of Electrical and Electronics Engineers) 1180 표준의 정밀도 요건들을 만족시키는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 30

제 19 항에 있어서,

상기 역방향 변환 모듈은, 고정 소수점 연산을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하는 경우에 16-비트 고정 소수점 수들을 이용하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 31

제 19 항에 있어서,

상기 역방향 변환 모듈은,

고정 소수점 연산을 이용하여, 상기 바이어싱된 계수들의 행렬의 각 행 벡터에 대해 일련의 버터플라이 구조 연산들을 적용함으로써, 중간 계수들의 행렬을 생성하고,

고정 소수점 연산을 이용하여, 상기 중간 계수들의 행렬의 각 열 벡터에 대해 상기 일련의 버터플라이 구조 연산들을 적용함으로써, 상기 변환된 계수들의 행렬을 생성하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 32

제 31 항에 있어서,

상기 버터플라이 구조 연산들은,

$$u = (((x * C') + (y * S') + (1 << (k\text{-}1))) >> k;$$

$$v = ((x * S') + (y * \text{-}C') + (1 << (k\text{-}1))) >> k$$

의 형태이고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 33

제 31 항에 있어서,

상기 버터플라이 구조 연산들 중 하나의 버터플라이 구조 연산에 의해 발생된 결과들과, 무한의 정밀도 연산을 이용하여 등가의 버터플라이 구조 연산에 의해 발생되는 결과들 사이의 차이들은, 0 을 중심으로 하며, 1 이하의 포지티브 크기 또는 네거티브 크기를 갖는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한

디바이스.

청구항 34

제 33 항에 있어서,

상기 버터플라이 구조 연산들은,

$$u = ((x * C) >> k) - ((y * -S) >> k);$$

$$v = ((x * S) >> k) - ((y * C) >> k)$$

의 형태이고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 35

소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 수단;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 수단;

상기 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 수단으로서, 상기 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들인, 상기 출력 계수들의 행렬을 발생시키는 수단; 및

미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 수단을 포함하고,

상기 소스 계수들의 행렬의 일 계수에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 상기 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 36

제 35 항에 있어서,

상기 출력 계수들의 행렬에서의 계수들은 픽셀 성분값들이고,

상기 디바이스는,

상기 픽셀 성분값들을 포함하는 픽셀들의 블록을 구성하는 수단을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 37

제 36 항에 있어서,

상기 미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 수단은, 상기 미디어 프리젠테이션 유닛으로 하여금 상기 픽셀들의 블록을 디스플레이하도록 하는 수단을 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 38

제 36 항에 있어서,

상기 픽셀들의 블록을 픽셀들의 다른 블록들과 결합함으로써, 제 1 비디오 프레임을 발생시키는 수단; 및

시간적 디코딩 (temporal decoding) 을 이용하여, 상기 제 1 비디오 프레임을 기준 비디오 프레임으로서 이용하는 하나 이상의 후속 비디오 프레임들을 발생시키는 수단을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 39

제 35 항에 있어서,

상기 소스 계수들의 행렬의 일 계수는 상기 소스 계수들의 행렬의 DC 계수인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 40

제 35 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 수단은, 상기 소스 계수들의 행렬의 일 계수에 중심점 바이어스 값을 가산하는 수단을 포함하고,

상기 중심점 바이어스 값은 2^{P-1} 과 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 41

제 40 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 42

제 35 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 수단은, 상기 소스 계수들의 행렬의 일 계수에 보완 바이어스 값을 가산하는 수단을 포함하고,

상기 보완 바이어스 값은, 상기 소스 계수들의 행렬의 일 계수가 음수인 경우에는 -1 과 동등하고, 상기 소스 계수들의 행렬의 일 계수가 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 43

제 35 항에 있어서,

의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하는 수단을 더 포함하고,

상기 바이어싱된 계수들의 행렬을 발생시키는 수단은, 상기 소스 계수들의 행렬의 일 계수에 상기 선택된 값을 가산하는 수단을 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 44

제 35 항에 있어서,

양자화된 계수들의 행렬을 역양자화함으로써, 역양자화된 계수들의 행렬을 발생시키는 수단; 및

상기 변환된 계수들의 행렬을 발생시키는 동안에 이용되는 고정 소수점 수들의 가수 비트들의 수만큼 상기 역양자화된 계수들의 행렬에서의 계수들을 좌측-시프팅시킴으로써, 상기 소스 계수들의 행렬을 발생시키는 수단을 더 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 45

제 44 항에 있어서,

상기 가수 비트들의 수는 3 이고,

상기 소스 계수들의 행렬이 주어지는 경우, 상기 출력 계수들의 행렬은 IEEE (Institute of Electrical and Electronics Engineers) 1180 표준의 정밀도 요건들을 만족시키는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 46

제 35 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 수단은, 승산 또는 제산 연산들을 이용하지 않으면서, 고정 소수점 연산을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하는 수단을 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 47

제 35 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 수단은, 고정 소수점 연산을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하는 경우에 16-비트 고정 소수점 수들을 이용하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 48

제 35 항에 있어서,

상기 변환된 계수들의 행렬을 발생시키는 수단은,

고정 소수점 연산을 이용하여 상기 바이어싱된 계수들의 행렬의 각 행 벡터에 대해 일련의 버터플라이 구조 연산들을 적용함으로써, 중간 계수들의 행렬을 발생시키는 수단; 및

고정 소수점 연산을 이용하여 상기 중간 계수들의 행렬의 각 열 벡터에 대해 상기 일련의 버터플라이 구조 연산들을 적용함으로써, 상기 변환된 계수들의 행렬을 발생시키는 수단을 포함하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 49

제 48 항에 있어서,

상기 일련의 버터플라이 구조 연산들을 적용하는 것은,

$$u = (((x * C') + (y * S') + (1 << (k\text{-}1))) >> k;$$

$$v = ((x * S') + (y * -C') + (1 << (k\text{-}1))) >> k$$

의 형태의 버터플라이 구조 연산들을 적용하는 것을 포함하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 50

제 48 항에 있어서,

상기 버터플라이 구조 연산들 중 하나의 버터플라이 구조 연산에 의해 발생된 결과들과, 무한의 정밀도 연산을 이용하여 등가의 버터플라이 구조 연산에 의해 발생되는 결과들 사이의 차이들은, 0 을 중심으로 하며, 1 이하의 포지티브 크기 또는 네거티브 크기를 갖는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 51

제 50 항에 있어서,

상기 일련의 버터플라이 구조 연산들을 적용하는 것은,

$$u = ((x * C) >> k) - ((y * -S) >> k);$$

$$v = ((x * S) >> k) - ((y * C) >> k)$$

의 형태의 버터플라이 구조 연산들을 적용하는 것을 포함하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 52

이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서,

실행되는 경우, 상기 명령들은, 프로세서가,

소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키게 하고;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키게 하고;

상기 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키게 하고;

미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하게 하고,

상기 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이고,

상기 소스 계수들의 행렬의 일 계수에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 상기 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 53

제 52 항에 있어서,

상기 출력 계수들의 행렬에서의 계수들은 픽셀 성분값들이고,

상기 명령들은 또한 상기 프로세서가, 상기 픽셀 성분값들을 포함하는 픽셀들의 블록을 구성하게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 54

제 53 항에 있어서,

상기 명령들은, 상기 프로세서가 상기 미디어 프리젠테이션 유닛으로 하여금 상기 픽셀들의 블록을 디스플레이하도록 하게 함으로써, 상기 프로세서가 상기 미디어 프리젠테이션 유닛으로 하여금 상기 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 55

제 53 항에 있어서,

상기 명령들은 또한 상기 프로세서가,

상기 픽셀들의 블록을 픽셀들의 다른 블록들과 결합함으로써, 제 1 비디오 프레임을 발생시키게 하고;

시간적 디코딩 (temporal decoding) 을 이용하여, 상기 제 1 비디오 프레임을 기준 비디오 프레임으로서 이용하는 하나 이상의 후속 비디오 프레임들을 발생시키게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 56

제 52 항에 있어서,

상기 소스 계수들의 행렬의 일 계수는 상기 소스 계수들의 행렬의 DC 계수인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 57

제 52 항에 있어서,

상기 명령들은 또한 상기 프로세서가, 상기 소스 계수들의 행렬의 일 계수에 중심점 바이어스 값을 가산하게 하고,

상기 중심점 바이어스 값은 2^{P-1} 과 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 58

제 52 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 59

제 52 항에 있어서,

상기 프로세서가 상기 바이어싱된 계수들의 행렬을 발생시키게 하는 명령들은, 상기 프로세서가 상기 소스 계수들의 행렬의 일 계수에 보완 바이어스 값을 가산하게 하고,

상기 보완 바이어스 값은, 상기 소스 계수들의 행렬의 일 계수가 음수인 경우에는 -1 과 동등하고, 상기 소스 계수들의 행렬의 일 계수가 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 60

제 52 항에 있어서,

상기 명령들은 또한 상기 프로세서가, 의사 랜덤에 기초하여 -1 또는 0 중 어느 하나와 동등한 값을 선택하게 하고,

상기 프로세서가 상기 바이어싱된 계수들의 행렬을 발생시키게 하는 명령들은, 상기 프로세서가 상기 소스 계수들의 행렬의 일 계수에 상기 선택된 값을 가산하게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 61

제 52 항에 있어서,

상기 명령들은 또한 상기 프로세서가,

양자화된 계수들의 행렬을 역양자화함으로써, 역양자화된 계수들의 행렬을 발생시키게 하고;

상기 변환된 계수들의 행렬을 발생시키는 동안에 이용되는 고정 소수점 수들의 가수 비트들의 수만큼 상기 역양자화된 계수들의 행렬에서의 계수들을 좌측-시프팅시킴으로써, 상기 소스 계수들의 행렬을 발생시키게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 62

제 61 항에 있어서,

상기 가수 비트들의 수는 3 이고,

상기 소스 계수들의 행렬이 주어지는 경우, 상기 출력 계수들의 행렬은 IEEE (Institute of Electrical and Electronics Engineers) 1180 표준의 정밀도 요건들을 만족시키는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 63

제 52 항에 있어서,

상기 명령들은, 상기 프로세서가, 승산 또는 제산 연산들을 이용하지 않으면서, 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하게 함으로써, 상기 프로세서가 상기 변환된 계수들의 행렬을 발생시키게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 64

제 52 항에 있어서,

상기 명령들이, 상기 프로세서가 고정 소수점 연산을 이용하여 반복적으로 상기 바이어싱된 계수들의 행렬에서의 계수들에 상기 1-차원 변환을 적용하게 하는 경우에, 상기 명령들은, 상기 프로세서가 16-비트 고정 소수점 수들을 이용하게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 65

제 52 항에 있어서,

상기 명령들은, 상기 프로세서가,

고정 소수점 연산을 이용하여 상기 바이어싱된 계수들의 행렬의 각 행 벡터에 대해 일련의 버터플라이 구조 연산들을 적용함으로써, 중간 계수들의 행렬을 발생시키게 하고;

고정 소수점 연산을 이용하여 상기 중간 계수들의 행렬의 각 열 벡터에 대해 상기 일련의 버터플라이 구조 연산들을 적용함으로써, 상기 변환된 계수들의 행렬을 발생시키게 함으로써, 상기 프로세서가 상기 변환된 계수들의

행렬을 발생시키게 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 66

제 65 항에 있어서,

상기 명령들은, 상기 프로세서가,

$$u = (((x * C') + (y * S') + (1 << (k\text{-}1))) >> k;$$

$$v = ((x * S') + (y * -C') + (1 << (k\text{-}1))) >> k$$

의 형태의 버터플라이 구조 연산들을 적용하게 함으로써, 상기 프로세서가 상기 일련의 버터플라이 구조 연산들을 적용하게 하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 67

제 65 항에 있어서,

상기 버터플라이 구조 연산들 중 하나의 버터플라이 구조 연산에 의해 발생된 결과들과, 무한의 정밀도 연산을 이용하여 등가의 버터플라이 구조 연산에 의해 발생되는 결과들 사이의 차이들은, 0 을 중심으로 하며, 1 이하의 포지티브 크기 또는 네거티브 크기를 갖는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 68

제 66 항에 있어서,

상기 명령들은, 상기 프로세서가,

$$u = ((x * C) >> k) - ((y * -S) >> k);$$

$$v = ((x * S) >> k) - ((y * C) >> k)$$

의 형태의 버터플라이 구조 연산들을 적용하게 함으로써, 상기 프로세서가 상기 일련의 버터플라이 구조 연산들을 적용하게 하고,

여기서, u, v, x 및 y 는 고정 소수점 수들이고,

x 및 y 는 입력값들이고, u 및 v 는 출력값들이고,

C, S 및 k 는 정수들인, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 69

소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 단계;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 단계;

상기 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 단계;

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 단계; 및

제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 단계를 포함하고,

상기 소스 계수들의 행렬은 미디어 데이터를 나타내고,

상기 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이고,

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 상기 제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 70

제 69 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 71

제 69 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 단계는, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 보완 바이어스 값을 가산하는 단계를 포함하고,

상기 보완 바이어스 값은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수인 경우에는 -1 과 동등하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 72

제 69 항에 있어서,

의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하는 단계를 더 포함하고,

상기 바이어싱된 계수들의 행렬을 발생시키는 단계는, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 상기 선택된 값을 가산하는 단계를 포함하는, 이산 코사인 변환의 계산 중의 에러 감소 방법.

청구항 73

소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 좌측-시프트 모듈;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 순방향 변환 모듈;

상기 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 스케일링 모듈;

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 계수 바이어싱 모듈; 및

제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 우측-시프트 모듈을 포함하고,

상기 소스 계수들의 행렬은 미디어 데이터를 나타내고,

상기 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이고,

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 상기 제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 74

제 73 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 75

제 73 항에 있어서,

상기 계수 바이어싱 모듈은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 보완 바이어스 값을 가산함으로써, 상기 바이어싱된 계수들의 행렬을 발생시키고,

상기 보완 바이어스 값은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수인 경우에는 -1 과 동등하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 76

제 73 항에 있어서,

상기 계수 바이어싱 모듈은, 의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 상기 선택된 값을 가산하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 77

소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 수단;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 수단;

상기 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 수단;

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 수단; 및

제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 수단을 포함하고,

상기 소스 계수들의 행렬은 미디어 데이터를 나타내고,

상기 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이고,

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 상기 하나 이상의 바이어스 값들의 가산은,

포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 상기 제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 78

제 77 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 79

제 77 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 수단은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 보완 바이어스 값을 가산함으로써, 상기 바이어싱된 계수들의 행렬을 발생시키고,

상기 보완 바이어스 값은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수인 경우에는 -1 과 동등하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 80

제 77 항에 있어서,

상기 바이어싱된 계수들의 행렬을 발생시키는 수단은, 의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 상기 선택된 값을 가산하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 디바이스.

청구항 81

이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서,

상기 명령들은, 프로그램가능한 프로세서로 하여금,

소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키도록 하고;

고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 상기 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키도록 하고;

상기 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키도록 하고;

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키도록 하고;

제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키도록 하고,

상기 소스 계수들의 행렬은 미디어 데이터를 나타내고,

상기 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 상기 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이고,

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 상기 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해

대칭이 되도록 하고,

상기 포지티브 에러들 및 네거티브 에러들은, 상기 제 1 크기만큼 상기 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{상기 제 1 크기})}$ 로 상기 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타내는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 82

제 81 항에 있어서,

상기 제 1 크기는 2^{P} 와 동등하고, 여기서 P 는, 상기 1-차원 변환에 이용된 고정 소수점 수들에서의 가수 비트들의 수에 상기 1-차원 변환을 적용함으로써 상기 변환된 계수들의 행렬에서의 계수들에 부가되는 비트들의 수를 더한 것과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 83

제 81 항에 있어서,

상기 명령들은, 상기 프로그램가능한 프로세서로 하여금 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 보완 바이어스 값을 가산하도록 함으로써, 상기 프로그램가능한 프로세서로 하여금 상기 바이어싱된 계수들의 행렬을 발생시키도록 하고,

상기 보완 바이어스 값은, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수인 경우에는 -1 과 동등하고, 상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들이 음수가 아닌 경우에는 0 과 동등한, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

청구항 84

제 81 항에 있어서,

상기 명령들은, 상기 프로그램가능한 프로세서로 하여금,

의사 랜덤에 기초하여, -1 또는 0 중 어느 하나와 동등한 값을 선택하도록 하고;

상기 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 상기 선택된 값을 가산하도록 함으로써, 상기 프로그램가능한 프로세서로 하여금 상기 바이어싱된 계수들의 행렬을 발생시키도록 하는, 이산 코사인 변환의 계산 중의 에러를 감소시키기 위한 명령들을 포함하는 컴퓨터 판독가능 매체.

명 세 서

[0001] 본 출원은, 2006 년 6 월 26 일자로 출원된 미국가출원 제 60/816,697 호, 2006 년 8 월 30 일자로 출원된 미국가출원 제 60/841,362 호, 2006 년 9 월 25 일자로 출원된 미국가출원 제 60/847,194 호, 2006 년 10 월 16 일자로 출원된 미국가출원 제 60/829,669 호, 및 2006 년 12 월 11 일자로 출원된 미국가출원 제 60/869,530 호의 이점을 주장하는데, 이들의 전체 내용은 본 명세서에 참조로서 포함되어 있다.

[0002] **기술분야**

[0003] 본 개시물은 컴퓨터 그래픽스 및 멀티미디어에 관한 것이고, 특히 그래픽스, 이미지, 및 비디오 정보의 압축에 관한 것이다.

[0004] **배경기술**

[0005] 고해상도 이미지 및 비디오가 비교적 콤팩트한 파일 또는 데이터 스트림으로서 저장 또는 송신되는 것을 허용하

기 위해서, 다수의 기존의 이미지 및 비디오 코딩 표준은 압축 기술을 이용한다. 이러한 코딩 표준은, JPEG (Joint Photographic Experts Group), MPEG (Moving Pictures Experts Group)-1, MPEG-2, MPEG-4 파트 2, H.261, H.263, 및 다른 이미지 또는 비디오 코딩 표준을 포함한다.

[0006] 다수의 이들 표준에 따르면, 비디오 프레임은 "공간적 (spatial)" 인코딩을 이용하여 압축된다. 이들 프레임은 오리지널 프레임 (즉, i-프레임) 일 수도 있고, 또는 모션 보상을 이용하는 시간적 (temporal) 인코딩 프로세스에 의해 발생된 잔여 프레임 (residual frame) 일 수도 있다. 공간적 인코딩 중에, 프레임은 동등한 크기의 픽셀 블록으로 쪼개진다. 예를 들어, 압축되지 않은 프레임은 8×8 픽셀 블록의 세트로 쪼개질 수도 있다. 각 픽셀 블록에 있어서, 픽셀 성분은 픽셀 성분값의 행렬로 분리된다. 예를 들어, 각 픽셀 블록은 Y 픽셀 성분값의 행렬, U 픽셀 성분값의 행렬, 및 V 픽셀 성분값의 행렬로 분할될 수도 있다. 이 실시예에 있어서, Y 픽셀 성분값은 휘도 (luminance) 값을 나타내고, U 픽셀 성분값 및 V 픽셀 성분값은 색차 (chrominance) 값을 나타낸다.

[0007] 또한, 공간적 인코딩 중에, 인코딩되고 있는 프레임에서의 픽셀 성분값의 행렬 각각에 대해 순이산 코사인 변환 (FDCT) 이 적용된다. 이상적인 1-차원 FDCT 는:

[0008] $$t(k) = c(k)\sum_{n=0}^{N-1} s(n)\cos\frac{\pi(2n+1)k)}{2N}$$

[0009] 으로 정의되고, 여기서 s 는 N 개의 오리지널 값의 어레이이고, t 는 N 개의 변환된 값의 어레이이고, 계수 c 는,

[0010] $$c(0) = \sqrt{1/N}, c(k) = \sqrt{2/N}$$ (여기서, 1 ≤ k ≤ N-1)

[0011] 로 주어진다.

[0012] 이상적인 2-차원 FDCT 는 공식:

[0013] $$t(i,j) = c(i,j)\sum_{n=1}^{N-1}\sum_{m=0}^{N-1} s(m,n)\cos\frac{\pi(2m+1)i}{2N}\cos\frac{\pi(2n+1)j}{2N}$$

[0014] 으로 정의되고, 여기서 s 는 N 개의 오리지널 값의 어레이이고, t 는 N 개의 변환된 값의 어레이이고, $c(i,j)$ 는 $c(i,j) = c(i)c(j)$ 로 주어지고, c(k) 는 1-차원 경우에서와 같이 정의된다.

[0015] 계수의 행렬은, 픽셀 성분값의 블록이 FDCT 를 이용하여 변환되는 경우에 생성된다. 그런 다음, 이 계수의 행렬은 양자화되고, 예를 들어 허프만 코드 또는 산술 코드 (arithmetic code) 를 이용하여 인코딩될 수도 있다. 비디오 비트스트림은, 압축되지 않은 일련의 비디오 프레임에서의 일련의 비디오 프레임의 픽셀 성분값의 블록 모두에 대해 이 프로세스를 수행한 합성 결과를 나타낸다.

[0016] 압축되지 않은 비디오 프레임은, 이 프로세스를 반전시킴으로써 비디오 비트스트림으로부터 도출될 수도 있다. 특히, 이 비트스트림에서의 계수의 행렬 각각은 압축해제되고, 압축해제된 값은 역양자화되어, 변환된 계수의 행렬을 도출한다. 그런 다음, 픽셀 성분값의 행렬을 도출하기 위해서 변환된 계수의 행렬 각각에 대해 역이산 코사인 변환 ("IDCT") 이 적용된다. 이상적인 1-차원 IDCT 는:

[0017] $$s(n) = \sum_{k=0}^{N-1} c(k)t(k)\cos\frac{\pi(2n+1)k}{2N}$$

[0018] 으로 정의되고, 여기서 s 는 N 개의 오리지널 값의 어레이이고, t 는 N 개의 변환된 값의 어레이이고, 계수 c 는,

[0019] $$c(0) = \sqrt{1/N}, c(k) = \sqrt{2/N}$$ (여기서, 1 ≤ k ≤ N-1)

[0020] 로 주어진다.

[0021] 이상적인 2-차원 IDCT 는 다음의 공식으로 정의된다:

[0022] $$s(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}c(i,j)t(i,j)\cos\frac{\pi(2m+1)i}{2N}\cos\frac{\pi(2n+1)j}{2N}$$.

[0023] 그런 다음, 그 결과적인 픽셀 성분값의 행렬은 픽셀 블록으로 재조립되고, 이들 픽셀 블록은 재조립되어, 디코딩된 프레임을 형성한다. 디코딩된 프레임이 i-프레임인 경우에는, 이 프레임은 현재 완전히 디코딩된다. 그러나, 압축되지 않은 프레임이 예측 프레임 (predictive frame) 또는 양방향-예측 프레임 (bi-predictive frame) 인 경우에는, 디코딩된 프레임은 단지 디코딩된 잔여 프레임이다. 디코딩된 프레임과 연관된 모션 벡터를 이용하여 복원된 프레임을 구성한 다음, 이 복원된 프레임을 디코딩된 잔여 프레임에 부가함으로써, 완성된 프레임이 발생된다.

[0024] 이상적인 환경 하에서는, FDCT 를 이용하여 픽셀 성분값의 블록을 인코딩하거나 IDCT 를 이용하여 픽셀 성분값의 블록을 디코딩함으로써 정보가 손실되지 않는다. 그 결과, 이들 이상적인 환경 하에서, 비디오 프레임의 디코딩된 버전은 비디오 프레임의 오리지널 버전과 동일하다. 그러나, FDCT 또는 IDCT 의 컴퓨팅은 계산상 어려울 수도 있는데, 그 이유는 FDCT 및 IDCT 의 계산이 실수 및 상당수의 승산 연산의 이용을 수반하기 때문이다. 이러한 이유로, FDCT 및 IDCT 에 이용된 실수는 제한된 정밀도의 수 (limited precision number) 를 이용하여 빈번하게 근사된다. 제한된 정밀도의 수를 이용하여 실수값을 나타내는 것으로부터 라운딩 에러가 기인한다. 또한, 양자화 및 역양자화가 부가적인 에러에 기여할 수도 있다.

[0025] 압축 및 압축해제 프로세스에서의 에러는, 압축되지 않은 오리지널 프레임과 압축되지 않은 최종 프레임 사이에 현저한 차이를 야기할 수도 있다. 예를 들어, 압축되지 않은 최종 프레임에서의 컬러는 압축되지 않은 오리지널 프레임에서의 컬러와 상이할 수도 있다. 또한, 인코더의 IDCT 구현과 디코더의 IDCT 구현 사이의 불일치에 의해 야기된 에러는 예측된 프레임의 시퀀스의 인코딩 및 디코딩 중에 누산될 수도 있다. 이들 누산된 에러는 통상적으로 "IDCT 드리프트" 로 언급된다.

[0026] ## 개 요

[0027] 고정 소수점 계산을 이용한 이산 코사인 변환의 계산 중에 라운딩 에러를 감소시키는 기술이 기재되어 있다. 이들 기술에 따르면, 스케일링된 계수의 행렬은, 계수의 행렬에서의 계수와 스케일 인수 (scale factor) 를 승산함으로써 계산된다. 다음에, 중심점 바이어스 값 (midpoint bias value) 및 보완 바이어스 값 (supplemental bias value) 이 스케일링된 계수의 행렬의 DC 계수에 가산된다. 다음에, 그 결과적인 스케일링된 계수의 행렬에 대해 역이산 코사인 변환이 적용된다. 그런 다음, 그 결과적인 행렬에서의 값이 우측-시프팅되어, 픽셀 성분값의 행렬을 도출한다. 본 명세서에 기재된 바와 같이, DC 계수에 대한 보완 바이어스 값의 가산은 이 우측-시프팅에 기인하는 라운딩 에러를 감소시킨다. 그 결과, 이들 기술을 이용하여 압축해제된 디지털 미디어 파일의 최종 버전은 디지털 미디어 파일의 오리지널 버전을 보다 근접하게 닮을 수도 있다.

[0028] 일 양태에 있어서, 일 방법은, 소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 고정 소수점 연산 (fixed-point arithmetic) 을 이용하여 반복적으로 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 단계를 포함한다. 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 또한, 이 방법은, 미디어 프리젠테이션 유닛으로 하여금 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 단계를 포함한다. 소스 계수들의 행렬의 일 계수에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0029] 또다른 양태에 있어서, 일 디바이스는, 소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 계수 바이어싱 모듈을 포함한다. 또한, 이 디바이스는, 고정

소수점 연산을 이용하여 반복적으로 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 역방향 변환 모듈을 포함한다. 또한, 이 디바이스는, 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 우측-시프트 모듈을 포함한다. 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 또한, 이 디바이스는, 미디어 프리젠테이션 유닛으로 하여금 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 프리젠테이션 드라이버를 포함한다. 소스 계수들의 행렬의 일 계수에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0030] 또다른 양태에 있어서, 일 디바이스는, 소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 고정 소수점 연산을 이용하여 반복적으로 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 수단을 포함한다. 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 또한, 이 디바이스는, 미디어 프리젠테이션 유닛으로 하여금 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 하는 수단을 포함한다. 소스 계수들의 행렬의 일 계수에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0031] 또다른 양태에 있어서, 본 발명은 명령들을 포함하는 컴퓨터 판독가능 매체에 관한 것이다. 실행시, 명령들은, 프로세서가, 소스 계수들의 행렬의 일 계수에 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키게 한다. 또한, 명령들은, 프로세서가, 고정 소수점 연산을 이용하여 반복적으로 바이어싱된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키게 한다. 또한, 명령들은, 프로세서가, 변환된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키게 한다. 출력 계수들의 행렬에서의 계수들은, 이상적인 역이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 또한, 명령들은, 미디어 프리젠테이션 유닛으로 하여금 출력 계수들의 행렬에 기초하여 가청 또는 가시 신호들을 출력하도록 한다. 소스 계수들의 행렬의 일 계수에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 변환된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 변환된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0032] 또다른 양태에 있어서, 일 방법은, 소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 고정 소수점 연산을 이용하여 반복적으로 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 단계를 포함한다. 또한, 이 방법은, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 단계를 포함한다. 소스 계수들의 행렬은 미디어 데이터를 나타낸다. 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하

며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0033] 또다른 양태에 있어서, 일 디바이스는, 소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 좌측-시프트 모듈을 포함한다. 또한, 이 디바이스는, 고정 소수점 연산을 이용하여 반복적으로 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 순방향 변환 모듈을 포함한다. 또한, 이 디바이스는, 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 스케일링 모듈을 포함한다. 또한, 이 디바이스는, 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 계수 바이어싱 모듈을 포함한다. 또한, 이 디바이스는, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 우측-시프트 모듈을 포함한다. 소스 계수들의 행렬은 미디어 데이터를 나타낸다. 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0034] 또다른 양태에 있어서, 일 디바이스는, 소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 고정 소수점 연산을 이용하여 반복적으로 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키는 수단을 포함한다. 또한, 이 디바이스는, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키는 수단을 포함한다. 소스 계수들의 행렬은 미디어 데이터를 나타낸다. 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0035] 또다른 양태에 있어서, 컴퓨터 판독가능 매체는 명령들을 포함한다. 명령들은, 프로그램가능한 프로세서로 하여금, 소스 계수들의 행렬에서의 각 계수를 좌측-시프팅시킴으로써, 조정된 계수들의 행렬을 발생시키도록 한다. 또한, 명령들은, 프로그램가능한 프로세서로 하여금, 고정 소수점 연산을 이용하여 반복적으로 조정된 계수들의 행렬에서의 계수들에 1-차원 변환을 적용함으로써, 변환된 계수들의 행렬을 발생시키도록 한다. 또한, 명령들은, 프로그램가능한 프로세서로 하여금, 변환된 계수들의 행렬에서의 각 계수를 스케일링함으로써, 스케일링된 계수들의 행렬을 발생시키도록 한다. 또한, 명령들은, 프로그램가능한 프로세서로 하여금, 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대해 하나 이상의 바이어스 값들을 가산함으로써, 바이어싱된 계수들의 행렬을 발생시키도록 한다. 또한, 명령들은, 프로그램가능한 프로세서로 하여금, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들을 우측-시프팅시킴으로써, 출력 계수들의 행렬을 발생시키도록 한다. 소스 계수들의 행렬은 미디어 데이터를 나타낸다. 출력 계수들의 행렬에서의 계수들은, 이상적인 순이산 코사인 변환을 이용하여 소스 계수들의 행렬을 변환함으로써 생성되는 값들의 근사치들이다. 스케일링된 계수들의 행렬에서의 하나 이상의 계수들에 대한 하나 이상의 바이어스 값들의 가산은, 포지티브 에러들 및 네거티브 에러들로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한

다. 포지티브 에러들 및 네거티브 에러들은, 제 1 크기만큼 바이어싱된 계수들의 행렬에서의 계수들의 제한된 정밀도의 고정 소수점 표현들을 우측-시프팅시키는 것으로부터 기인한 값들과, 정밀도를 고려하지 않고, $2^{(제 1 크기)}$ 로 바이어싱된 계수들의 행렬에서의 계수들을 제산하는 것으로부터의 결과들 사이의 차이들을 나타낸다.

[0036] 몇몇 경우에, 컴퓨터 판독가능 매체는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있는데, 이 컴퓨터 프로그램 제품은 제조자에게 판매되고/되거나 일 디바이스에 사용될 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있고, 몇몇 경우에는 포장재도 포함할 수도 있다.

[0037] 하나 이상의 실시예의 상세가 아래의 상세한 설명 및 첨부 도면에 설명되어 있다. 본 발명의 다른 특징, 목적 및 이점은, 상세한 설명, 도면 및 특허청구범위로부터 자명할 것이다.

[0038] **도면의 간단한 설명**

[0039] 도 1 은 미디어 파일을 인코딩 및 디코딩하는 예시적인 디바이스를 도시한 블록도이다.

[0040] 도 2 는 인코딩 모듈의 예시적인 상세를 도시한 블록도이다.

[0041] 도 3 은 디코딩 모듈의 예시적인 상세를 도시한 블록도이다.

[0042] 도 4 는 인코딩 모듈의 예시적인 동작을 도시한 흐름도이다.

[0043] 도 5 는 디코딩 모듈의 예시적인 동작을 도시한 흐름도이다.

[0044] 도 6 은 역이산 코사인 변환 ("IDCT") 모듈의 예시적인 상세를 도시한 블록도이다.

[0045] 도 7 은 IDCT 모듈의 예시적인 동작을 도시한 흐름도이다.

[0046] 도 8 은 순이산 코사인 변환 ("FDCT") 모듈의 예시적인 상세를 도시한 블록도이다.

[0047] 도 9 는 FDCT 모듈의 예시적인 동작을 도시한 흐름도이다.

[0048] 도 10 은 제 1 예시적인 1-차원 변환을 도시한 흐름도이다.

[0049] 도 11 은 제 2 예시적인 1-차원 변환을 도시한 흐름도이다.

[0050] 도 12 는 IDCT 모듈에 의해 이용되는 예시적인 스케일링 1-차원 변환을 도시한 흐름도이다.

[0051] **상세한 설명**

[0052] 도 1 은 미디어 파일을 인코딩 및 디코딩하는 예시적인 디바이스 (2) 를 도시한 블록도이다. 디바이스 (2) 는, 퍼스널 컴퓨터, 모바일 무선전화기, 서버, 네트워크 어플라이언스, 차량에 통합된 컴퓨터, 비디오 게이밍 플랫폼, 휴대용 비디오 게임 디바이스, 컴퓨터 워크스테이션, 컴퓨터 키오스크, 디지털 사이니지 (digital signage), 메인프레임 컴퓨터, 텔레비전 셋탑 박스, 네트워크 전화기, PDA (Personal Digital Assistant), 비디오 게임 플랫폼, 모바일 미디어 플레이어, 홈 미디어 플레이어, 디지털 비디오 프로젝터, 퍼스널 미디어 플레이어 (예를 들어, iPod), 또는 또다른 타입의 전자 디바이스를 포함할 수도 있다.

[0053] 디바이스 (2) 는, 미디어 데이터를 발생시키는 미디어 소스 (4) 를 포함할 수도 있다. 미디어 소스 (4) 는, 이미지 데이터를 캡처하는 디지털 비디오 또는 스틸 포토 카메라를 포함할 수도 있다. 미디어 소스 (4) 는 디바이스 (2) 에 내장될 수도 있고, 또는 주변 디바이스로서 디바이스 (2) 에 부착될 수도 있다. 또한, 미디어 소스 (4) 는, 오디오 데이터를 레코딩하는 마이크로폰을 포함할 수도 있다. 미디어 소스 (4) 는 미디어 데이터를 프로세서 (6) 로 제공할 수도 있다. 프로세서 (6) 는 디지털 신호 프로세서 ("DSP"), 마이크로프로세서, 또는 일부 다른 타입의 집적 회로를 포함할 수도 있다.

[0054] 프로세서 (6) 가 미디어 소스 (4) 로부터 미디어 데이터를 수신하는 경우, 인코딩 모듈 (8) 은 미디어 데이터를 인코딩할 수도 있다. 인코딩 모듈 (8) 은 프로세서 (6) 에 의해 실행되는 소프트웨어를 포함할 수도 있다. 대안적으로, 인코딩 모듈 (8) 은, 미디어 데이터를 인코딩하는 프로세서 (6) 내의 전문 하드웨어를 포함할 수도 있다. 또다른 대안에 있어서, 인코딩 모듈 (8) 은, 미디어 데이터를 인코딩하는 소프트웨어와 하드웨어의 임의의 조합을 포함할 수도 있다.

[0055] 인코딩 모듈 (8) 은 미디어 저장소 (10) 에 인코딩된 미디어 데이터를 저장할 수도 있다. 미디어 저장소 (10) 는 플래시 메모리, RAM (Random Access Memory), 하드디스크 드라이브, 또는 일부 다른 타입의 휘발성 또는 비휘발성 데이터 저장 유닛을 포함할 수도 있다.

[0056] 디코딩 모듈 (12) 은 미디어 저장소 (10) 로부터 인코딩된 미디어 데이터를 취득할 수도 있다. 디코딩 모듈 (12) 은 프로세서 (6) 에 의해 실행되는 소프트웨어를 포함할 수도 있다. 대안적으로, 디코딩 모듈 (12) 은, 인코딩된 미디어 데이터를 디코딩하는 프로세서 (6) 내의 전문 하드웨어를 포함할 수도 있다. 또다른 대안에 있어서, 디코딩 모듈 (12) 은, 인코딩된 미디어 데이터를 디코딩하도록 협력하는 소프트웨어와 하드웨어의 조합을 포함할 수도 있다.

[0057] 디바이스 (2) 에서의 프리젠테이션 드라이버 (13) 는 미디어 프리젠테이션 유닛 (14) 으로 하여금 디코딩 모듈 (12) 에 의해 디코딩된 미디어 데이터를 프리젠테이션하도록 한다. 예를 들어, 미디어 프리젠테이션 유닛 (14) 은, 이미지 또는 비디오 미디어 데이터를 프리젠테이션하는 컴퓨터 모니터를 포함할 수도 있다. 또다른 실시예에 있어서, 미디어 프리젠테이션 유닛 (14) 은, 오디오 미디어 데이터를 프리젠테이션하는 오디오 출력 디바이스 (예를 들어, 스피커) 를 포함할 수도 있다. 미디어 프리젠테이션 유닛 (14) 은 디바이스 (2) 에 통합될 수도 있고, 또는 주변 디바이스로서 유선 링크 또는 무선 링크를 통해 디바이스 (2) 에 접속될 수도 있다. 프리젠테이션 드라이버 (13) 는, 미디어 프리젠테이션 유닛 (14) 으로 하여금 미디어 데이터를 프리젠테이션하도록 하는 디바이스 드라이버나 다른 소프트웨어, 하드웨어나 펌웨어 유닛, 또는 일부 다른 메커니즘을 포함할 수도 있다.

[0058] 또한, 디바이스 (2) 는 네트워크 인터페이스 (16) 를 포함할 수도 있다. 네트워크 인터페이스 (16) 는 유선 링크 또는 무선 링크를 통한 디바이스 (2) 와 컴퓨터 네트워크 사이의 통신을 용이하게 할 수도 있다. 예를 들어, 네트워크 인터페이스 (16) 는 디바이스와 모바일 전화 네트워크 사이의 통신을 용이하게 할 수도 있다.
디바이스 (2) 는 네트워크 인터페이스 (16) 를 통해 미디어 파일을 수신할 수도 있다. 예를 들어, 디바이스 (2) 는, 네트워크 인터페이스 (16) 를 통해 사진, 비디오 클립, 스트리밍 비디오 (예를 들어, 텔레비전, 비디오 회의, 영화), 오디오 클립 (예를 들어, 호출음 (ringtone), 노래, MP3 파일), 스트리밍 오디오 (예를 들어, 디지털 라디오 방송국, 음성 통화 등) 를 수신할 수도 있다. 네트워크 인터페이스 (16) 가 미디어 파일 또는 비디오 비트스트림을 수신하는 경우, 네트워크 인터페이스 (16) 는 미디어 저장소 (10) 에 미디어 파일 또는 비디오 비트스트림을 저장할 수도 있다.

[0059] 비디오 신호는 픽처의 시퀀스에 관하여 설명될 수도 있는데, 이는 프레임 (전체 픽처), 또는 필드 (예를 들어, 프레임의 홀수 라인 또는 짝수 라인 중 어느 하나를 포함하는 픽처) 를 포함한다. 또한, 각 프레임 또는 필드는 프레임 또는 필드의 하위부분 또는 2 개 이상의 슬라이스를 포함할 수도 있다. 본 명세서에 이용된 바와 같이, 단독으로 또는 다른 단어와 협력하여, "프레임" 이라는 용어는 픽처, 프레임, 필드, 또는 이들의 슬라이스를 언급할 수도 있다.

[0060] 인코딩 모듈 (8) 이 일련의 비디오 프레임을 인코딩하는 경우, 인코딩 모듈 (8) 은 비디오 프레임 중 하나를 "i-프레임" 으로 선택함으로써 시작할 수도 있다. 예를 들어, 인코딩 모듈 (8) 은 모든 8 번째 프레임을 i-프레임으로서 선택할 수도 있다. i-프레임은, 다른 프레임을 참조하지 않는 프레임이다. i-프레임을 선택한 이후에, 인코딩 모듈 (8) 은 "공간적 인코딩" 을 이용하여 i-프레임을 인코딩한다. 또한, 인코딩 모듈 (8) 은 "시간적 인코딩" 을 이용하여 나머지 프레임을 인코딩할 수도 있다.

[0061] 공간적 인코딩을 이용하여 프레임을 인코딩하기 위해서, 인코딩 모듈 (8) 은 프레임 데이터를 픽셀 블록으로 쪼갤 수도 있다. 예를 들어, 인코딩 모듈 (8) 은, 프레임 데이터를, 폭이 8 개의 픽셀이며 높이가 8 개의 픽셀인 픽셀 블록으로 쪼갤 수도 있다 (즉, 각 픽셀 블록은 64 개의 픽셀을 포함함). 그런 다음, 인코딩 모듈 (8) 은 각 픽셀 블록에서의 픽셀의 픽셀 성분값을 분리된 픽셀 성분값의 행렬로 분리할 수도 있다. 픽셀의 픽셀 성분값은, 픽셀의 양상 (appearance) 을 특성화하는 값이다. 예를 들어, 각 픽셀은 Y 픽셀 성분값, Cr 픽셀 성분값, 및 Cb 픽셀 성분값을 특정할 수도 있다. Y 픽셀 성분값은 픽셀의 휘도를 나타내고, Cr 픽셀 성분값은 픽셀의 레드 색차를 나타내고, Cb 픽셀 성분값은 픽셀의 블루 색차를 나타낸다. 이 실시예에 있어서, 인코딩 모듈 (8) 이 픽셀 블록의 픽셀 성분값을 분리하는 경우, 인코딩 모듈 (8) 은 Y 픽셀 성분값의 행렬, Cr 픽셀 성분값의 행렬, 및 Cb 픽셀 성분값의 행렬을 획득할 수도 있다.

[0062] 픽셀 성분값을 픽셀 성분값의 행렬로 분리한 이후에, 인코딩 모듈 (8) 은 픽셀 성분값의 행렬 각각에 대한 변환된 계수의 행렬을 발생시킨다. 인코딩 모듈 (8) 은, 먼저 픽셀 성분값의 행렬에서의 픽셀 성분값을 좌측-시프팅시켜 조정된 계수의 행렬을 발생시킴으로써, 픽셀 성분값의 행렬에 대한 변환된 계수의 행렬을 발생시킬 수

도 있다. 그런 다음, 인코딩 모듈 (8) 은 고정 소수점 연산을 이용하여, 반복적으로 조정된 계수의 행렬에 1-차원 변환을 적용함으로써, 계수의 행렬을 발생시킨다. 몇몇 구현에 있어서, 그런 다음, 인코딩 모듈 (8) 은, 스케일 인수의 세트로 변환된 계수의 행렬을 스케일링함으로써, 변환된 계수의 행렬을 발생시킬 수도 있다. 이들 스케일 인수 각각은 정수값이다. 스케일 인수는, 1-차원 변환 내의 인수가 단순 유리수를 이용하여 근사될 수도 있는 방식으로 선택되었다. 스케일링을 이용하지 않는 구현에 있어서, 이 변환을 적용함으로써 발생된 계수의 행렬은 변환된 계수의 행렬이다.

[0063] 변환된 계수의 행렬에서의 각 계수는, 인코딩된 계수의 행렬에 대해 이상적인 2-차원 순이산 코사인 변환 ("FDCT") 을 적용함으로써 생성되는 값의 행렬에서의 대응하는 값에 근사한다. 이상적인 1-차원 FDCT 는:

[0064] $$t(k) = c(k)\sum_{n=0}^{N-1} s(n)\cos\frac{\pi(2n+1)k)}{2N}$$

[0065] 으로 정의되고, 여기서 s 는 N 개의 오리지널 값의 어레이이고, t 는 N 개의 변환된 값의 어레이이고, 계수 c 는,

[0066] $c(0) = \sqrt{1/N}, c(k) = \sqrt{2/N}$ (여기서, 1 ≤ k ≤ N-1)

[0067] 로 주어진다.

[0068] 이상적인 2-차원 FDCT 는 공식:

[0069] $$t(i,j) = c(i,j)\sum_{n=1}^{N-1}\sum_{m=0}^{N-1} s(m,n)\cos\frac{\pi(2m+1)i}{2N}\cos\frac{\pi(2n+1)j}{2N}$$

[0070] 으로 정의되고, 여기서 s 는 N 개의 오리지널 값의 어레이이고, t 는 N 개의 변환된 값의 어레이이고, $c(i,j)$ 는 $c(i,j) = c(i)c(j)$ 로 주어지고, c(k) 는 1-차원 경우에서와 같이 정의된다.

[0071] 변환된 계수의 행렬을 도출한 이후에, 인코딩 모듈 (8) 은, 변환된 계수의 행렬에서의 계수를 양자화함으로써 양자화된 계수의 행렬을 발생시킨다. 변환된 계수의 양자화는, 변환된 계수의 행렬에서의 고주파 계수와 연관된 정보의 양을 감소시킬 수도 있다. 양자화된 계수의 행렬을 발생시킨 이후에, 인코딩 모듈 (8) 은 양자화된 계수의 행렬에 엔트로피 인코딩 방식을 적용할 수도 있다. 예를 들어, 인코딩 모듈 (8) 은 이 양자화된 계수의 행렬에서의 양자화된 계수에 허프만 인코딩 방식을 적용할 수도 있다. 인코딩 모듈 (8) 이 양자화된 계수의 행렬 각각에 엔트로피 인코딩 방식을 적용하는 경우, 인코딩 모듈 (8) 은 비디오 비트스트림의 일부로서 인코딩된 행렬을 출력할 수도 있다.

[0072] 시간적 인코딩을 이용하여 프레임을 인코딩하기 위해서, 인코딩 모듈 (8) 은 프레임을 "매크로블록" 으로 분할할 수도 있다. 이용되는 코딩 표준에 종속하여, 이들 매크로블록은 고정 또는 가변 크기일 수도 있고, 중복 또는 비중복될 수도 있다. 예를 들어, 각 매크로블록은 16×16 픽셀 블록일 수도 있다. 프레임에서의 각 매크로블록에 대해, 인코딩 모듈 (8) 은 하나 이상의 기준 프레임에서의 소스 매크로블록을 식별하려고 시도할 수도 있다. 코딩 표준에 종속하여, 기준 프레임은 i-프레임, 예측 프레임, 또는 양방향-예측 프레임일 수도 있다. 인코딩 모듈 (8) 이 기준 프레임에서의 소스 매크로블록을 식별할 수 있는 경우에는, 인코딩 모듈 (8) 은 매크로블록에 대한 모션 벡터를 기록한다. 모션 벡터는, 식별된 소스 매크로블록에 관한 매크로블록의 수평 변위를 나타내는 x 값, 및 식별된 소스 매크로블록에 관한 매크로블록의 수직 변위를 나타내는 y 값을 포함한다. 인코딩 모듈 (8) 이 매크로블록에 대한 소스 매크로블록을 식별할 수 없는 경우에는, 인코딩 모듈 (8) 에는 매크로블록에 대한 모션 벡터를 기록하는 것이 요구되지 않을 수도 있다. 다음에, 인코딩 모듈 (8) 은 "복원된" 프레임을 발생시킨다. 복원된 프레임은, 현재의 프레임에 대해 기록된 모션 벡터에 따라 기준 프레임으로부터 매크로블록을 이동시키는 것으로부터 기인한 프레임을 포함한다. 복원된 프레임을 발생시킨 이후에, 인코딩 모듈 (8) 은, 복원된 프레임의 각 픽셀에서의 픽셀 성분값을 현재의 프레임의 대응 픽셀에서의 대응 픽셀 성분값으로부터 감산하여, "잔여" 프레임을 야기한다. 그런 다음, 인코딩 모듈 (8) 은 엔트로피 인코딩 방식을 이용하여, 현재의 프레임의 매크로블록에 대한 모션 벡터를 압축한다. 또한, 인코딩 모듈 (8) 은 전술한 공간적 인코딩 기술을 이용하여, 잔여 프레임을 압축한다.

[0073] 디코딩 모듈 (12) 은 인코딩 모듈 (8) 과 유사하지만 반전하여 프로세스를 수행할 수도 있다. 예를 들어,

공간적 디코딩 프로세스를 수행하기 위해서, 디코딩 모듈 (12) 은 인코딩된 비디오 비트스트림에서의 양자화된 계수의 인코딩된 행렬 각각에 대해 엔트로피 디코딩 방식을 적용할 수도 있다. 그런 다음, 디코딩 모듈 (12) 은 각각의 양자화된 계수의 행렬에서의 계수를 역양자화함으로써, 각각의 양자화된 계수의 행렬에 대해 역양자화된 계수의 행렬을 발생시킬 수도 있다. 각각의 역양자화된 계수의 행렬에 대해, 디코딩 모듈 (12) 은 역양자화된 계수의 행렬을 스케일링함으로써 스케일링된 계수의 행렬을 발생시킨다.

[0074] 스케일링된 계수의 행렬을 발생시킨 이후에, 디코딩 모듈 (12) 은, 이 행렬의 DC 계수에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산함으로써 바이어싱된 계수의 행렬을 발생시킨다. 이 행렬의 DC 계수는, 이 행렬에서의 다른 계수의 평균값과 동등한 계수이다. 통상적으로, DC 계수는 이 행렬의 좌측 최상부 코너에 있는 계수이다. 보다 상세하게 후술하는 바와 같이, DC 계수에 대한 바이어스 값의 가산은, 디코딩 모듈 (12) 이 행렬의 각 행 및 열에 대해 역이산 코사인 변환 인수 분해 (factorization) 를 적용함으로써 생성된 값을 우측-시프팅시키는 경우에 라운딩 에러를 감소시킬 수도 있다. 이들 라운딩 에러는, 이들 우측-시프트가 계산상 보다 고비용의 제산 연산을 대신하고, 고정 소수점 연산에서의 우측 시프트 연산이 항상 제산 연산과 동일한 결과를 생성하지는 않는다는 사실에 기인할 수도 있다.

[0075] 디코딩 모듈 (12) 이 바이어싱된 계수의 행렬을 발생시킨 이후에, 디코딩 모듈 (12) 은 고정 소수점 연산을 이용하여, 바이어싱된 계수의 행렬에 대한 변환된 계수의 행렬을 발생시킨다. 디코딩 모듈 (12) 은, 반복적으로 바이어싱된 계수의 행렬에 대해 1-차원 변환을 적용함으로써 변환된 계수의 행렬을 발생시킨다. 예를 들어, 디코딩 모듈 (12) 은, 바이어싱된 계수의 행렬의 각 행 벡터에 대해 1-차원 변환을 적용함으로써 중간 계수의 행렬을 발생시킬 수도 있다. 이 실시예에 있어서, 그런 다음, 디코딩 모듈 (12) 은, 중간 계수의 행렬에서의 각 열 벡터에 대해 1-차원 변환을 적용함으로써 변환된 계수의 행렬을 발생시킬 수도 있다.

[0076] 디코딩 모듈 (12) 은 일련의 "버터플라이 구조 연산" 을 이용하여 이러한 1-차원 변환을 적용할 수도 있다. 일반적으로, "버터플라이 구조 연산" 은, 제 1 입력값과 제 1 상수를 승산함으로써 제 1 중간값이 생성되고, 제 1 입력값과 제 2 상수를 승산함으로써 제 2 중간값이 생성되고, 제 2 입력값과 제 1 상수를 승산함으로써 제 3 중간값이 생성되고, 제 2 입력값과 제 2 상수를 승산함으로써 제 4 중간값이 생성되고, 제 1 중간값과 제 3 중간값을 가산함으로써 제 1 출력값이 생성되고, 제 2 중간값과 제 4 중간값의 음수를 가산함으로써 제 2 출력값이 생성되는 연산을 언급한다. 버터플라이 연산에 있어서, 상수는 1 을 포함하여 임의의 유리수 또는 무리수일 수도 있다. 예시적인 버터플라이 구조 연산이 도 10, 도 11 및 도 12 의 실시예에 예시된 변환에 도시되어 있다.

[0077] 수를 나타내는데 이용가능한 비트의 수가 제한된 시스템에 있어서, 버터플라이 구조 연산에서 무리 상수에 의한 승산을 수행하는 것은 비실용적일 수도 있다. 이러한 이유로, 디코딩 모듈 (12) 은, 무리 상수에 근사하는 유리 분수와 값을 승산함으로써 무리 상수에 의한 승산에 근사할 수도 있다. 유리 분수와 값을 효율적으로 승산하기 위해서, 디코딩 모듈 (12) 은 유리 분수의 분자와 이 값을 승산한 다음, 그 결과적인 값을 유리 분수의 분모의 $\log_2$ 만큼 우측-시프팅시킬 수도 있다. 전술한 바와 같이, 우측-시프트 연산은 라운딩 에러를 야기할 수도 있는데, 그 이유는 고정 소수점 연산에서의 우측-시프트 연산이 항상 대응하는 제산 연산과 동등한 결과를 생성하지는 않기 때문이다.

[0078] 보다 상세하게 후술하는 바와 같이, 디코딩 모듈 (12) 은 라운딩 에러를 감소시키기 위해서 일부 유리 분수에서 음의 분자를 이용할 수도 있다. 음의 분자의 이용은, 값의 우측-시프팅 이전에 중심점 바이어스 값을 가산할 필요성을 제거할 수도 있다. 이는 유리할 수도 있는데, 그 이유는 중심점 바이어스 값의 가산이 역이산 코사인 변환의 적용에 불필요한 복잡도를 부가할 수도 있기 때문이다.

[0079] 디코딩 모듈 (12) 이 변환된 계수의 행렬을 발생시킨 경우, 디코딩 모듈 (12) 은, 역양자화된 계수의 행렬의 계수를 스케일링함으로써 부가된 비트의 수에 이 변환을 적용함으로써 부가된 비트의 수를 더한 것과 동등한 포지션의 수만큼 변환된 계수의 행렬에서의 각 계수를 우측-시프팅시킴으로써, 조정된 계수의 행렬을 발생시킨다. 그런 다음, 디코딩 모듈 (12) 은, 조정된 계수의 행렬에서의 계수를 클리핑 (clipping) 함으로써 클리핑된 계수의 행렬을 발생시킬 수도 있다. 조정된 계수의 행렬에서의 계수의 클리핑은 조정된 계수를 변경하여, 이들은 픽셀 성분값에 대해 허용된 범위 내에 있게 된다. 그에 따라, 클리핑된 계수의 행렬은 픽셀 성분값의 행렬로서 특성화될 수도 있다.

[0080] 픽셀 성분값의 행렬을 발생시킨 이후에, 디코딩 모듈 (12) 은, 픽셀 성분값의 행렬을 픽셀 블록에 대한 다른 픽셀 성분값을 저장하는 행렬과 결합함으로써 픽셀 블록을 발생시킬 수도 있다. 다음에, 디코딩 모듈 (12) 은

픽셀 블록을 비디오 프레임으로 결합할 수도 있다.

[0081] 예측 프레임을 디코딩하기 위해서, 디코딩 모듈 (12) 은 전술한 공간적 디코딩 기술을 이용하여, 예측 프레임에 대한 잔여 이미지에서의 양자화된 계수의 행렬을 디코딩할 수도 있다. 또한, 디코딩 모듈 (12) 은 엔트로피 디코딩 방식을 이용하여, 예측 프레임의 모션 벡터를 디코딩할 수도 있다. 다음에, 디코딩 모듈 (12) 은, 모션 벡터에 따라 예측 프레임의 기준 프레임의 매크로블록을 "이동" 시킴으로써 복원된 프레임을 발생시킬 수도 있다. 복원된 프레임을 발생시킨 이후에, 디코딩 모듈 (12) 은 디코딩된 잔여 프레임의 각 픽셀에서의 픽셀 성분값을 복원된 프레임의 대응하는 픽셀에서의 대응하는 픽셀 성분값에 가산한다. 이 가산 결과가 복원된 예측 프레임이다.

[0082] 본 개시물에 기재된 기술은 몇몇 이점을 제공할 수도 있다. 예를 들어, 이산 코사인 변환 및 역이산 코사인 변환의 계산 중의 라운딩 에러의 감소는 이미지 데이터에서의 가시 에러 (visible error) 를 감소시킬 수도 있고, 오디오 데이터에서의 가청 에러 (audible error) 를 감소시킬 수도 있다. 또한, 이들 기술이 고정 소수점 계산에 적용될 수도 있기 때문에, 이들 기술은, 모바일 전화기, PDA (Personal Digital Assistant), 및 퍼스널 미디어 플레이어와 같은 보다 소형이며 보다 덜 복잡한 디바이스에 적용될 수도 있다. 특히, 이들 기술은, IEEE (Institute of Electrical and Electronics Engineers) 표준 1180 에 의해 특정된 정밀도 요건에 따르면서, 가수 부분에서의 매우 제한된 비트의 수 (예를 들어, 3 비트) 를 갖는 고정 소수점 수를 이용하여 적용될 수도 있는데, 이 표준의 전체 내용은 본 명세서에 참조로서 포함되어 있다. 또한, 이들 기술은, ITU-T (International Telecommunication Union Standardization Sector) 권고 H.261, H.263, H.264, T.81 (JPEG) 뿐만 아니라, ISO (International Organization for Standardization)/MEC MPEG (Moving Pictures Experts Group)-1, MPEG-2, 및 MPEG-4 파트 2 미디어 포맷을 포함한 포맷에 적용될 수도 있다.

[0083] 도 2 는 인코딩 모듈 (8) 의 예시적인 상세를 도시한 블록도이다. 인코딩 모듈 (8) 은 "모듈" 세트를 포함할 수도 있다. 이들 모듈은 인코딩 모듈 (8) 의 소프트웨어 명령들의 서브세트를 포함할 수도 있다. 대안적으로, 이들 모듈은 특수-목적 하드웨어를 포함할 수도 있다. 또다른 대안에 있어서, 이들 모듈은 소프트웨어 명령들 및 특수-목적 하드웨어를 포함할 수도 있다.

[0084] 도 2 의 실시예에 예시된 바와 같이, 인코딩 모듈 (8) 은, 인코딩 모듈 (8) 이 비디오 프레임을 i-프레임으로서 처리하는지, 예측 프레임으로서 처리하는지, 또는 양방향-예측 프레임으로서 처리하는지를 제어하는 프레임 제어 모듈 (20) 을 포함한다. 예를 들어, 인코딩 모듈 (8) 이 비디오 프레임을 수신하는 경우, 프레임 제어 모듈 (20) 은, 이 비디오 프레임이 i-프레임, 예측 프레임, 또는 양방향-예측 프레임이라고 이 비디오 프레임과 연관된 비트스트림 플래그가 나타내는지 여부를 판정할 수도 있다. 프레임 제어 모듈 (20) 이, 이 비디오 프레임이 i-프레임이라고 비트스트림 플래그가 나타낸다고 판정하는 경우에는, 프레임 제어 모듈 (20) 은 이 비디오 프레임으로 하여금 비디오 프레임에 대해 즉시 공간적 인코딩을 수행하는 모듈 세트에 의해 처리되도록 할 수도 있다. 한편, 이 비디오 프레임이 예측 프레임 또는 양방향-예측 프레임이라고 프레임 제어 모듈 (20) 이 판정하는 경우에는, 프레임 제어 모듈 (20) 은 이 비디오 프레임으로 하여금 시간적 인코딩을 수행하는 모듈 세트에 의해 처리되도록 할 수도 있다.

[0085] 인코딩 모듈 (8) 은, 비디오 프레임에 공간적 인코딩을 적용하기 위한 일련의 모듈을 포함한다. 이들 모듈은, 블록 스플리터 모듈 (22), 성분 추출 모듈 (24), 순방향 변환 모듈 (26), 양자화 모듈 (28), 및 엔트로피 인코딩 모듈 (30) 을 포함한다. 블록 스플리터 모듈 (22) 은 미디어 소스 (4), 네트워크 인터페이스 (16) 또는 또다른 소스로부터 인코딩되지 않은 비디오 프레임을 수신할 수도 있다. 블록 스플리터 모듈 (22) 이 인코딩되지 않은 비디오 프레임을 수신하는 경우, 블록 스플리터 모듈 (22) 은 이 비디오 프레임을 픽셀 블록으로 분리할 수도 있다. 블록 스플리터 모듈 (22) 은 픽셀 블록을 성분 추출 모듈 (24) 로 제공할 수도 있다.

[0086] 성분 추출 모듈 (24) 이 픽셀 블록을 수신하는 경우, 성분 추출 모듈 (24) 은 각 픽셀의 픽셀 성분값을 상이한 컬러 포맷으로 변환할 수도 있다. 예를 들어, 성분 추출 모듈 (24) 은 각 픽셀을 레드-그린-블루 (RGB) 컬러 포맷으로부터 YCrCb 컬러 포맷으로 변환할 수도 있다. 블록에서의 픽셀을 상이한 컬러 포맷으로 변환한 이후에, 성분 추출 모듈 (24) 은 블록에서의 픽셀의 픽셀 성분값을 픽셀 성분값의 행렬로 분리할 수도 있다. 예를 들어, 성분 추출 모듈 (24) 은 일 픽셀 블록으로부터 Y 값의 행렬, Cr 값의 행렬, 및 Cb 값의 행렬을 추출할 수도 있다. Y 값은 픽셀의 휘도를 특정하고, Cr 값은 픽셀의 레드 색차를 특정하고, Cb 값은 픽셀의 블루 색차를 특정한다. 성분 추출 모듈 (24) 이 픽셀 성분값의 행렬을 추출한 경우, 성분 추출 모듈 (24) 은 이들 행렬 각각을 개별적으로 순방향 변환 모듈 (26) 로 제공할 수도 있다.

[0087] 순방향 변환 모듈 (26) 이 픽셀 성분값의 행렬을 수신하는 경우, 순방향 변환 모듈 (26) 은 변환된 계수의 행렬

을 발생시킨다. 이 변환된 계수의 행렬에서의 각 계수는, 이상적인 순이산 코사인 변환을 이용하여 픽셀 성분값의 행렬을 변환함으로써 생성되는 계수에 근사한다.

[0088] 순방향 변환 모듈 (26) 은 고정 소수점 연산을 이용하여, 픽셀 성분값의 행렬에 1-차원 변환을 적용한다. 고정 소수점 연산을 이용하는 것은 몇몇 상황에서 유리할 수도 있다. 예를 들어, 모바일 전화기와 같은 소형의 디바이스는, 부동 소수점 연산을 수행하는데 요구되는 부동 소수점 유닛을 포함하지 않을 수도 있다. 순방향 변환 모듈 (26) 은, 픽셀 성분값 각각을 좌측-시프팅시킴으로써 스케일링된 계수의 행렬을 발생시키는 프로세스를 시작할 수도 있다. 예를 들어, 순방향 변환 모듈 (26) 은, 1-차원 변환을 적용하는 경우에 순방향 변환 모듈 (26) 이 이용하는 수의 고정 소수점 표현 (fixed-point representations of numbers) 의 정밀도의 비트의 수 (즉, 가수 비트의 수) 에 이 변환을 적용하는 것으로부터 기인하는 변환된 계수를 스케일링함으로써 제거된 정밀도의 비트의 수를 더한 것만큼 픽셀 성분값 각각을 좌측-시프팅시킴으로써, 조정된 계수의 행렬을 발생시킬 수도 있다. 픽셀 성분값 각각을 좌측-시프팅시킨 이후에, 순방향 변환 모듈 (26) 은 조정된 계수의 행렬의 행 벡터 각각에 대해 이 변환을 수행할 수도 있다. 조정된 계수의 행렬의 행 벡터 각각에 대해 이산 코사인 변환을 수행하는 것은 중간 계수의 행렬을 발생시킨다. 다음에, 순방향 변환 모듈 (26) 은 중간 계수의 행렬의 열 벡터 각각에 대해 이 변환을 수행할 수도 있다. 중간 계수의 행렬의 열 벡터 각각에 대해 변환을 수행하는 것은 변환된 계수의 행렬을 야기시킨다.

[0089] 변환된 계수의 행렬을 발생시킨 이후에, 순방향 변환 모듈 (26) 은 상이한 스케일 인수로 변환된 계수의 행렬의 상이한 위치에서의 변환된 계수를 스케일링한다. 후술하는 바와 같이, 디코딩 모듈 (12) 은 역방향 변환의 적용시 이들 스케일 인수의 역수를 이용할 수도 있다. 순방향 변환 모듈 (26) 이 스케일 인수로 변환된 계수를 스케일링하는 것을 완료한 경우, 순방향 변환 모듈 (26) 은 그 결과적인 스케일링된 계수의 행렬을 양자화 모듈 (28) 로 출력할 수도 있다.

[0090] 양자화 모듈 (28) 이 순방향 변환 모듈 (26) 로부터 이 스케일링된 계수의 행렬을 수신하는 경우, 양자화 모듈 (28) 은 스케일링된 계수를 양자화할 수도 있다. 양자화 모듈 (28) 은, 이용되는 코딩 표준에 종속하여 각종 방식으로 스케일링된 계수를 양자화할 수도 있다. 예를 들어, MPEG-4 파트 2 표준에 따르면, 양자화 모듈 (28) 은 다음의 양자화 행렬을 이용하여, i-프레임에 대한 스케일링된 계수의 행렬에서의 계수를 양자화할 수도 있다.

[0091]

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

[0092] 또한, 이 실시예에 있어서, 양자화 모듈 (28) 은 다음의 양자화 행렬을 이용하여, 예측 프레임 또는 양방향-예측 프레임에 대한 스케일링된 계수의 행렬에서의 계수를 양자화할 수도 있다.

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

[0093]

[0094] 양자화 모듈 (28) 이 양자화된 계수의 행렬을 발생시킨 이후에, 엔트로피 인코딩 모듈 (30) 은 엔트로피 인코딩 방식을 이용하여 양자화된 계수의 행렬을 압축할 수도 있다. 엔트로피 인코딩 방식을 이용하여 양자화된 계수의 행렬을 압축하기 위해서, 엔트로피 인코딩 모듈 (30) 은 양자화된 계수를 이 계수의 지그재그 패턴을 취함으로써 벡터로 구성할 수도 있다. 다시 말하면, 엔트로피 인코딩 모듈 (30) 은 양자화된 계수의 2-차원 행렬에서의 양자화된 계수 모두를 예측가능하게 양자화된 계수의 1-차원 벡터로 배열할 수도 있다. 그런 다음, 엔트로피 인코딩 모듈 (30) 은 양자화된 계수의 벡터에 대해 허프만 코딩 또는 산술 코딩과 같은 엔트로피 인코딩 방식을 적용할 수도 있다.

[0095] 또한, 인코딩 모듈 (8) 은 비디오 프레임의 시간적 인코딩을 수행하기 위한 하나 이상의 모듈을 포함한다. 도 2 의 실시예에 예시된 바와 같이, 인코딩 모듈 (8) 은 모션 추정 모듈 (32), 복원된 프레임 발생 모듈 (34), 및 잔여 프레임 발생 모듈 (36) 을 포함한다. 모션 추정 모듈 (32) 은 현재의 비디오 프레임에서의 각 매크로블록에 대한 기준 이미지에서의 소스 매크로블록을 식별하려고 시도한다. 모션 추정 모듈 (32) 은, 현재의 프레임에서의 매크로블록과 유사한 픽셀을 포함하는 기준 이미지에서의 매크로블록을 탐색함으로써, 현재의 프레임에서의 매크로블록에 대한 소스 매크로블록을 식별하려고 시도할 수도 있다. 모션 추정 모듈 (32) 은, 현재의 프레임에서의 매크로블록에 대한 소스 매크로블록을 식별하기 위해서 상이한 코딩 표준에 따른 상이한 크기의 영역을 탐색할 수도 있다. 예를 들어, 모션 추정 모듈 (32) 은, 폭이 32 개의 픽셀이며 높이가 32 개의 픽셀의 픽셀 영역 내에서 소스 매크로블록을 탐색할 수도 있는데, 여기서 현재의 매크로블록은 탐색 영역의 중심에 있다. 모션 추정 모듈 (32) 이 현재의 프레임에서의 매크로블록에 대한 소스 매크로블록을 식별하는 경우, 모션 추정 모듈 (32) 은 현재의 프레임에서의 매크로블록에 대한 모션 벡터를 계산한다. 현재의 프레임에서의 매크로블록에 대한 모션 벡터는, 식별된 소스 매크로블록과 현재의 프레임의 매크로블록 사이의 수평 위치에서의 차이를 나타내는 x 값을 특정한다. 모션 추정 모듈 (32) 이 모션 벡터를 계산하였거나 현재의 프레임에서의 각 매크로블록에 대한 소스 매크로블록을 식별할 수 없었던 이후에, 모션 추정 모듈 (32) 은 현재의 프레임에 대해 계산된 모션 벡터를 복원된 프레임 발생 모듈 (34) 로 제공할 수도 있다.

[0096] 복원된 프레임 발생 모듈 (34) 은 모션 벡터 및 기준 프레임을 이용하여, 복원된 프레임을 발생시킬 수도 있다.

복원된 프레임 발생 모듈 (34) 은, 기준 프레임에서의 소스 매크로블록에 대해 현재의 프레임에서의 각 매크로블록에 대한 모션 벡터를 적용함으로써 복원된 프레임을 발생시킬 수도 있다. 사실상, 복원된 프레임 발생 모듈 (34) 은 프레임을 생성하는데, 이 프레임에서 기준 프레임의 매크로블록은 현재의 프레임의 대응하는 모션 벡터에 의해 표시된 위치로 "이동" 되었다.

[0097] 잔여 프레임 발생 모듈 (36) 은, 복원된 프레임에서의 픽셀 성분값을 현재의 프레임에서의 대응하는 픽셀 성분값으로부터 감산함으로써 잔여 프레임을 발생시킬 수도 있다. 일반적으로, 잔여 프레임은, 복원된 프레임 또는 현재의 프레임 중 어느 하나보다 적은 정보를 포함한다. 잔여 프레임 발생 모듈 (36) 이 잔여 프레임을 발생시킨 이후에, 잔여 프레임 발생 모듈 (36) 은, 잔여 프레임을 공간적으로 인코딩하는 프로세스를 시작하기 위해서 이 잔여 프레임을 블록 스플리터 모듈 (22) 로 제공한다. 또한, 모션 추정 모듈 (32) 은, 모션 벡터를 압축하기 위해서 현재의 프레임에 대한 모션 벡터를 엔트로피 인코딩 모듈 (30) 로 제공할 수도 있다.

잔여 프레임이 공간적으로 인코딩되고, 엔트로피 인코딩 모듈 (30) 이 모션 벡터를 인코딩한 이후에, 스트림 출력 모듈 (38) 은 공간적으로 인코딩된 잔여 프레임 및 인코딩된 모션 벡터를 비디오 비트스트림의 일부로서 포맷팅할 수도 있다.

[0098] 도 3 은 디코딩 모듈 (12) 의 예시적인 상세를 도시한 블록도이다. 디코딩 모듈 (12) 은 엔트로피 디코딩 모듈 (44), 역양자화 모듈 (46), 역방향 변환 모듈 (48), 픽셀 복원 모듈 (50), 프레임 버퍼 (51), 블록 결합기 모듈 (52), 프레임 제어 모듈 (53), 기준 프레임 저장 모듈 (54), 복원된 프레임 발생 모듈 (56), 및 예측 프레임 발생 모듈 (58) 을 포함할 수도 있다. 이들 모듈의 일부 또는 전부는 디코딩 모듈 (12) 의 소프트웨어 명령들의 서브세트를 포함할 수도 있다. 대안적으로, 이들 모듈의 일부 또는 전부는 특수-목적 하드웨어 또는 펌웨어를 포함할 수도 있다. 또다른 대안에 있어서, 이들 모듈은 소프트웨어 명령들 및 특수-목적 하드웨어 또는 펌웨어를 포함할 수도 있다.

[0099] 디코딩 모듈 (12) 이 비디오 프레임을 포함한 비트스트림을 수신하는 경우, 엔트로피 디코딩 모듈 (44) 은 비디오 프레임에서의 양자화된 계수의 행렬에 대해 엔트로피 디코딩 방식을 적용할 수도 있다. 비트스트림은, 이 비트스트림에서의 양자화된 계수의 행렬에 어떤 엔트로피 디코딩 방식을 적용할지를 엔트로피 디코딩 모듈 (44) 에 나타내는 값을 포함할 수도 있다. 또한, 엔트로피 디코딩 모듈 (44) 은 동일하거나 상이한 엔트로피 디코딩 방식을 적용하여, 비디오 프레임의 모션 벡터를 디코딩할 수도 있다.

[0100] 엔트로피 디코딩 모듈 (44) 이 비디오 파일에서의 양자화된 계수의 행렬에 대해 엔트로피 디코딩 방식을 적용한 이후에, 역양자화 모듈 (46) 은 양자화된 계수의 행렬 각각에서의 계수를 역양자화할 수도 있다. 코딩 표준에 종속하여, 역양자화 모듈 (46) 은 각종 방식으로 이들 계수를 역양자화할 수도 있다. 예를 들어, MPEG-4 파트 2 표준에 따르면, 역양자화 모듈 (46) 은 2 가지 상이한 방식으로 전술한 2 개의 양자화 행렬을 이용할 수도 있다. 첫번째로, 역양자화 모듈 (46) 은 이들 양자화 행렬을 이용하여, H.263 스타일 역양자화를 수행할 수도 있다. H.263 스타일 역양자화에 있어서, 역양자화 모듈 (46) 은 다음과 같이 양화자된 값 $QF[v][u]$ 으로부터 복원된 계수 $F''[v][u]$ 을 획득하는데:

[0101] $$|F''[v][u]| = \begin{cases} 0, & (QF[v][u] = 0 \text{ 인 경우}) \\ (2\times|QF[v][u]|+1)\times quantiser_scale, & (QF[v][u]\neq 0,\ quantiser_scale \text{이 홀수인 경우}) \\ (2\times|QF[v][u]|+1)\times quantiser_scale-1, & (QF[v][u]\neq 0,\ quantiser_scale \text{이 짝수인 경우}) \end{cases}$$

[0102] $$F''[v][u]:\ F''[v][u] = \mathrm{Sign}(QF[v][u])\times|F''[v][u]|$$,

[0103] 이고, 이는 $quantiser_scale$ 에 의한 단 1 회의 승산을 수반한다.

[0104] 두번째로, 역양자화 모듈 (46) 은 이들 양자화 행렬을 이용하여, MPEG-1/2 스타일 역양자화를 수행할 수도 있다. MPEG-1/2 스타일 역양자화에 있어서, 역양자화 모듈 (46) 은 부가적인 가중 행렬 $W[w][v][u]$을 이용하는데, 여기서 w 는 어떤 가중 행렬이 이용되고 있는지를 나타내고:

[0105] $$F''[v][u] = \begin{cases} 0, & (QF[v][u] = 0 \text{ 인 경우}) \\ ((2\times QF[v][u]+k)\times W[w][v][u]\times quantiser_scale)/16, & (QF[v][u]\neq 0 \text{인 경우}) \end{cases}$$

[0106] 여기서, $k = \begin{cases} 0 & (\text{인트라 블록}) \\ Sign(QF[v][u]) & (\text{넌-인트라 블록}) \end{cases}$ 이다.

[0107] 역방향 변환 모듈 (48) 이 역양자화된 계수의 행렬을 수신하는 경우, 역방향 변환 모듈 (48) 은 픽셀 성분값의 행렬을 발생시킨다. 상세하게 후술하는 바와 같이, 역방향 변환 모듈 (48) 은, 먼저 역양자화된 계수의 행렬에서의 각 계수를 스케일링하고, 이 행렬의 DC 계수에 바이어스 값을 가산하고, 일련의 1-차원 이산 코사인 변환을 적용하고, 그 결과적인 값을 우측-시프팅시키고, 우측-시프팅된 값을 클리핑한 다음, 클리핑된 값을 출력함으로써, 픽셀 성분값의 행렬을 발생시킨다.

[0108] 역방향 변환 모듈 (48) 이 픽셀 성분값의 행렬을 출력한 이후에, 픽셀 복원 모듈 (50) 은, 픽셀 성분값의 행렬과 비디오 프레임 내의 등가의 위치와 연관된 픽셀 성분값의 행렬을 결합함으로써 픽셀의 행렬을 발생시킬 수도 있다. 예를 들어, 픽셀 복원 모듈 (50) 은 역방향 변환 모듈 (48) 로부터 Y 픽셀 성분값의 행렬, Cb 픽셀 성분값의 행렬, 및 Cr 픽셀 성분값의 행렬을 수신할 수도 있다. 이들 3 개의 행렬 각각은 단일 8×8 픽셀 블록에 대한 픽셀 성분을 포함할 수도 있다. 픽셀 각각은 Y 픽셀 성분값, Cb 픽셀 성분값, 및 Cr 픽셀 성분값을 포함할 수도 있다. 픽셀의 행렬을 발생시킨 이후에, 픽셀 복원 모듈 (50) 은 픽셀 블록을 블록 결합기 모듈 (52) 로 제공할 수도 있다.

[0109] 블록 결합기 모듈 (52) 이 픽셀 블록을 수신하는 경우, 블록 결합기 모듈 (52) 은, 블록 결합기 모듈 (52) 이 비디오 프레임에서의 픽셀 블록의 일부 또는 전부를 수신할 때까지 픽셀 블록을 버퍼링할 수도 있다. 하나 이상의 픽셀 블록을 수신한 이후에, 블록 결합기 모듈 (52) 은 픽셀 블록을 비디오 프레임으로 결합할 수도 있고, 비디오 프레임을 프레임 버퍼 (51) 로 출력할 수도 있다. 비디오 프레임은, 미디어 프리젠테이션 유닛 (14) 에 디스플레이될 때까지 프레임 버퍼 (51) 에 저장될 수도 있다. 또한, 블록 결합기 모듈 (52) 은 비디오 프레임을 프레임 저장 모듈 (54) 로 출력할 수도 있다. 프레임 저장 모듈 (54) 에서의 비디오 프레임은 예측 프레임 및 양방향-예측 프레임의 복원을 위해 기준 프레임으로서 이용될 수도 있다. 또한, 프레임 저장 모듈 (54) 에서의 비디오 프레임은, 예측 프레임 및 양방향-예측 프레임의 복원에 이용되는 잔여 프레임일 수도 있다.

[0110] 예측 프레임 또는 양방향-예측 프레임을 복원하기 위해서, 디코딩 모듈 (12) 은 복원된 프레임 발생 모듈 (56) 을 포함한다. 복원된 프레임 발생 모듈 (56) 은 엔트로피 디코딩 모듈 (44) 로부터 디코딩된 모션 벡터를 수신한다. 또한, 복원된 프레임 발생 모듈 (56) 은 프레임 저장 모듈 (54) 로부터 현재의 프레임의 기준 프레임을 취득한다. 그런 다음, 복원된 프레임 발생 모듈 (56) 은 매크로블록을 기준 프레임에서의 그 위치로부터 모션 벡터에 의해 표시된 위치로 "이동" 시킨다. 복원된 프레임은 이와 같이 매크로블록을 이동시키는 것으로부터 기인한다. 복원된 프레임 발생 모듈 (56) 이 복원된 프레임을 발생시킨 이후에, 복원된 프레임 발생 모듈 (56) 은 복원된 프레임을 예측 프레임 발생 모듈 (58) 로 제공한다.

[0111] 예측 프레임 발생 모듈 (58) 이 일시 프레임 (temporary frame) 을 수신하는 경우, 예측 프레임 발생 모듈 (58) 은 프레임 저장 모듈 (54) 로부터 현재의 프레임에 대한 잔여 프레임을 취득할 수도 있다. 잔여 프레임을 취득한 이후에, 예측 프레임 발생 모듈 (58) 은 복원된 프레임 및 잔여 프레임의 각 픽셀에서의 대응하는 컬러 성분값을 부가할 수도 있다. 복원된 비디오 프레임은 이 부가로부터 기인한다. 다음에, 예측 프레임 발생 모듈 (58) 은 미디어 프리젠테이션 유닛 (14) 상의 결과적인 디스플레이를 위해 이 복원된 프레임을 프레임 버퍼 (51) 로 출력할 수도 있다.

[0112] 도 4 는 인코딩 모듈 (8) 의 예시적인 동작을 도시한 흐름도이다. 도 4 에 설명된 동작이 순차적인 방식으로 설명되어 있지만, 이 동작이 파이프라인 방식으로 수행될 수도 있다는 것이 주목되어야 한다.

[0113] 초기에, 인코딩 모듈 (8) 은 인코딩되지 않은 비디오 프레임의 시퀀스를 수신한다 (60). 예를 들어, 인코딩 모듈 (8) 은 미디어 소스 (4) 로부터 픽셀 세트의 형태로 인코딩되지 않은 프레임의 시퀀스를 수신할 수도 있다. 인코딩 모듈 (8) 이 인코딩되지 않은 프레임의 시퀀스를 수신하는 경우, 인코딩 모듈 (8) 에서의 프레임 제어 모듈 (20) 은, 인코딩되지 않은 프레임의 시퀀스에서의 현재의 프레임이 i-프레임으로서 인코딩되어야 하는지 또는 예측 프레임이나 양방향-예측 프레임으로서 인코딩되어야 하는지를 판정할 수도 있다 (62).

[0114] 현재의 프레임이 i-프레임으로서 인코딩되어야 한다고 프레임 제어 모듈 (20) 이 판정하는 경우에는, 인코딩 모듈 (8) 에서의 블록 스플리터 모듈 (22) 은 현재의 프레임을 픽셀 블록으로 분할할 수도 있다 (64). 예를 들어, 인코딩 모듈 (8) 에서의 블록 스플리터 모듈 (22) 은 현재의 프레임을 2×2, 4×4, 또는 8×8 픽셀 블록으로 분할할 수도 있다.

[0115] 현재의 프레임을 픽셀 블록으로 분할한 이후에, 성분 추출 모듈 (24) 은 픽셀 블록 각각에서 픽셀 성분값을 분리할 수도 있다 (66). 그 결과, 각 픽셀 블록에 대한 픽셀 성분값의 3 개의 블록, 즉 픽셀의 휘도를 나타내는 Y 값의 블록, 픽셀의 블루 색차를 나타내는 Cb 값의 블록, 및 픽셀의 레드 색차를 나타내는 Cr 값의 블록이 존재할 수도 있다.

[0116] 그런 다음, 인코딩 모듈 (8) 에서의 순방향 변환 모듈 (26) 은 픽셀 성분값의 행렬 각각에 대한 스케일링된 계수의 행렬을 발생시킬 수도 있다 (68). 이들 스케일링된 계수의 행렬에서의 계수는, 각각의 픽셀 성분값의 행렬에 대해 이상적인 2-차원 순이산 코사인 변환을 이용함으로써 생성되는 값들의 근사치이다.

[0117] 순방향 변환 모듈 (26) 이 픽셀 성분값의 행렬 각각에 대한 스케일링된 계수의 행렬을 발생시킨 이후에, 인코딩 모듈 (8) 에서의 양자화 모듈 (28) 은 스케일링된 계수의 행렬 각각에서의 계수를 양자화할 수도 있다 (70). 일단 양자화 모듈 (28) 이 스케일링된 계수의 행렬 각각에서의 계수를 양자화하면, 엔트로피 인코딩 모듈 (30) 은 양자화된 계수의 행렬 각각에 대해 엔트로피 인코딩 프로세스를 수행할 수도 있다 (72). 예를 들어, 인코딩 모듈 (8) 은 양자화된 계수의 행렬 각각에 대해 산술 인코딩 방식 또는 허프만 인코딩 방식을 적용할 수도 있다. 엔트로피 인코딩 프로세스는 데이터를 더 압축한다. 그러나, 엔트로피 인코딩 프로세스는 정보의 손실을 야기하지는 않는다. 양자화된 계수의 행렬 각각에 대해 엔트로피 인코딩 프로세스를 수

행한 이후에, 인코딩 모듈 (8) 에서의 스트림 출력 모듈 (38) 은 비디오 프레임의 시퀀스에 대한 비트스트림에 대해 양자화된 계수의 인코딩된 행렬을 부가할 수도 있다 (74). 스트림 출력 모듈 (38) 이 비트스트림에 이 인코딩된 행렬을 부가한 이후에, 프레임 제어 모듈 (20) 은, 현재의 프레임이 프레임 시퀀스의 최종 비디오 프레임인지 여부를 판정할 수도 있다 (76). 현재의 프레임이 프레임 시퀀스의 최종 프레임인 경우에는 (76 의 "예"), 인코딩 모듈 (8) 은 프레임 시퀀스의 인코딩을 완료하였다 (78). 한편, 현재의 프레임이 프레임 시퀀스의 최종 프레임이 아닌 경우에는 (76 의 "아니오"), 인코딩 모듈 (8) 은 루프백하여, 새로운 현재의 프레임이 i-프레임으로서 인코딩되어야 하는지 여부를 판정할 수도 있다 (62).

[0118] 현재의 프레임이 i-프레임으로서 인코딩되지 않아야 하는 경우에는 (62 의 "아니오"), 인코딩 모듈 (8) 에서의 모션 추정 모듈 (32) 은 현재의 프레임을 매크로블록 세트로 분할할 수도 있다 (80). 다음에, 모션 추정 모듈 (32) 은 현재의 프레임에서의 매크로블록 각각에 대한 하나 이상의 기준 프레임에서의 소스 매크로블록을 식별하려고 시도할 수도 있다 (82). 그런 다음, 모션 추정 모듈 (32) 은, 모션 추정 모듈 (32) 이 소스 매크로블록을 식별할 수 있었던 현재의 프레임에서의 매크로블록 각각에 대한 모션 벡터를 계산할 수도 있다 (84).

모션 추정 모듈 (32) 이 모션 벡터를 계산한 이후에, 복원된 프레임 발생 모듈 (34) 은 이 모션 벡터를 이용하여, 기준 프레임에서의 식별된 매크로블록을 모션 벡터에 의해 표시된 위치로 "이동" 시킴으로써 복원된 프레임을 발생시킨다 (86). 그런 다음, 잔여 프레임 발생 모듈 (36) 은, 복원된 프레임에서의 픽셀 성분값을 현재의 프레임에서의 대응하는 픽셀 성분값으로부터 감산함으로써, 현재의 프레임에 대한 잔여 프레임을 발생시킬 수도 있다 (88). 잔여 프레임 발생 모듈 (36) 이 잔여 프레임을 발생시킨 이후에, 엔트로피 인코딩 모듈 (30) 은 엔트로피 인코딩 방식을 이용하여, 현재의 프레임에 대한 모션 벡터를 인코딩할 수도 있다 (90). 또한, 잔여 프레임에 대해 단계 66 내지 단계 74 를 적용함으로써, 잔여 프레임에 대해 공간적 인코딩을 적용할 수도 있다.

[0119] 도 5 는 디코딩 모듈 (12) 의 예시적인 동작을 도시한 흐름도이다. 도 5 에 설명된 동작이 순차적인 방식으로 설명되어 있지만, 이 동작이 파이프라인 방식으로 수행될 수도 있다는 것이 주목되어야 한다.

[0120] 초기에, 디코딩 모듈 (12) 은 인코딩된 비디오 프레임을 수신한다 (100). 인코딩된 비디오 프레임을 수신한 이후에, 디코딩 모듈 (12) 에서의 엔트로피 디코딩 모듈 (44) 은 인코딩된 프레임 내의 데이터 블록에 대해 엔트로피 디코딩 프로세스를 수행할 수도 있다 (102). 엔트로피 디코딩 모듈 (44) 은, 프레임을 인코딩하는데 이용된 엔트로피 인코딩 프로세스와 등가의 엔트로피 디코딩 프로세스를 수행할 수도 있다. 예를 들어, 인코딩 모듈 (8) 이 프레임을 인코딩하는데 허프만 인코딩을 이용한 경우, 엔트로피 디코딩 모듈 (44) 은 프레임을 디코딩하는데 허프만 디코딩을 이용한다. 프레임에서의 각 데이터 블록에 대해 엔트로피 디코딩 프로세스를 적용한 결과, 엔트로피 디코딩 모듈 (44) 은 양자화된 계수의 행렬 세트를 생성하였다.

[0121] 다음에, 디코딩 모듈 (12) 에서의 역양자화 모듈 (46) 은 양자화된 계수의 행렬 각각에서의 계수를 역양자화할 수도 있다 (104). 양자화된 계수의 행렬에서의 각 계수를 역양자화한 이후에, 디코딩 모듈 (12) 에서의 역방향 변환 모듈 (48) 은 픽셀 성분값의 행렬을 발생시킨다 (106). 픽셀 성분값의 행렬 중 하나에서의 픽셀 성분값은, 이상적인 2-차원 역이산 코사인 변환을 이용하여 양자화된 계수의 행렬 중 하나를 변환함으로써 생성되는 대응하는 값의 근사치이다.

[0122] 역방향 변환 모듈 (48) 이 이 계수의 행렬 각각에 대한 픽셀 성분값의 행렬을 컴퓨팅한 경우, 디코딩 모듈 (12) 에서의 픽셀 복원 모듈 (50) 은 픽셀 블록을 생성하기 위해서 픽셀 성분값의 적절한 행렬을 결합할 수도 있다 (108). 예를 들어, 디코딩 모듈 (12) 은, YCrCb 픽셀 블록을 생성하기 위해서 Y 값의 블록과 연관된 Cr 값의 블록 및 연관된 Cb 값의 블록을 결합할 수도 있다. 픽셀 복원 모듈 (50) 이 픽셀 블록을 생성한 이후에, 블록 결합기 모듈 (52) 은 픽셀 블록을 비디오 프레임으로 재결합할 수도 있다 (110).

[0123] 다음에, 디코딩 모듈 (12) 에서의 프레임 제어 모듈 (53) 은, 현재의 프레임이 i-프레임인지 여부를 판정할 수도 있다 (112). 현재의 프레임이 i-프레임인 경우에는 (112 의 "예"), 블록 결합기 모듈 (52) 은 비디오 프레임을 프레임 버퍼 (51) 로 출력할 수도 있다 (114). 한편, 현재의 프레임이 i-프레임이 아닌 경우에는 (즉, 현재의 프레임이 예측 프레임이거나 양방향-예측 프레임인 경우에는) (112 의 "아니오"), 엔트로피 디코딩 모듈 (44) 은 엔트로피 디코딩 방식을 이용하여, 현재의 프레임의 모션 벡터를 디코딩한다 (116). 다음에, 복원된 프레임 발생 모듈 (56) 은 프레임 저장 모듈 (54) 에서의 하나 이상의 기준 프레임 및 디코딩된 모션 벡터를 이용하여, 복원된 프레임을 발생시킨다 (118). 그런 다음, 예측 프레임 발생 모듈 (58) 은 블록 결합기 모듈 (52) 에 의해 발생된 프레임 및 복원된 프레임을 이용하여, 예측 프레임을 발생시킬 수도 있다 (120).

[0124] 도 6 은 역방향 변환 모듈 (48) 의 예시적인 상세를 도시한 블록도이다. 도 6 의 실시예에 예시된 바와 같

이, 역방향 변환 모듈 (48) 은 입력 모듈 (140) 을 포함할 수도 있다. 입력 모듈 (140) 은 역양자화 모듈 (46) 로부터 일 계수의 행렬을 수신할 수도 있다. 예를 들어, 입력 모듈 (140) 은, 계수의 행렬을 저장하는 디바이스 (2) 의 메모리 모듈에서의 위치를 표시하는 포인터를 수신할 수도 있다. 대안적으로, 입력 모듈 (140) 은, 계수의 행렬을 저장하는 내부 데이터 구조체를 포함할 수도 있다.

[0125] 입력 모듈 (140) 이 역양자화된 계수의 행렬을 수신하는 경우, 입력 모듈 (140) 은 이 역양자화된 계수의 행렬을 역방향 변환 모듈 (48) 에서의 스케일링 모듈 (142) 로 제공할 수도 있다. 스케일링 모듈 (142) 은, 역양자화된 계수의 행렬에서의 각 계수를 스케일링함으로써 스케일링된 계수의 행렬을 발생시킨다. 스케일링 모듈 (142) 은, 수의 고정 소수점 표현을 나타내도록 역방향 벡터 변환 모듈 (146) 에 의해 이용된 가수 비트의 수만큼 각 계수를 좌측-시프팅시킴으로써, 역양자화된 계수의 행렬에서의 계수를 스케일링할 수도 있다. 가수 비트는, 소수점 (radix point) 의 우측으로의 비트이다 (즉, 수의 분수부). 이와 같이, 스케일링 모듈 (142) 은 역양자화된 계수의 행렬에서의 계수의 표현을 가수 비트의 적절한 수를 갖는 고정 소수점 표현으로 효과적으로 변환한다. 예를 들어, 역방향 벡터 변환 모듈 (146) 이 3 가수 비트를 포함한 고정 소수점 수를 이용하는 경우, 스케일링 모듈 (142) 은, 3 개의 포지션만큼 역양자화된 계수의 행렬에서의 계수 각각을 좌측-시프팅시킴으로써 스케일링된 계수의 행렬을 발생시킨다.

[0126] 스케일링 모듈 (142) 이 스케일링된 계수의 행렬을 발생시킨 이후에, 계수 바이어싱 모듈 (144) 은, 스케일링된 계수의 행렬의 DC 계수에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산함으로써, 바이어싱된 계수의 행렬을 발생시킬 수도 있다. 전술한 바와 같이, 스케일링된 계수의 행렬의 DC 계수는 통상적으로 이 행렬의 좌측 최상부 위치에 있는 계수이다. 일반적으로, DC 계수는 행렬에서의 다른 계수의 평균값을 나타낸다.

[0127] 보완 바이어스 값은, 포지티브 에러 및 네거티브 에러로 하여금, 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 하는 값이다. 예를 들어, 보완 바이어스 값은, DC 계수가 음수가 아닌 경우에는 0 과 동등하며, DC 계수가 음수인 경우에는 -1 과 동등한 부호-적응형 (sign-adaptive) 바이어스 값일 수도 있다. 16-비트 프로세서에서 DC 계수에 대해 부호-적응형 바이어스 값을 가산하기 위해서, 계수 바이어싱 모듈 (144) 은 다음의 공식을 이용할 수도 있다:

[0128] [공식 1]

[0129] $$DC_coefficient = DC_coefficient + (1 << (P + 2)) + (DC_coefficient >> 15)$$.

[0130] [공식 1] 에 있어서, 중심점 바이어스를 제공하도록 항 (1 << (P+2)) 이 가산된다. P 는, 역방향 벡터 변환 모듈 (146) 에 의해 수행되는 변환에서 이용된 고정 소수점 가수 비트 (즉, 소수점의 우측으로의 비트) 의 수를 나타내는 상수이다. 수 2 가 P 에 가산되는데, 그 이유는 우측-시프트 모듈 (148) 이 (P+3) 만큼 모든 계수를 우측-시프팅시킬 수도 있기 때문이고, 여기서 수 '3' 은, 역이산 코사인 변환을 수행함으로써 부가된 정밀도의 비트로부터 비롯된다. 이 점을 부연하기 위해서, 1 을 (P+2) 만큼 좌측-시프팅시킴으로써 수 x 가 발생되고, x 를 (P+3) 만큼 우측-시프팅시킴으로써 수 z 가 발생되는 경우, z=1/2 이다. (다르게 말하면, $2^{P+2} / 2^{P+3} = 2^0 / 2^1 = ½$ 이다). 따라서, DC 계수에 (1 << (P+2)) 을 가산하는 것은, DC 계수에 (1 << (P+3))/2 를 가산하는 것과 등가이다.

[0131] [공식 1] 에 있어서, 항 $(DC_coefficient >> 15)$ 은 15 개의 포지션만큼 16-비트 DC 계수를 우측으로 시프팅시킨다. 나머지 1 비트는, DC 계수의 부호를 나타낸다. 예를 들어, DC 계수가 0b1111 1100 0011 0101 이었다고 가정한다. 이 실시예에 있어서, 부호 확장 (sign extension) 을 이용하여, DC 계수를 15 개의 포지션만큼 우측으로 시프팅시키면, 0b1111 1111 1111 1111 (십진법, -1) 의 값이 된다. 유사하게, DC 계수가 0b0111 1111 1111 1111 인 경우, DC 계수를 15 개의 포지션만큼 우측으로 시프팅시키면, 0b0000 0000 0000 0000 (십진법, 0) 의 값이 된다.

[0132] 또다른 실시예에 있어서, 보완 바이어스 값은 디더링 (dithering) 바이어스 값일 수도 있다. 디더링 바이어스 값은, -1 또는 0 과 동등한 값이다. 16-비트 프로세서에서 DC 계수에 부호-디더링 바이어스 값을 가산하기 위해서, IDCT 모듈 (48) 은 다음의 공식을 이용할 수도 있다:

[0133] [공식 2]

[0134] $$DC_coefficient = DC_coefficient + (1 << (P + 2)) + dither(-1 \mid 0)$$.

[0135] [공식 2] 에 있어서, P 는 $DC_coefficient$ 에서의 가수 비트의 수를 나타낸다. 항 (1 << (P+2)) 은 중심점 바이어스를 가산한다. 항 $dither(-1|0)$ 은, IDCT 모듈 (48) 이 의사 랜덤에 기초하여 -1 또는 0 을 선택한다는 것을 나타낸다.

[0136] 또한, 계수 바이어싱 모듈 (144) 은, 스케일링된 계수의 행렬에서의 AC 계수 각각에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산할 수도 있다. 행렬의 AC 계수는, 이 행렬에서 DC 계수 이외의 모든 계수이다. AC 계수 각각에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산하는 것은 라운딩 에러를 더 감소시킬 수도 있다.

[0137] 계수 바이어싱 모듈 (144) 이 바이어싱된 계수의 행렬을 발생시킨 이후에, 역방향 벡터 변환 모듈 (146) 은, 고정 소수점 연산을 이용하여 반복적으로 바이어싱된 계수의 행렬에 1-차원 변환을 적용함으로써, 변환된 계수의 행렬을 발생시킨다. 예를 들어, 역방향 벡터 변환 모듈 (146) 은, 고정 소수점 연산을 이용하여 바이어싱된 계수의 행렬에서의 계수의 각 행 벡터에 대해 1-차원 변환을 적용함으로써, 중간 값의 행렬을 발생시킬 수도 있다. 다음에, 역방향 벡터 변환 모듈 (146) 은, 고정 소수점 연산을 이용하여 중간 값의 행렬에서의 각 열 벡터에 대해 1-차원 변환을 적용함으로써, 변환된 계수의 행렬을 컴퓨팅할 수도 있다. 예시적인 1-차원 변환이 아래의 도 10 및 도 11 에 예시되어 있다.

[0138] 역방향 벡터 변환 모듈 (146) 이 변환된 계수의 행렬을 발생시킨 이후에, 우측-시프트 모듈 (148) 은, 스케일링 중에 또한 변환의 적용 중에 부가된 비트의 수와 동등한 포지션의 수만큼 변환된 계수의 행렬에서의 계수 각각을 우측-시프팅시킴으로써 조정된 계수의 행렬을 발생시킬 수도 있다. 예를 들어, 변환을 적용하는 것이 부가적인 3 비트를 야기하고, 계수를 스케일링하는 것이 부가적인 3 비트를 부가하는 경우, 우측-시프트 모듈 (148) 은 6 (3+3) 개의 포지션만큼 계수 각각을 우측-시프팅시킬 수도 있다.

[0139] 우측-시프트 모듈 (148) 은 2^b 로 계수 각각을 제산하는 대체물로서 이 우측-시프트를 수행할 수도 있는데, 여기서 b 는, 스케일링 중에 스케일링 모듈 (142) 에 의해 계수의 가수 부분에 부가된 비트의 수에 역방향 벡터 변환 모듈 (146) 에 의해 부가된 부가적인 비트의 수를 더한 것과 동등하다. 소스 이미지 데이터와 결과 이미지 데이터 사이의 차이는, $x_n/2^b$ 와 $x_n >> b$ 사이의 차이로 인해 발생할 수도 있는데, 여기서 x_n 은 위치 n 에 있는 행렬에서의 계수이다. 이들 차이의 일부는, 고정 소수점 연산의 제한된 정밀도에 있어서, $(x_n >> b)$ 이 x_n 의 모든 값에 대해 $(x_n/2^b)$ 와 동등하지 않다는 사실에 기인한 라운딩 에러로 인해 발생할 수도 있다. 예를 들어, 인코딩 모듈 (8) 은 16 비트의 고정 소수점 수, x_n = 63 (이진법으로, 0000 0000 0011 1111), 및 b = 6 을 이용할 수도 있다. 이 실시예에 있어서, 0000 0000 0011 1111 을 16 개의 포지션만큼 우측으로 시프팅시키면, 0000 0000 0000 0000 의 이진수가 된다. 그에 따라, 63 >> 6 = 0 이다. 한편, $63/2^6$ = 63/64 = 0.984375 이다. 0 과 0.984 사이의 차이는, 소스 이미지와 결과 이미지 사이의 가시 차이를 야기할 수도 있다.

[0140] DC 계수에 바이어스 값을 가산하는 것은 라운딩 에러로 인한 차이를 감소시킨다. 예를 들어, 계수 바이어싱 모듈 (144) 은 x_n 에 중심점 바이어스 값 c 를 가산할 수도 있다. 중심점 바이어스 값 c 는 $2^b/2 = 2^{(b-1)}$ 과 동등할 수도 있다. 예를 들어, b = 6 인 경우, $c = 2^6/2 = 64/2 = 32$ 이다. 이 실시예에 있어서, x_n = 63 인 경우, 63 + 32 = 95 (0b0000 0000 0101 1111) 이다. 95 를 6 개의 포지션만큼 우측-시프팅시킨 것 = 1 (이진법으로, 0b0000 0000 0000 0001) 이다. 중심점 바이어스 값을 가산한 이후에 생성된 값 1 은, 중심점 바이어스 값을 가산하지 않고 생성된 값 0 보다 0.984375 의 정확한 값에 더 근접한다.

[0141] 중심점 바이어스 값 c 를 가산하는 것에 부가하여, 계수 바이어싱 모듈 (144) 은 또한 DC 계수에 보완 바이어스 값을 가산할 수도 있다. 예를 들어, 계수 바이어싱 모듈 (144) 은 부호 적응형 바이어스 값 $d(x_n)$ 을 가산할 수도 있다. 부호-적응형 바이어스 값 $d(x_n)$ 은, x_n 이 음수인 경우에는 -1 과 동등할 수도 있고, x_n 이 음수가 아닌 경우에는 0 과 동등할 수도 있다. 부호-적응형 바이어스 값 $d(x_n)$ 을 가산하는 것은, 부호-적응형 바이어스 값 $d(x_n)$ 이 없는 값보다 양호한 근사치인 값을 야기할 수도 있다. 예를 들어, 부호-적응형 바이어스 값 $d(x_n)$ 없이, 함수 $(x_n + c) >> b$ 는 0 에 대해 대칭적이지 않다. 예를 들어, x_n = 32, b = 6,

c = 32 인 경우, $(x_n + c) >> b = 1$ 이다. 그러나, $x_n = -32$, b = 6, c = 32 인 경우, $(x_n + c) >> b = 0$ 이다. 이러한 0 에 대한 대칭성의 결여는, 연속적인 예측 비디오 프레임을 계산하는 경우에 점진적으로 보다 큰 에러를 야기할 수도 있다. 또한, $x_n >> b$ 와 $x_n/2^b$ 사이의 차이는, x_n 이 0 보다 작은 경우보다 x_n 이 0 보다 큰 경우에 더 크다. 또한, 이는 에러를 생성할 수도 있다.

[0142] 부호-적응형 바이어스 값 $d(x_n)$ 은 이들 문제점을 정정할 수도 있다. 예를 들어, $x_n = 32$, b = 6, c = 32 이면, $d(x_n) = 0$ 이라고 가정한다. 그에 따라, $(x_n + c + d(x_n)) >> b = 1$ 이다. 이하, $x_n = -32$, b = 6, c = 32 이면, $d(x_n) = -1$ 이라고 가정한다. 그에 따라, $(x_n + c + d(x_n)) >> b = -1$ 이다. 이 실시예는, 함수 $(x_n + c + d(x_n)) >> b$ 이 0 에 대해 대칭이고, x_n 이 0 보다 작은 경우보다 x_n 이 0 보다 큰 경우에 보다 큰 차이를 생성하지 않는다는 것을 나타낸다.

[0143] 대안적인 구현에 있어서, 계수 바이어싱 모듈 (144) 은 부호-적응형 바이어스 값 d 를 가산하는 대신에 디더링 바이어스 값 e 를 가산할 수도 있다. 계수 바이어싱 모듈 (144) 이 DC 계수에 디더링 바이어스 값 e 를 가산하는 경우, 계수 바이어싱 모듈 (144) 은, 의사 랜덤에 기초하여 e 의 값이 되도록 값 0 또는 값 -1 중 어느 하나를 선택할 수도 있다. 디더링 바이어스 값 e 의 값을 선택하기 위해서, 계수 바이어싱 모듈 (144) 은 다음의 명령들을 포함할 수도 있다.

```
#define IB1 1
#define IB2 2
#define IB5 16
#define IB18 131072
#define MASK (IB1+IB2+IB5)

static unsigned long iseed = 0xAAAAAAAA;
int ditherBit() {
      if (iseed & IB18) {
            iseed = ((iseed ^ MASK) << 1) | IB1;
            return 1;
      } else {
            iseed <<= 1;
            return 0;
      }
}
```

[0144]

[0145] 다수의 비디오 인코딩 표준은, 화상 집합 ("GOP": Group Of Pictures) 으로서 공지된 것을 이용한다. GOP 는, GOP 내의 i-프레임 및/또는 다른 예측 프레임이나 양방향-예측 프레임을 참조하는 예측 프레임 및/또는 양방향-예측 프레임의 세트 및 i-프레임을 포함한다. 예를 들어, 미디어 파일은 계수 행렬의 세트를 이용하여 인코딩되는 i-프레임을 포함할 수도 있다. 비디오 시퀀스를 생성하기 위해서, 디코딩 모듈 (12) 은 이 i-프레임에 기초하여 예측 프레임을 생성할 수도 있다. i-프레임을 디코딩함으로써 야기된 에러는, i-프레임에 기초한 예측 프레임에 반영된다. 또한, 예측 프레임을 디코딩함으로써 야기된 에러는 다음의 예측 프레임에 통합된다. 프레임을 디코딩함으로써 야기된 에러가 0 에 대해 대칭적이지 않거나 보다 큰 포지티브 또는 네거티브 크기를 갖는 경우, 이들 에러는 연속적인 예측 프레임에서의 픽셀 성분값의 값을 신속하게 증가시키거나 감소시킬 수도 있다. 예를 들어, 에러가 보다 큰 포지티브 에러를 갖는 경향이 있는 경우, 이들 에러는 연속적인 예측 프레임에서 합산되어, 정확한 픽셀 성분값보다 높은 픽셀 성분값을 야기할 수도 있다. 그 결과, GOP 내의 연속적인 예측 프레임에서의 픽셀은 부적절하게 컬러 또는 휘도를 변경할 수도 있다. 이는 드리프트 에러로서 공지되어 있다. 너무 심각한 드리프트 에러를 회피하기 위해서, i-프레임으로부터 단지 제한된 개수의 프레임만이 발생될 수 있다. 라운딩으로 인해, 에러는, 보다 큰 정밀도를 갖는 수를 이용하여 변환을 수행하는 경우보다 제한된 가수 비트의 수 (예를 들어, 3 가수 비트) 를 갖는 고정 소수점 수를 이용하여 변환을 수행하는 경우에 크기가 더 클 수도 있다. 그에 따라, 드리프트 에러는 제한된 가수 비트의 수

를 갖는 고정 소수점 수를 이용하여 변환을 수행하는 경우에 특히 문제가 있을 수도 있다.

[0146] 의사 랜덤에 기초하여 디더링 바이어스 값 e 의 값을 선택하는 것은, 각 예측 프레임이 동등하게 보다 큰 포지티브 크기를 갖는 에러 또는 보다 큰 네거티브 크기를 갖는 에러를 가질 가능성이 있다는 것을 의미한다. 따라서, GOP 내에서, 포지티브 에러 및 네거티브 에러는 크기에 있어서 동등한 경향이 있고, 0 에 대해 대칭이 되는 경향이 있다. 포지티브 에러 및 네거티브 에러가 평균하여 0 에 대해 대칭이고, 또한 포지티브 에러 및 네거티브 에러가 평균하여 크기에 있어서 동등하기 때문에, 이들 에러가 후속 예측 프레임에 전달되어 과대화될 (exaggerated) 가능성은 없다. 이는, 포지티브 크기를 갖는 에러가 또다른 프레임에서 네거티브 크기를 갖는 에러를 제거할 가능성이 있고, 또한 네거티브 크기를 갖는 에러가 또다른 프레임에서 포지티브 크기를 갖는 에러를 제거할 가능성이 있기 때문이다. 그에 따라, 디더링 바이어스 값이 에러를 0 에 대해 대칭이 되게 하는 경향이 있고, 또한 포지티브 에러 및 네거티브 에러를 크기에 있어서 동등하게 하는 경향이 있기 때문에, GOP 전체에 걸쳐 보다 적은 드리프트 에러가 존재할 가능성이 있다. 이러한 이유로, 보다 큰 픽처가 GOP 에 포함될 수 있다. 보다 큰 픽처가 GOP 에 포함될 수 있기 때문에, 전체 비디오 시퀀스는 양호한 압축 레이트를 가질 수도 있다. 마찬가지로, 부호-적응형 바이어스 값을 가산하는 것은, 크기에 있어서 동등한 경향이 있으며 0 에 대해 대칭이 되는 경향이 있는 각 프레임 내의 에러를 야기한다. 그 결과, 이들 에러는 후속 예측 프레임에 전달되어 과대화되지 않는다.

[0147] 우측-시프트 모듈 (148) 이 계수를 우측-시프팅시킨 이후에, 클리핑 모듈 (150) 은 계수를 픽셀 성분값의 최대 허용가능한 범위로 제한하기 위해서 이 계수를 "클리핑" 할 수도 있다. 예를 들어, 통상적인 JPEG 이미지에 있어서, 컬러 성분값은 -256 내지 255 의 범위에 있을 수도 있다. 계수의 행렬이 270 과 동등한 계수를 포함한 경우, 클리핑 모듈 (150) 은 이 계수를 255 로 감소시킴으로써 이 계수를 최대 허용가능한 범위로 제한한다. 클리핑 모듈 (150) 이 계수의 클리핑을 완료한 이후에, 이들 계수는 픽셀 성분값을 나타낼 수도 있다. 클리핑 모듈 (150) 이 행렬에서의 계수의 클리핑을 완료하는 경우, 클리핑 모듈 (150) 은 클리핑된 계수의 행렬을 출력 모듈 (152) 로 제공할 수도 있다.

[0148] 출력 모듈 (152) 이 (이하 픽셀 성분값인) 계수의 행렬을 수신하는 경우, 출력 모듈 (152) 은 픽셀 성분값의 행렬을 픽셀 복원 모듈 (50) 로 출력할 수도 있다.

[0149] 도 7 은 역방향 변환 모듈 (48) 의 예시적인 동작을 도시한 흐름도이다. 초기에, 입력 모듈 (140) 은 역양자화된 계수의 행렬을 수신한다 (170). 입력 모듈 (140) 이 역양자화된 계수의 행렬을 수신하는 경우, 스케일링 모듈 (142) 은 역양자화된 계수의 행렬에서의 각각의 값을 스케일링함으로써, 스케일링된 계수의 행렬을 발생시킬 수도 있다 (172). 예를 들어, 스케일링 모듈 (142) 은, 역양자화된 계수의 행렬에서의 각 계수와 스케일 인수의 행렬에서의 동등한 위치의 값을 승산하는 연산을 수행할 수도 있다.

[0150] 스케일링 모듈 (142) 이 스케일링된 계수의 행렬을 발생시킨 이후에, 계수 바이어싱 모듈 (144) 은 스케일링된 계수의 행렬의 DC 계수에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산함으로써, 바이어싱된 계수의 행렬을 발생시킬 수도 있다 (174). 계수 바이어싱 모듈 (144) 이 바이어싱된 계수의 행렬의 DC 계수에 바이어스 값을 가산한 이후에, 역방향 벡터 변환 모듈 (146) 은, 행 카운터가 최대 행 카운터 미만인지 여부를 판정할 수도 있다 (176). 초기에, 행 카운터는 0 으로 설정될 수도 있다. 최대 행 카운터는, 계수의 행렬에서의 행의 수와 동등한 고정값 (static value) 일 수도 있다. 예를 들어, 계수의 행렬이 8 개의 행을 갖는 경우, 최대 행 카운터는 8 과 동등하다.

[0151] 행 카운터가 최대 행 카운터 미만인 경우에는 (176 의 "예"), 역방향 벡터 변환 모듈 (146) 은 고정 소수점 연산을 이용하여, 행 카운터에 의해 표시된 계수의 행렬의 행 벡터에 대해 1-차원 변환을 적용할 수도 있다 (178). 역방향 벡터 변환 모듈 (146) 이 이 계수의 행렬의 행 벡터에 대해 변환을 적용하는 경우, 역방향 벡터 변환 모듈 (146) 은 이 계수의 행 벡터에서의 오리지널 계수를 중간 계수의 벡터로 대체할 수도 있다. 역방향 벡터 변환 모듈 (146) 이 이 계수의 행렬의 행 벡터에 대해 변환을 적용한 이후에, 역방향 벡터 변환 모듈 (146) 은 행 카운터를 증분시킬 수도 있다 (180). 그런 다음, 역방향 벡터 변환 모듈 (146) 은 루프백하여, 다시 행 카운터가 최대 행 카운터 미만인지 여부를 판정할 수도 있다 (176).

[0152] 행 카운터가 최대 행 카운터 미만이 아닌 경우에는 (즉, 이상인 경우에는) (176 의 "아니오"), 역방향 벡터 변환 모듈 (146) 은, 열 카운터가 최대 열 카운터 미만인지 여부를 판정할 수도 있다 (182). 초기에, 열 카운터는 0 으로 설정될 수도 있다. 최대 열 카운터는, 계수의 행렬에서의 열의 수와 동등한 고정값일 수도 있다. 예를 들어, 계수의 행렬이 8 개의 열을 포함하는 경우, 최대 열 카운터는 8 과 동등하다.

[0153] 열 카운터가 최대 열 카운터 미만인 경우에는 (182 의 "예"), 역방향 벡터 변환 모듈 (146) 은 열 카운터에 의해 표시된 중간 계수의 행렬의 열 벡터에 대해 1-차원 변환을 적용할 수도 있다 (184). 역방향 벡터 변환 모듈 (146) 이 중간 계수의 행렬의 열 벡터에 대해 변환을 적용하는 경우, 역방향 벡터 변환 모듈 (146) 은 이 열 벡터에서의 중간 계수를 변환된 계수의 벡터로 대체한다.

[0154] 역방향 벡터 변환 모듈 (146) 이 이 계수의 행렬의 열 벡터에 대해 변환을 적용한 이후에, 역방향 벡터 변환 모듈 (146) 은 열 카운터를 증분시킬 수도 있다 (186). 그런 다음, 역방향 벡터 변환 모듈 (146) 은 루프백하여, 다시 열 카운터가 최대 열 카운터 미만인지 여부를 판정할 수도 있다 (182).

[0155] 열 카운터가 최대 열 카운터 미만이 아닌 경우에는 (즉, 이상인 경우에는) (182 의 "아니오"), 우측-시프트 모듈 (148) 은 행렬에서의 변환된 계수 각각을 우측-시프팅시킬 수도 있다 (188). 우측-시프트 모듈 (148) 이 계수를 우측-시프팅시키는 경우, 우측-시프트 모듈 (148) 은 특정 개수의 포지션만큼 이 계수를 우측으로 시프팅시킬 수도 있다. 행렬에서의 제 2 중간 계수 각각을 우측-시프팅시킨 결과가 조정된 값의 행렬이다. 우측-시프트 모듈 (148) 이 변환된 계수 각각을 우측-시프팅시킨 이후에, 클리핑 모듈 (150) 은, 조정된 계수가 픽셀 성분값의 적절한 범위 내에 있다는 것을 보증하기 위해서 이 조정된 계수를 클리핑할 수도 있다 (190). 예를 들어, 클리핑 모듈 (150) 은, 조정된 계수가 -256 내지 255 의 범위 내에 있다는 것을 보증하기 위해서 이 조정된 계수를 클리핑할 수도 있다. 클리핑 모듈 (150) 이 조정된 계수의 클리핑을 완료하는 경우, 출력 모듈 (152) 은 그 결과적인 픽셀 성분값의 행렬을 출력할 수도 있다 (192).

[0156] 도 8 은 순방향 변환 모듈 (26) 의 예시적인 상세를 도시한 블록도이다. 도 8 의 실시예에 예시된 바와 같이, 순방향 변환 모듈 (26) 은, 성분 추출 모듈 (24) 로부터 픽셀 성분값의 행렬을 수신하는 입력 모듈 (210) 을 포함한다. 입력 모듈 (210) 이 픽셀 성분값의 행렬을 수신하는 경우, 입력 모듈 (210) 은 픽셀 성분값의 행렬을 좌측-시프트 모듈 (212) 로 제공할 수도 있다. 좌측-시프트 모듈 (212) 은, 순방향 변환을 수행하는 동안에 순방향 벡터 변환 모듈 (214) 이 이용한 값에 이용된 가수 비트의 수에서 순방향 변환을 수행함으로써 제거된 가수 비트의 수를 뺀 것만큼 픽셀 성분값의 행렬에서의 픽셀 성분값 모두를 좌측으로 시프팅시킬 수도 있다. 예를 들어, 순방향 변환을 수행하는 동안에 값에 10 가수 비트가 이용되고, 순이산 코사인 변환을 수행함으로써 3 가수 비트가 제거되는 경우에는, 좌측-시프트 모듈 (212) 은 픽셀 성분값을 7 개의 포지션만큼 좌측으로 시프팅시킬 수도 있다. 또다른 실시예에 있어서, 순방향 변환을 수행하는 동안에 값에 3 가수 비트가 이용되고, 순방향 변환을 수행함으로써 3 가수 비트가 제거되는 경우에는, 좌측-시프트 모듈 (212) 은 픽셀 성분값을 0 개의 포지션만큼 좌측으로 시프팅시킬 수도 있다.

[0157] 좌측-시프트 모듈 (212) 이 픽셀 성분값을 시프팅시킨 이후에, 순방향 벡터 변환 모듈 (214) 은 픽셀 성분값의 행렬에서의 각 열 벡터에 대해 순방향 변환을 적용하여, 중간 값의 행렬을 생성할 수도 있다. 다음에, 순방향 벡터 변환 모듈 (214) 은 중간 값의 행렬에서의 각 행 벡터에 대해 순방향 변환을 적용하여, 변환된 계수의 행렬을 생성할 수도 있다. 순방향 벡터 변환 모듈 (214) 이 벡터에 순방향 변환을 적용하는 경우, 순방향 벡터 변환 모듈 (214) 은 아래의 도 12 에서 설명된 순방향 변환을 적용할 수도 있다. 아래의 도 12 에서 설명된 변환은 도 11 에서 설명된 변환의 역이라는 것에 주목하라.

[0158] 순방향 벡터 변환 모듈 (214) 이 변환된 계수의 행렬을 생성한 이후에, 스케일링 모듈 (216) 은 변환된 계수의 행렬에서의 각 변환된 계수에 대해 스케일 인수를 적용할 수도 있다. 스케일링 모듈 (216) 은, 역방향 변환 모듈 (48) 에서의 스케일링 모듈 (142) 에 의해 이용된 스케일 인수의 역수를 적용할 수도 있다. 예를 들어, 변환의 효율을 개선하기 위해서 변환으로부터 하나 이상의 값이 인수분해된 경우, 이들 값은 스케일 인수의 행렬의 기초가 될 수도 있다. 변환된 계수의 행렬에서의 계수는, 이들 값과 계수를 승산함으로써 정정될 수도 있다.

[0159] 라운딩 에러를 감소시키기 위해서, 순방향 변환 모듈 (26) 에서의 계수 바이어싱 모듈 (218) 은 스케일링된 계수의 행렬에서의 계수에 대해 중심점 바이어스 값 및 보완 바이어스 값을 가산할 수도 있다. 스케일링된 계수의 행렬에서의 계수에 대해 부호-적응형 바이어스 값 또는 디더링 바이어스 값을 가산하는 것은, 역방향 변환 모듈 (48) 에서의 계수 바이어싱 모듈 (144) 이 부호-적응형 바이어스 값 또는 디더링 바이어스 값을 가산하는 경우와 유사한 효과를 갖는다. 즉, 계수에 대한 바이어스 값의 가산은 포지티브 에러 및 네거티브 에러로 하여금 평균하여 크기에 있어서 동등하도록 하며, 평균하여 0 에 대해 대칭이 되도록 한다. 이들 에러는, 제 1 크기만큼 스케일링된 계수의 행렬에서의 계수의 제한된 정밀도의 고정 소수점 표현을 우측-시프팅시키는 것으로부터 기인한 값과, 정밀도를 고려하지 않고, $2^{(\text{제 1 크기})}$ 로 스케일링된 계수의 행렬에서의 계수를 제산하는 것으로부터의 결과 사이의 차이를 나타낸다.

[0160] 계수 바이어싱 모듈 (218) 이 바이어싱된 계수의 행렬을 발생시킨 이후에, 순방향 변환 모듈 (26) 에서의 우측-시프트 모듈 (220) 은 바이어싱된 계수의 행렬에서의 계수를 우측-시프팅시킴으로써, 출력 계수의 행렬을 발생시킬 수도 있다. 우측-시프트 모듈 (220) 은, 바이어싱된 계수의 행렬의 계수에서의 가수 비트의 수에 변환을 수행함으로써 계수에 부가된 비트의 수를 더한 것만큼 바이어싱된 계수의 행렬에서의 각 계수를 우측-시프팅시킬 수도 있다.

[0161] 다음의 수학식은, 계수 바이어싱 모듈 (218) 이 부호-적응형 바이어스 값을 가산하는 경우에, 변환된 계수의 행렬에 대한 스케일링 모듈 (216), 계수 바이어싱 모듈 (218), 및 우측-시프트 모듈 (220) 의 영향을 요약하는데:

[0162] $$F[v][u] = (F'[v][u] * S[v][u] + (1 << (P+Q) - ((F'[v][u] >= 0) ? 0 : 1)) >> (P+Q)$$

[0163] 여기서, v = 0..7, u = 0..7 이고, $S[v][u]$ 는 스케일 인수의 행렬에서의 엔트리이고, F 는 스케일링된 계수의 행렬이고, F' 는 변환된 계수의 행렬이고, P 는 변환된 계수의 행렬에서의 계수의 가수 비트의 수를 나타내고, Q 는 변환을 적용함으로써 변환된 계수의 행렬에서의 계수에 부가된 비트의 수를 나타낸다.

[0164] 다음의 수학식은, 계수 바이어싱 모듈 (218) 이 디더링 바이어스 값을 가산하는 경우에, 변환된 계수의 행렬에 대한 스케일링 모듈 (216), 계수 바이어싱 모듈 (218), 및 우측-시프트 모듈 (220) 의 영향을 요약하는데:

[0165] $$F[v][u] = (F'[v][u] * S[v][u] + (1 << 19) - (dither(0:1)) >> 20$$

[0166] 여기서, v = 0 … 7, u = 0 … 7 이고, $S[v][u]$ 는 스케일 인수의 행렬에서의 엔트리이고, F 는 스케일링된 계수의 행렬이고, F' 는 변환된 계수의 행렬이다.

[0167] 스케일링 모듈 (216) 이 스케일링된 계수의 행렬을 발생시킨 이후에, 출력 모듈 (222) 은 이 계수의 행렬을 양자화 모듈 (28) 로 출력할 수도 있다.

[0168] 도 9 는 순방향 변환 모듈 (26) 의 예시적인 동작을 도시한 흐름도이다. 초기에, 입력 모듈 (210) 은 픽셀 성분값의 행렬을 수신한다 (230). 입력 모듈 (210) 이 픽셀 성분값의 행렬을 수신하는 경우, 좌측-시프트 모듈 (212) 은 픽셀 성분값의 행렬에서의 각각의 값을 좌측-시프팅시킴으로써 조정된 계수의 행렬을 발생시킬 수도 있다 (232). 예를 들어, 좌측-시프트 모듈 (212) 은 계수 모두를 10 개의 포지션만큼 좌측으로 시프팅시킬 수도 있다. 이 실시예에 있어서, 순방향 벡터 변환 모듈 (214) 이, 수가 분수 부분에서 10 비트를 이용하여 인코딩되는 고정 소수점 연산을 이용할 수도 있기 때문에, 좌측-시프트 모듈 (212) 은 계수 모두를 10 개의 포지션만큼 좌측으로 시프팅시킬 수도 있다. 따라서, 계수를 10 개의 포지션만큼 좌측으로 시프팅시킴으로써, 좌측-시프트 모듈 (212) 은 픽셀 성분값을 10 가수 비트를 갖는 고정 소수점 수로 효과적으로 변환한다.

[0169] 조정된 계수의 행렬에서의 각 픽셀 성분값을 좌측-시프팅시킨 이후에, 순방향 벡터 변환 모듈 (214) 은, 열 카운터가 최대 열 카운터 미만인지 여부를 판정할 수도 있다 (234). 초기에, 열 카운터는 0 으로 설정될 수도 있다. 최대 열 카운터는, 조정된 계수의 행렬에서의 열의 수와 동등한 고정값일 수도 있다. 예를 들어, 조정된 계수의 행렬이 8 개의 열을 갖는 경우, 최대 열 카운터는 8 과 동등하다.

[0170] 열 카운터가 최대 열 카운터 미만인 경우에는 (234 의 "예"), 순방향 벡터 변환 모듈 (214) 은 열 카운터에 의해 표시된 열 벡터에 대해 1-차원 순방향 변환을 컴퓨팅할 수도 있다 (236). 순방향 벡터 변환 모듈 (214) 이 조정된 계수의 행렬의 열 벡터에 대해 순방향 변환을 컴퓨팅하는 경우, 순방향 벡터 변환 모듈 (214) 은 열 벡터에서의 오리지널 조정된 계수를 중간 계수로 대체한다. 순방향 벡터 변환 모듈 (214) 이 조정된 계수의 행렬의 열 벡터에 대해 순방향 변환을 컴퓨팅한 이후에, 순방향 벡터 변환 모듈 (214) 은 열 카운터를 증분시킬 수도 있다 (238). 그런 다음, 순방향 벡터 변환 모듈 (214) 은 루프백하여, 다시 열 카운터가 최대 열 카운터 미만인지 여부를 판정할 수도 있다 (234).

[0171] 열 카운터가 최대 열 카운터 미만이 아닌 경우에는 (즉, 이상인 경우에는) (234 의 "아니오"), 순방향 벡터 변환 모듈 (214) 은, 행 카운터가 최대 행 카운터 미만인지 여부를 판정할 수도 있다 (240). 초기에, 행 카운터는 0 으로 설정될 수도 있다. 최대 행 카운터는, 계수의 행렬에서의 행 벡터의 수와 동등한 고정값일 수도 있다. 예를 들어, 계수의 행렬이 8 개의 행을 포함하는 경우, 최대 행 카운터는 8 과 동등하다.

[0172] 행 카운터가 최대 행 카운터 미만인 경우에는 (240 의 "예"), 순방향 벡터 변환 모듈 (214) 은 행 카운터에 의해 표시된 행 벡터에 대해 1-차원 이산 코사인 변환을 컴퓨팅할 수도 있다 (242). 순방향 벡터 변환 모듈

(214) 이 이미 행렬의 열 벡터에 대해 순방향 변환을 컴퓨팅했기 때문에, 이하 계수의 행렬은 중간 계수를 포함한다. 순방향 벡터 변환 모듈 (214) 이 중간 계수의 행 벡터에 대해 순방향 변환을 컴퓨팅하는 경우, 순방향 벡터 변환 모듈 (214) 은 행 벡터에서의 중간 계수를 변환된 계수로 대체한다.

[0173] 순방향 벡터 변환 모듈 (214) 이 계수의 행렬의 행 벡터에 대해 이산 코사인 변환을 컴퓨팅한 이후에, 순방향 벡터 변환 모듈 (214) 은 행 카운터를 증분시킬 수도 있다 (244). 그런 다음, 순방향 벡터 변환 모듈 (214) 은 루프백하여, 다시 행 카운터가 최대 행 카운터 미만인지 여부를 판정할 수도 있다 (240).

[0174] 행 카운터가 최대 행 카운터 미만이 아닌 경우에는 (즉, 이상인 경우에는) (240 의 "아니오"), 스케일링 모듈 (216) 은 변환된 계수의 행렬에서의 각 변환된 계수를 스케일링할 수도 있다 (246). 스케일링 모듈 (216) 이 스케일링된 계수의 행렬을 발생시킨 이후에, 계수 바이어싱 모듈 (218) 은 스케일링된 계수의 행렬에서의 계수에 대해 하나 이상의 바이어스 값을 가산함으로써 바이어싱된 계수의 행렬을 발생시킬 수도 있다 (248). 예를 들어, 계수 바이어싱 모듈 (218) 은 스케일링된 계수의 행렬에서의 각 계수에 대해 중심점 바이어스 값 및 부호-적응형 또는 디더링 보완 바이어스 값을 가산할 수도 있다. 다음에, 우측-시프트 모듈 (220) 은 바이어싱된 계수의 행렬에서의 각 계수를 우측-시프팅시킬 수도 있다 (250). 우측-시프트 모듈 (220) 은, 계수 각각에서의 가수 비트의 수에 변환을 적용함으로써 계수에 부가된 비트의 수를 더한 것만큼 각 계수를 우측-시프팅시킴으로써 조정된 계수의 행렬을 발생시킬 수도 있다. 우측-시프트 모듈 (220) 이 조정된 계수의 행렬을 발생시킨 이후에, 출력 모듈 (222) 은 그 결과적인 조정된 계수의 행렬을 출력할 수도 있다 (252).

[0175] 도 10 은 제 1 예시적인 1-차원 변환 (260) 을 도시한 도면이다. 도 10 의 실시예에 예시된 바와 같이, 변환 (260) 은 입력값으로서 X_0 내지 X_7 을 취할 수도 있다. 값 X_0 내지 X_7 은 입력 계수 행렬의 하나의 행 또는 열의 계수를 나타낼 수도 있다. 변환 (260) 은 값 x_0 내지 x_7 을 출력할 수도 있다. 값 X_0 내지 X_7 이 입력 계수 행렬에서의 계수의 행에서의 값인 경우에는, 값 x_0 내지 x_7 은 중간값의 행을 나타낼 수도 있다. 값 X_0 내지 X_7 이 계수의 열에서의 값인 경우에는, 값 x_0 내지 x_7 은 시프팅된 픽셀 성분값의 열을 나타낼 수도 있다. 도 10 의 실시예에 예시된 바와 같이, "+" 심볼을 포위한 원은 가산 연산을 나타내고, "X" 심볼을 포위한 원은 승산 연산을 나타낸다. 문자/숫자 조합은, 일 값이 승산되는 값을 나타낸다. 예를 들어, 라인 x_2 상에서, "A1" 심볼이 "X" 심볼을 포위한 원 위에 위치된다. 이는, 라인 X_2 상의 값이 값 A1 과 승산된다는 것을 나타낸다.

[0176] 변환 (260) 은 "버터플라이 구조 연산" 의 세트를 포함한다. "버터플라이 구조 연산" 은 소프트웨어 또는 하드웨어 구조일 수도 있는데, 여기서 제 1 입력값 x 와 제 1 인수 C 를 승산하여 제 1 중간값을 생성하고, 제 2 입력값 y 와 제 2 인수 S 를 승산하여 제 2 중간값을 생성한 다음, 제 1 중간값과 제 2 중간값을 가산함으로써, 제 1 출력값 u 가 계산된다. 이러한 보다 복잡한 "버터플라이 구조 연산" 에 있어서, 제 2 입력값 y 와 제 1 인수 C 를 승산하여 제 3 중간값을 생성하고, 제 1 입력값 x 와 제 2 인수 S 를 승산하여 제 4 중간값을 생성한 다음, 제 4 중간값으로부터 제 3 중간값을 감산함으로써, 제 2 출력값 v 가 계산될 수도 있다. 다음의 공식은, "버터플라이 구조 연산" 의 수학적 결과를 요약한다:

$$u = x * C + y * S;$$

[0177] $$v = x * S - y * C.$$

[0178] 버터플라이 구조 연산 (264) 은 변환 (260) 에 이용된 일 버터플라이 구조 연산을 나타낸다. 버터플라이 구조 연산은, 제 1 입력값으로부터 제 2 입력값으로 및 제 2 입력값으로부터 제 1 입력값으로의 값의 크로스오버 때문에 버터플라이 구조 연산이 2 개의 "날개" 를 갖는 것처럼 보인다는 사실로 인해 이러한 명칭을 갖는다.

[0179] 변환 (260) 에 이용된 바와 같이, 제 1 인수 C 및 제 2 인수 S 는 무리수일 수도 있다. 예를 들어, C 는 $\sqrt{2}\cos(3\pi/8)$ 와 동등할 수도 있고, S 는 $\sqrt{2}\cos(3\pi/8)$ 와 동등할 수도 있다. 역방향 벡터 변환 모듈 (146) 이 16-비트 레지스터에서 고정 소수점 연산을 이용할 수도 있기 때문에, 또한 C 및 S 가 빈번하게 무리값이기 때문에, 입력값 x 및 y 와 C 및 S 를 승산하는 것은 계산상 불편할 수도 있다. 이러한 이유로, 역방향 벡터 변환 모듈 (146) 은 C 및 S 에 대한 유리수 근사치를 이용할 수도 있다. 이들 정수 근사치는 $(C'/2^j)$ 및 $(S'/2^k)$ 의 형태일 수도 있고, 여기서 C' 및 S' 는 C 및 S 의 "정수화 (integerized)" 버전이고, j 및 k 는 정수이다. 예를 들어, $C = \sqrt{2}\cos(3\pi/8)$ 이라 하자. 이 실시예에 있어서, 역방향 벡터 변

환 모듈 (146) 은, C' = 2217 및 j = 12 를 이용하여 $\sqrt{2}\cos(3\pi/8)$ 에 근사할 수도 있는데, 그 이유는 $2217/2^{12}$ = 0.541259766 이고, $\sqrt{2}\cos(3\pi/8) \approx 0.5411961....$ 이기 때문이다. 이 실시예에 있어서, 0.5411961 이 대략적으로 0.541259766 과 동등하다는 것은 명백하다. 따라서, 다음의 형태:

$$u = x * C + y * S;$$

[0180] $$v = x * S - y * C$$

[0181] 의 계산은, 다음의 형태:

$$u' = x * (C'/2^j) + y * (S'/2^k);$$

[0182] $$v' = x * (S'/2^k) - y * (C'/2^j)$$

[0183] 의 계산으로 대체될 수도 있다.

[0184] 이들 계산을 보다 단순화하기 위해서, 2^j 및 2^k 에 의한 제산 연산은 j 및 k 개의 포지션만큼 비트 단위의 (bitwise) 우측 시프트 연산으로 대체될 수도 있는데, 이는 ">>" 을 이용하여 다음과 같이 표시된다:

$$u'' = ((x * C') >> j) + ((y * S') >> k);$$

[0185] $$v'' = ((x * S') >> k) - ((y * C') >> j).$$

[0186] 그러나, DC 계수에 바이어스 값을 가산하는 것에 관하여 전술한 바와 같이, 제산 연산을 비트 단위의 우측 시프트 연산으로 대체하는 것은, u' 와 u'' 및 v' 와 v'' 사이의 차이를 야기할 수도 있다. 이들 계산에서 항에 중심점 바이어스 값을 가산하는 것은, u' 와 u'' 및 v' 와 v'' 사이의 차이를 감소시킬 수도 있다. 중심점 바이어스 값이 가산되는 경우, 계산은 다음의 형태를 취할 수도 있다:

[0187] $$u''' = ((x * C' + (1 << (j-1))) >> j) + ((y * S' + (1 << (k-1))) >> k);$$

[0188] $$v''' = ((x * S' + (1 << (k-1))) >> k) - ((y * C' + (1 << (j-1))) >> j).$$

[0189] 중심점 바이어스 값의 가산이 u 및 u'' 와 v 및 v'' 사이의 차이보다 작은 u 및 u''' 와 v 및 v''' 사이의 차이를 야기할 수도 있지만, 중심점 바이어스 값의 가산은 버터플라이 구조 연산에 대해 계산 복잡도를 부가할 수도 있다. 또한, 중심점 바이어스 값의 가산은, 16-비트 레지스터에서 고정 소수점 연산을 이용하는 경우에 u''' 및 v''' 의 계산을 비실용적으로 할 수도 있다. 중심점 바이어스 값의 가산이 우측 시프트 이전에 일어나고, 그 결과 레지스터 오버플로우를 야기할 수도 있기 때문에, 중심점 바이어스 값의 가산은 u''' 및 v''' 의 계산을 비실용적으로 할 수도 있다.

[0190] $v = x * S - y * C$ 와 $v'' = ((x * S') >> k) - ((y * C') >> j)$ 사이의 평균 차이는 대략 0 이다. 다시 말하면, 값 x 및 y 모두에 대한 값 (v'' - v) 모두의 평균은 대략 0 과 동등하다. 또한, $v = x * S - y * C$ 와 $v''' = ((x * S' + (1 << (k-1))) >> k) - ((y * C' + (1 << (j-1))) >> j)$ 사이의 평균 차이도 대략 0 이다. 이는, v'' 및 v''' 가 항상 대략 동등하기 때문이다. j 가 k 와 동등한 경우, 중심점 바이어스 값이, 감산에 의해 제거되기 때문에, v'' 및 v''' 는 항상 대략 동등하고:

$$v''' = ((x * C' + m) >> j) - ((y * S' + m) >> k) \approx$$

$$(x * C' + m)/2^j - (y * S' + m)/2^k =$$

$$(x * C')/2^j + (m/2^j) - (y * S')/2^k - (m/2^k) =$$

$$(x * C')/2^j - (y * S')/2^k \approx$$

[0191] $$v'' = ((x * C') >> j) - ((y * S') >> k)$$

[0192] 여기서, m 은 중심점 바이어스 값을 나타낸다. 이 실시예가 예시된 바와 같이, j 가 k 와 동등한 경우에

$(m/2^j)$ 로부터 $(m/2^k)$ 를 감산하는 것은 중심점 바이어스값 m 을 제거한다. v 와 v'' 사이의 평균 차이가 대략 0 이기 때문에, 역방향 벡터 변환 모듈 (146) 은, 고의로 v''' 대신에 v'' 을 계산함으로써 발생된 값으로 포지티브 또는 네거티브 바이어스를 도입하지 않고, 또한 v'' 및 v''' 가 대략 동등하기 때문에, 역방향 벡터 변환 모듈 (146) 은 v''' 대신에 v'' 을 이용할 수도 있다.

[0193] 또한, $u = x * C + y * S$ 와 $u''' = ((x * C' + (1 << (j-1))) >> j) + ((y * S' + (1 << (k-1))) >> k)$ 사이의 평균 차이도 대략 0 과 동등하다. 동시에, $u = x * C + y * S$ 와 $u'' = ((x * C') >> j) + ((y * S') >> k)$ 사이의 차이는 대략 0 과 동등하지는 않다. 오히려, u 와 u'' 사이의 평균 차이는 -1/2 이다. 따라서, u'' 및 u''' 는 대략 동등하지 않다. j 가 k 와 동등한 경우에도, 중심점 바이어스 값이 제거되지 않기 때문에, u'' 및 u''' 는 대략 동등하지 않다:

$$u''' = ((x * C' + m) >> k) + ((y * S' + m) >> k) \approx$$
$$((x * C' + m) / 2^j) + ((y * S' + m) / 2^k) =$$
$$((x * C') / 2^j) + (m / 2^j) + ((y * S') / 2^k) + (m / 2^k) =$$
$$((x * C') / 2^j) + ((y * S') / 2^k) + (m / 2^j) + (m / 2^k) \neq$$

[0194] $$u'' = ((x * C') >> j) + ((y * S') >> k).$$

[0195] u'' 가 u''' 와 대략 동등하지 않기 때문에, u'' 가 u''' 대신에 이용되지 않을 수도 있다. u''' 대신에 u'' 를 이용하려는 시도는, u 와의 상당한 차이를 야기할 수도 있다.

[0196] 모든 계산에 중심점 바이어스 값을 가산하는 것과 연관된 복잡도 및 오버플로우 이슈를 회피하기 위해서, 다음의 공식이 u''' 및 u'' 대신에 이용될 수도 있다:

[0197] $$u'''' = ((x * C') >> j) - ((y * -S') >> k).$$

[0198] u'''' 에서 S' 의 음수 버전이 이용되고, $((y * -S) >> k)$ 이 감산된다는 것을 제외하고, u'''' 은 u'' 와 동등하다. u''' 와 같이, u'''' 은 중심점 바이어스 값을 가산하지 않는다. 그러나, u''' 와 달리, u'''' 와 u 사이의 차이는 약 0 을 중심으로 한다. u'''' 와 u 사이의 차이가 약 0 을 중심으로 하기 때문에, 라운딩 에러는 후속 버터플라이 구조 연산의 적용을 통해 확대되지 않는다. 이러한 이유로, 역방향 벡터 변환 모듈 (146) 은 u'''' 을 이용하여, 버터플라이 구조 연산에서 u 를 계산할 수도 있다. 이러한 이유로, j 가 k 와 동등한 경우, 변환 (260) 을 적용하기 위해서 역방향 벡터 변환 모듈 (146) 에 의해 이용된 버터플라이 구조 연산은 다음의 형태를 가질 수도 있다:

$$u'''' = ((x * C') >> j) - ((y * -S') >> k);$$

[0199] $$v''' = ((x * S') >> k) - ((y * C') >> j).$$

[0200] 그러므로, 이러한 타입의 버터플라이 구조 연산에 의해 발생된 결과와, 무한의 정밀도 연산을 이용하여 등가의 버터플라이 구조 연산에 의해 발생되는 결과 사이의 차이는 약 0 을 중심으로 하며, 1 이하의 포지티브 또는 네거티브 크기를 갖는다.

[0201] 대안적으로, j 가 k 와 동등한 경우, 역방향 벡터 변환 모듈 (146) 은, 버터플라이 구조 연산의 말단으로 우측 시프트가 연기된 버터플라이 구조 연산을 이용할 수도 있다:

[0202] $$u' = x * (C' / 2^k) + y * (S' / 2^k);$$

$$((x * C') / 2^k) + ((y * S') / 2^k) =$$

$$((x * C') + (y * S')) / 2^k \approx$$

$$u^* = \quad (((x * C') + (y * S') + (1 << (k\text{-}1))) >> k;$$

$$v' = \quad x * (S' / 2^k) - y * (C' / 2^j) =$$

$$((x * S') / 2^k) - ((y * C') / 2^k) =$$

$$((x * S') / 2^k) + ((-1)(y * C') / 2^k) =$$

$$((x * S') / 2^k) + ((y * -C') / 2^k) =$$

$$((x * S') + (y * -C')) / 2^k \approx$$

[0203] $$v^* = \quad ((x * S') + (y * -C') + (1 << (k\text{-}1))) >> k.$$

[0204] 버터플라이 구조 연산의 말단으로의 우측 시프트의 연기는, 버터플라이 구조 연산을 수행하는데 필요한 시프트 연산의 총 수를 감소시킬 수도 있고, 정확성을 향상시킬 수도 있다. 또한, 가장 최신의 16-비트 단일 명령 복수 데이터 ("SIMD") 프로세서 및 디지털 신호 프로세서 상에서 이용가능한 4-방향 승산-누산 명령들이 효율적으로 u^* 및 v^* 를 컴퓨팅하는데 이용될 수도 있다.

[0205] 승산 연산은, 가산, 감산, 및 비트 단위의 시프트 연산에 비해 계산상 고비용이 소요될 수도 있다. 이러한 이유로, 승산 연산과 동일한 효과를 갖는 일련의 가산, 감산, 및 비트 단위의 시프트 연산을 이용하는 것이 보다 단순할 수도 있다. 예를 들어, C' = 2217 이라고 가정한다. 이 실시예에 있어서, $r = x * C'$은 다음의 단계로 대체될 수도 있다:

$$x2 = (x << 3) - x;$$

$$x3 = x + (x2 << 6);$$

$$x4 = x3 - x2;$$

[0206] $$r = x3 + (x4 << 2).$$

[0207] 이 실시예에 있어서, x2, x3 및 x4 는 중간값이다. 이를 예시하기 위해서, x = 1 인 실시예를 고려하라:

$$7 = (1 << 3) - 1;$$

$$449 = 1 + (7 << 6);$$

$$442 = 449 - 7;$$

$$2217 = 449 + (442 << 2).$$

[0208] $$2217 = (1 * 2217) = (x * 2217, \text{여기서 } x = 1).$$

[0209] 이 실시예에 있어서, 1*2217 = 2217 이고, x =1 인 경우에 이 시퀀스 연산에 의해 생성된 값은 2217 이다.

[0210] 다음의 표는, 역방향 벡터 변환 모듈 (146) 이 상수 인수 A1, A2, B1, B2, C1, C2, D1 및 D2 에 이용할 수도 있는 정수값 근사치의 예시적인 세트를 요약한다.

표 1

인수	오리지널 값	정수값	다음의 곱을 컴퓨팅하기 위한 알고리즘: **x=x*F1, y=x*F2:**	복잡도
A1	$\sqrt{2}\cos(3\pi/8)$	2217	x2 = (x << 3) - x, x3 = x + (x2 << 6), x4 = x3 - x2, (2217 * x) = x3 + (x4 << 2); (5352 * x) * 2 = x4 + (x2 << 7);	5회의 가산 4회의 시프트
A2	$\sqrt{2}\sin(3\pi/8)$	5352		
B1	$\sqrt{2}\cos(5\pi/16)$	3218	x2 = x + (x << 3), x3 = x2 + (x << 4), x4 = (x3 << 5) – x, (3218 * x) * 2 = x2 + (x3 << 6); (4816 * x) * 2 = x4 + x*2;	5회의 가산 4회의 시프트
B2	$\sqrt{2}\sin(5\pi/16)$	4816		
C1	$\sqrt{2}(\cos(\pi/16)-\cos(5\pi/16))$	2463	x2 = x + (x << 4), x3 = (x2 << 3) – x, x4 = x3 + (x2 << 6), x5 = x2 + (x4 << 1) (2463 * x) = x5; (3686 * x) = x4 + x5;	5회의 가산 4회의 시프트
C2	$\sqrt{2}(\sin(5\pi/16)-\sin(\pi/16))$	3686		
D1	$\sqrt{2}(\cos(5\pi/16)-\sin(\pi/16))$	2088	x2 = x + (x << 6), x3 = x2 << 5, x4 = x2 + x3, x5 = x3 + (x << 3) (2088 * x) = x5; (10497 * x) = x4 + (x5 << 2)	4회의 가산 4회의 시프트
D2	$\sqrt{2}(\cos(\pi/16)+\sin(5\pi/16))$	10497		

[0211]

[0212] **변환 (260) 에 이용된 예시적인 상수 인수 근사치**

[0213] 표 1 에 있어서, "정수값" 열에서의 값은, 2^{12} = 4096 으로 제산되는 경우에 "오리지널 값" 열에서의 값에 근사한다. 예를 들어, 2217/4096 = 0.541259766 이고, $\sqrt{2}\cos(3\pi/8) \approx 0.5411961$ 이다. 유사하게, 5352/4096 = 1.30664062 이고, $\sqrt{2}\sin(3\pi/8) \approx 1.30656296$ 이다. "다음의 곱을 컴퓨팅하기 위한 알고리즘" 열에서의 공식은, 역방향 벡터 변환 모듈 (146) 이 대응하는 정수값에 의한 승산 연산 대신에 이용할 수도 있는 방법을 요약한다.

[0214] 도 11 은 제 2 예시적인 변환 (270) 을 도시한 도면이다. 도 10 에서와 같이, 값 X_0, X_1, X_2, X_3, X_4, X_5, X_6, 및 X_7 은 입력 계수를 나타내고, 값 x_0, x_1, x_2, x_3, x_4, x_5, x_6, 및 x_7 은 출력값을 나타낸다. "+" 심볼을 포위한 원 다음에 라인과 연관된 값은, 이 원으로 향하는 화살표와 연관된 값을 가산한 결과이다. "x" 심볼을 포위한 원 다음에 라인과 연관된 값은, 이 원을 통과하는 라인과 연관된 값과, 이 원의 옆에 위치된 계수를 승산한 결과이다. 화살표 옆의 심볼 "-" 은 이 화살표와 연관된 값의 음수화 (negation) 를 나타낸다.

예를 들어, 값 "10" 이 "-" 심볼 이전에 화살표와 연관되는 경우, 값 "-10" 이 "-" 심볼 이후에 이 화살표와 연관된다. 또한, 감산 및 음수화된 계수를 이용하여 라운딩 에러를 감소시키는 전술한 기술이 변환 (270) 에서 이용될 수도 있다는 것이 주목되어야 한다.

[0215] 변환 (270) 에 있어서, $\alpha = \sqrt{2}\cos(3\pi/8)$, $\beta = \sqrt{2}\sin(3\pi/8)$, $\gamma = \sqrt{2}\cos(\pi/16)$, $\delta = \sqrt{2}\sin(\pi/16)$, $\varepsilon = \sqrt{2}\cos(3\pi/16)$, 및 $\zeta = \sqrt{2}\sin(3\pi/16)$ 의 값이 유리 분수를 이용하여 근사될 수도 있다. 예를 들어, α, β, γ, δ, ε, 및 ζ 의 값은 아래의 표 2 에 열거된 정수 근사치를 이용하

여 근사될 수도 있다.

표 2

인수	오리지널 값	정수 근사치	다음의 곱을 컴퓨팅하기 위한 알고리즘: **x=x*F1, y=x*F2:**	복잡도
α	$\sqrt{2}\cos(3\pi/8)$	8867/16384	x2 = (x << 7) - x, x3 = x2 - (x << 5), x4 = (x3 << 5) - x2, α * x ≈ ((x4 + (x3 <<4)) <<1) + x, β * x =(x3 << 8) - x4	6회의 가산 6회의 시프트
β	$\sqrt{2}\sin(3\pi/8)$	21407/16384		
sqrt2	$1/\sqrt{2}$	11585/16384	x2 = -x + (x << 3), x3 = x2 << 7, x4 = x3 - x2, x5 = x4 + (x3 << 2), $1/\sqrt{2}$ * x = x5 + (x4 <<3),	4회의 가산 4회의 시프트
γ	$\sqrt{2}\cos(\pi/16)$	5681/4096	x2=x + (x << 7), x3=x2 - (x << 4), y=x3 + (x3 << 2), (δ * x) = y x5=x2+y, (γ * x) = x2 + (x5 << 3)	5회의 가산 4회의 시프트
δ	$\sqrt{2}\sin(\pi/16)$	565/2048		
ε	$\sqrt{2}\cos(3\pi/16)$	9633/8192	x2 = x + (x << 5), x3 = x2 - (x << 3), x4 = x2 + (x3 << 8), (ζ * x) = ((x4 + (x << 2)) << 1) - x, (ε * x) = x4 + (x3 << 7)	6회의 가산 6회의 시프트
ζ	$\sqrt{2}\sin(3\pi/16)$	12873/16384		

[0216]

[0217] **변환 (270) 에 이용된 예시적인 상수값 근사치**

[0218] 표 2 에 있어서, "정수 근사치" 열에서의 값은, "오리지널 값" 열에서의 값의 정수화 버전을 나타낸다. 예를 들어, $\sqrt{2}\cos(3\pi/8) \approx 0.5411961$ 이고, 8867/16384 = 0.54119873046875 이다. 표 2 의 "다음의 곱을 컴퓨팅하기 위한 알고리즘" 열에서의 수학식은, 역방향 벡터 변환 모듈 (146) 이 "정수 근사치" 열에서의 값에 의한 승산 연산 대신에 이용할 수도 있는 알고리즘을 나타낸다.

[0219] 도 12 는 순방향 벡터 변환 모듈 (214) 에 의해 이용될 수도 있는 예시적인 변환 (280) 을 도시한 흐름도이다. 도 12 에 있어서, 값 X_0, X_1, X_2, X_3, X_4, X_5, X_6, 및 X_7 은 출력 계수를 나타내고, 값 x_0, x_1, x_2, x_3, x_4, x_5, x_6, 및 x_7 은 입력값을 나타낸다. 또한, 감산 및 음수화된 계수를 이용하여 라운딩 에러를 감소시키는 전술한 기술이 변환 (280) 에서 이용될 수도 있다는 것이 주목되어야 한다.

[0220] 도 12 의 실시예에 있어서, $\alpha = \sqrt{2}\cos(3\pi/8)$, $\beta = \sqrt{2}\sin(3\pi/8)$, $\gamma = \sqrt{2}\cos(\pi/16)$, $\delta = \sqrt{2}\sin(\pi/16)$, $\varepsilon = \sqrt{2}\cos(3\pi/16)$, 및 $\zeta = \sqrt{2}\sin(3\pi/16)$ 의 값은 유리 분수를 이용하여 근사될 수도 있다. 예를 들어, α, β, γ, δ, ε, 및 ζ 의 값은 표 2 에 열거된 정수 근사치를 이용하여 근사될 수도 있다.

[0221] 본 명세서에 기재된 기술은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 모듈 또는 컴포넌트로서 기재된 임의의 특징은 집적 논리 디바이스에 함께 또는 상호운용가능한 이산 논리 디바이스로서 개별적으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이들 기술은, 실행시, 전술한 방법 중 하나 이상을 수행하는 명령들을 포함한 컴퓨터 판독가능 매체에 의해 적어도 부분적으로 실현될 수도 있다. 컴퓨터 판독가능 매체는, 포장재를 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다.

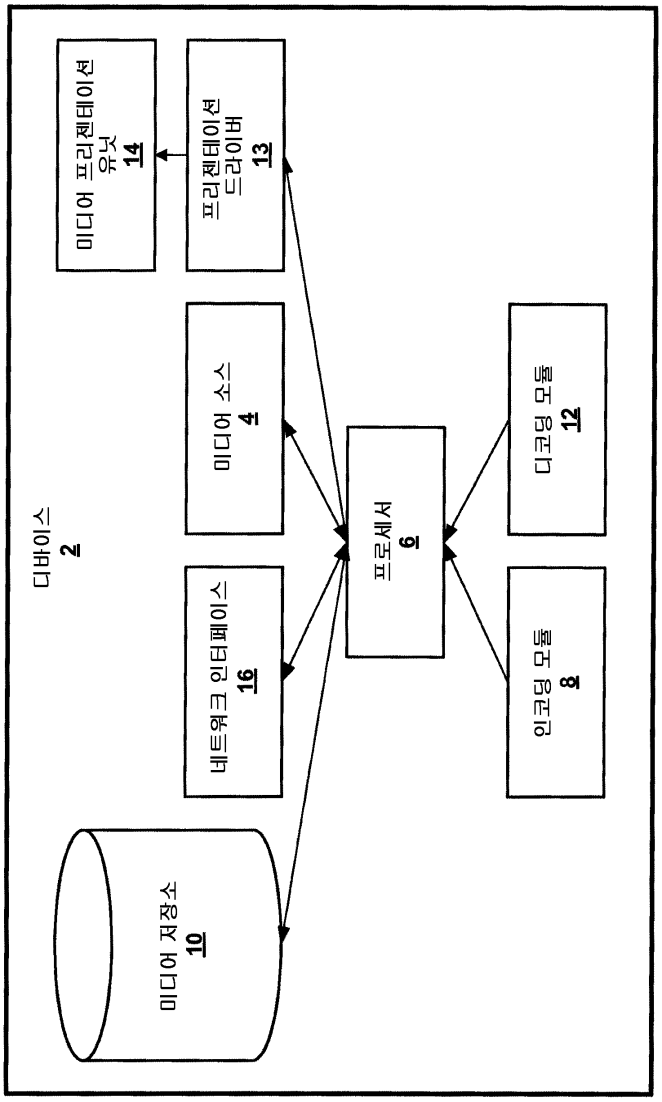
컴퓨터 판독가능 매체는, SDRAM (Synchronous Dynamic RAM) 과 같은 RAM (Random Access Memory), ROM (Read Only Memory), NVRAM (Non-Volatile RAM), EEPROM (Electrically Erasable Programmable ROM), 플래시 메모리, 자기 또는 광학 데이터 저장 매체 등을 포함할 수도 있다. 부가적으로 또는 대안적으로, 이들 기술은, 명령들 또는 데이터 구조체의 형태로 코드를 반송 또는 통신하며, 컴퓨터에 의해 액세스, 판독, 및/또는 실행될 수 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0222] 코드는, 하나 이상의 디지털 신호 프로세서 (DSP), 범용 마이크로프로세서, 주문형 집적 회로 (ASIC), 필드 프로그래머블 게이트 어레이 (FPGA), 또는 다른 등가의 집적 또는 이산 논리 회로와 같은 하나 이상의 프로세서에 의해 실행될 수도 있다. 따라서, 본 명세서에 이용된 바와 같은 "프로세서" 라는 용어는, 임의의 전술한 구조체 또는 본 명세서에 기재된 기술의 구현에 적합한 임의의 다른 구조체를 언급할 수도 있다. 또한, 몇몇 양태에 있어서, 본 명세서에 기재된 기능성은, 인코딩 및 디코딩을 위해 구성된 전용 소프트웨어 모듈이나 하드웨어 모듈 내에 제공될 수도 있고, 또는 결합된 비디오 인코더-디코더 (CODEC) 에 통합될 수도 있다.

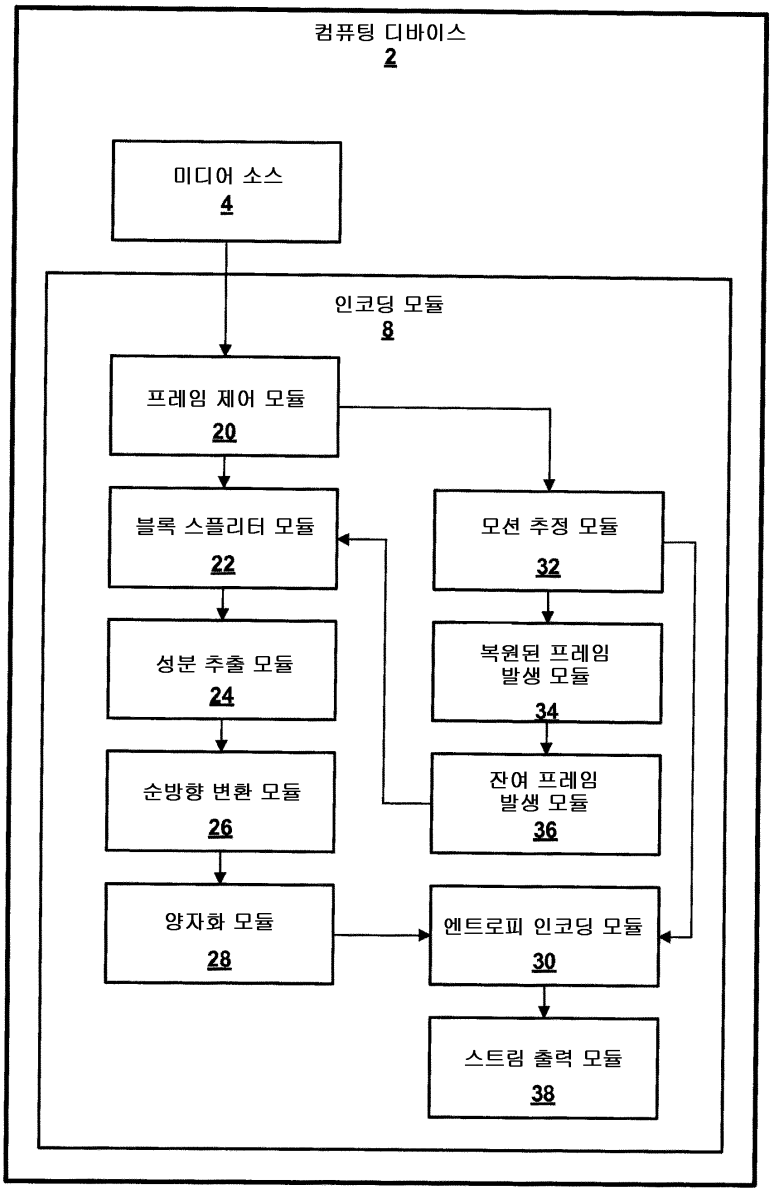
[0223] 본 발명의 각종 실시형태가 전술되었다. 이들 실시형태 및 다른 실시형태는 다음의 특허청구범위의 범위 내에 있다.

도면

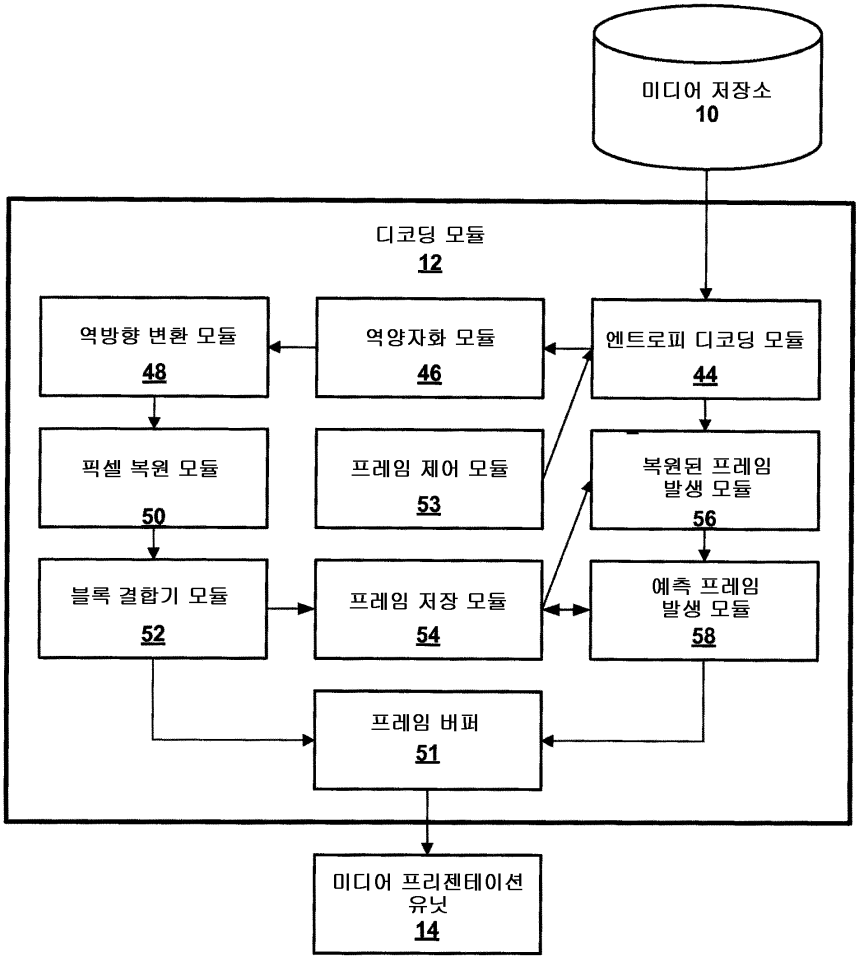
도면1



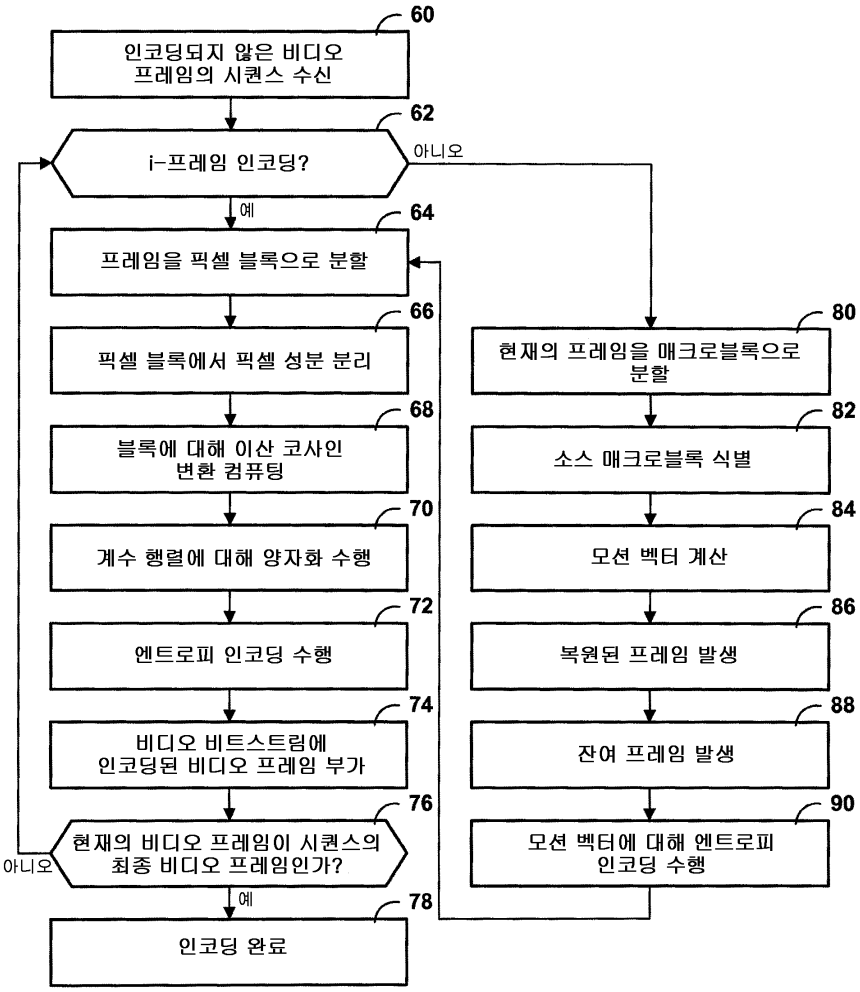
도면2


컴퓨팅 디바이스
2
미디어 소스
4
인코딩 모듈
8
프레임 제어 모듈
20
블록 스플리터 모듈
22
성분 추출 모듈
24
순방향 변환 모듈
26
양자화 모듈
28
모션 추정 모듈
32
복원된 프레임
발생 모듈
34
잔여 프레임
발생 모듈
36
엔트로피 인코딩 모듈
30
스트림 출력 모듈
38

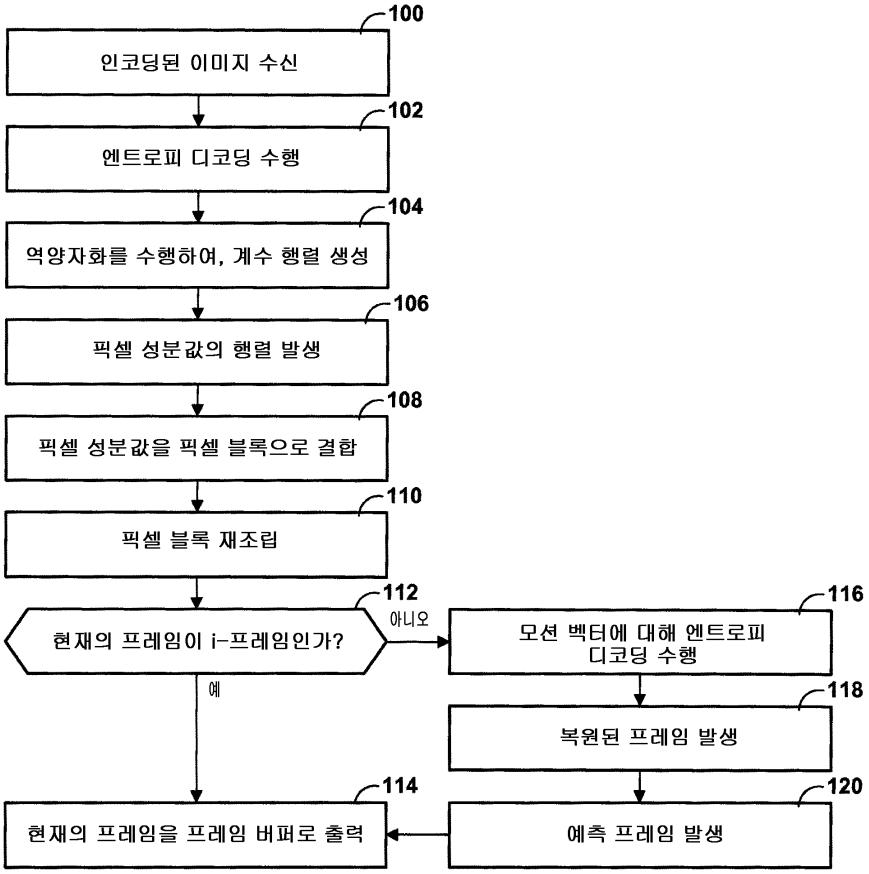
도면3


미디어 저장소
10
디코딩 모듈
12
역방향 변환 모듈
48
역양자화 모듈
46
엔트로피 디코딩 모듈
44
픽셀 복원 모듈
50
프레임 제어 모듈
53
복원된 프레임
발생 모듈
56
블록 결합기 모듈
52
프레임 저장 모듈
54
예측 프레임
발생 모듈
58
프레임 버퍼
51
미디어 프리젠테이션
유닛
14

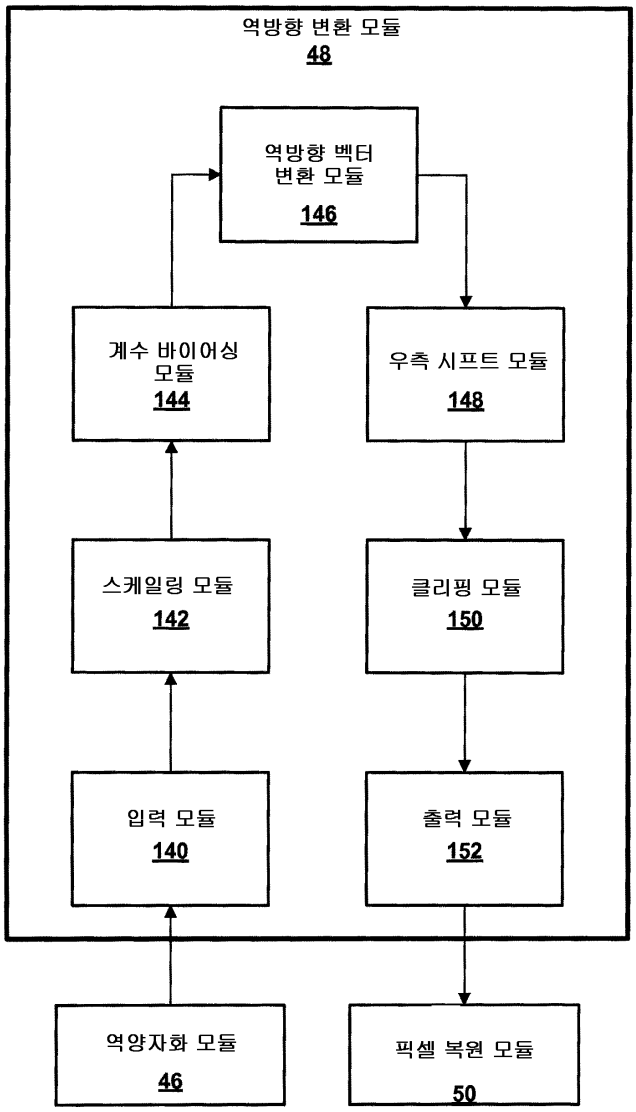
도면4


60
인코딩되지 않은 비디오 프레임의 시퀀스 수신
62
i-프레임 인코딩?
아니오
예
64
프레임을 픽셀 블록으로 분할
66
픽셀 블록에서 픽셀 성분 분리
68
블록에 대해 이산 코사인 변환 컴퓨팅
70
계수 행렬에 대해 양자화 수행
72
엔트로피 인코딩 수행
74
비디오 비트스트림에 인코딩된 비디오 프레임 부가
76
현재의 비디오 프레임이 시퀀스의 최종 비디오 프레임인가?
아니오
예
78
인코딩 완료
80
현재의 프레임을 매크로블록으로 분할
82
소스 매크로블록 식별
84
모션 벡터 계산
86
복원된 프레임 발생
88
잔여 프레임 발생
90
모션 벡터에 대해 엔트로피 인코딩 수행

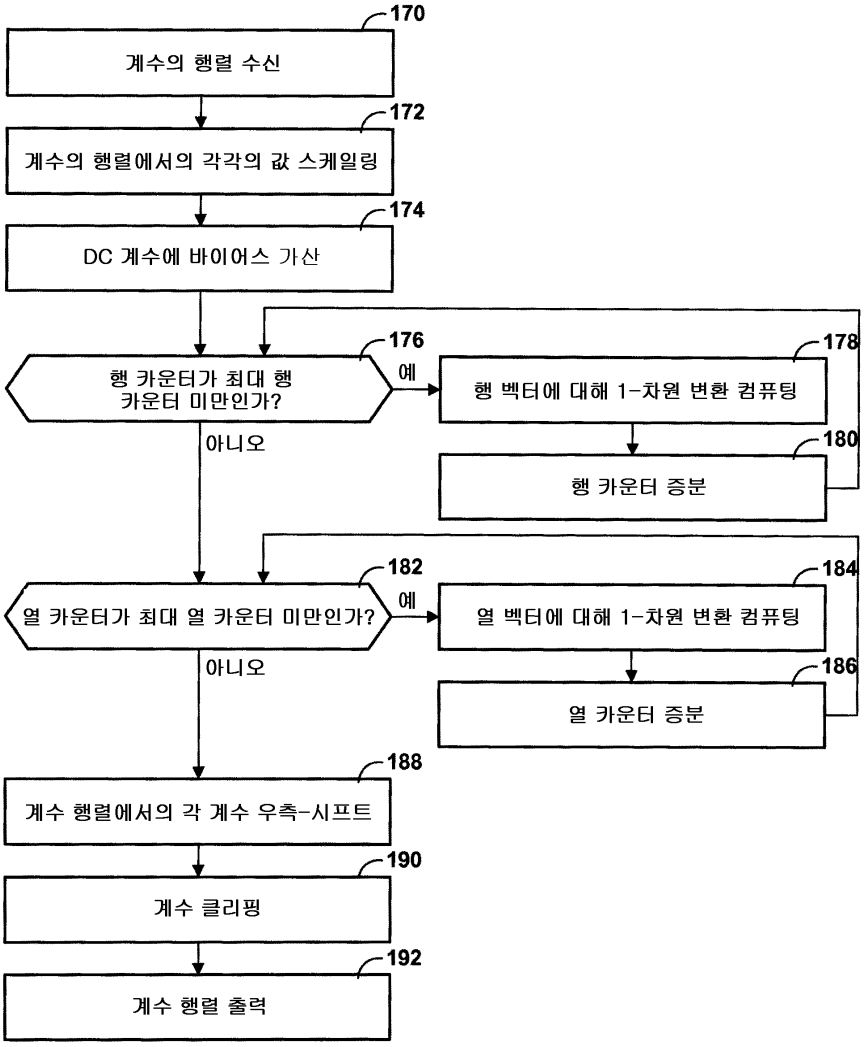
도면5


100
인코딩된 이미지 수신
102
엔트로피 디코딩 수행
104
역양자화를 수행하여, 계수 행렬 생성
106
픽셀 성분값의 행렬 발생
108
픽셀 성분값을 픽셀 블록으로 결합
110
픽셀 블록 재조립
112
현재의 프레임이 i-프레임인가?
아니오
예
116
모션 벡터에 대해 엔트로피 디코딩 수행
118
복원된 프레임 발생
120
예측 프레임 발생
114
현재의 프레임을 프레임 버퍼로 출력

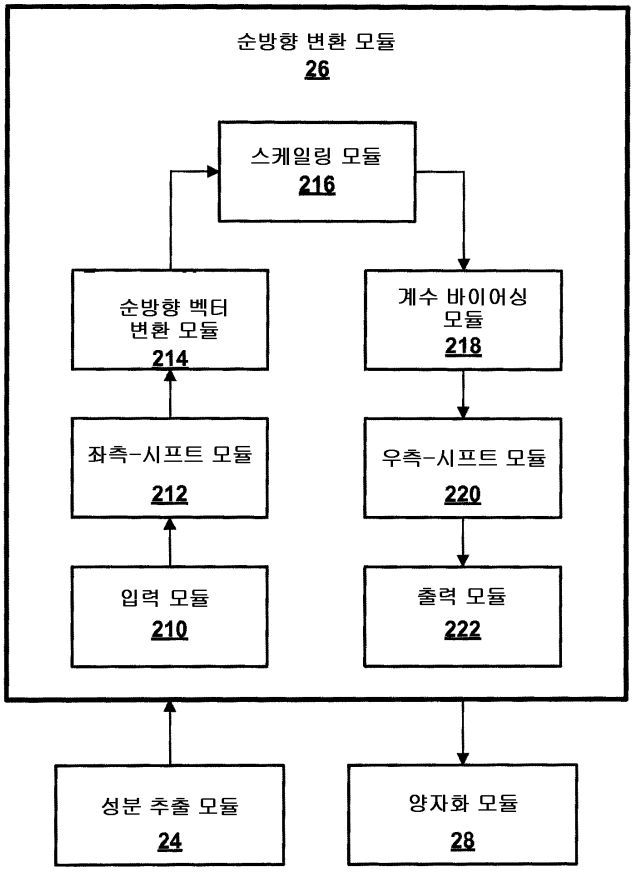
도면6


역방향 변환 모듈
48
역방향 벡터 변환 모듈
146
계수 바이어싱 모듈
144
우측 시프트 모듈
148
스케일링 모듈
142
클리핑 모듈
150
입력 모듈
140
출력 모듈
152
역양자화 모듈
46
픽셀 복원 모듈
50

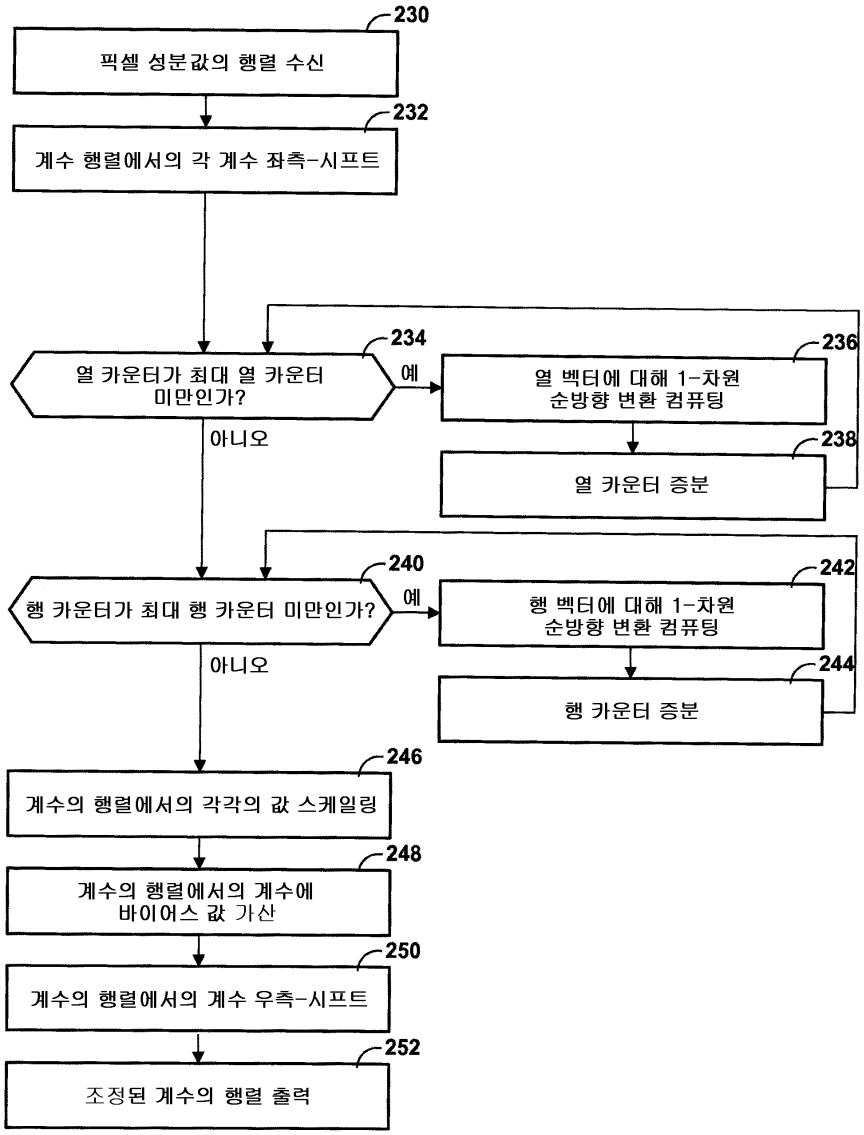
도면7


170
계수의 행렬 수신
172
계수의 행렬에서의 각각의 값 스케일링
174
DC 계수에 바이어스 가산
176
행 카운터가 최대 행 카운터 미만인가?
예
178
행 벡터에 대해 1-차원 변환 컴퓨팅
아니오
180
행 카운터 증분
182
열 카운터가 최대 열 카운터 미만인가?
예
184
열 벡터에 대해 1-차원 변환 컴퓨팅
아니오
186
열 카운터 증분
188
계수 행렬에서의 각 계수 우측-시프트
190
계수 클리핑
192
계수 행렬 출력

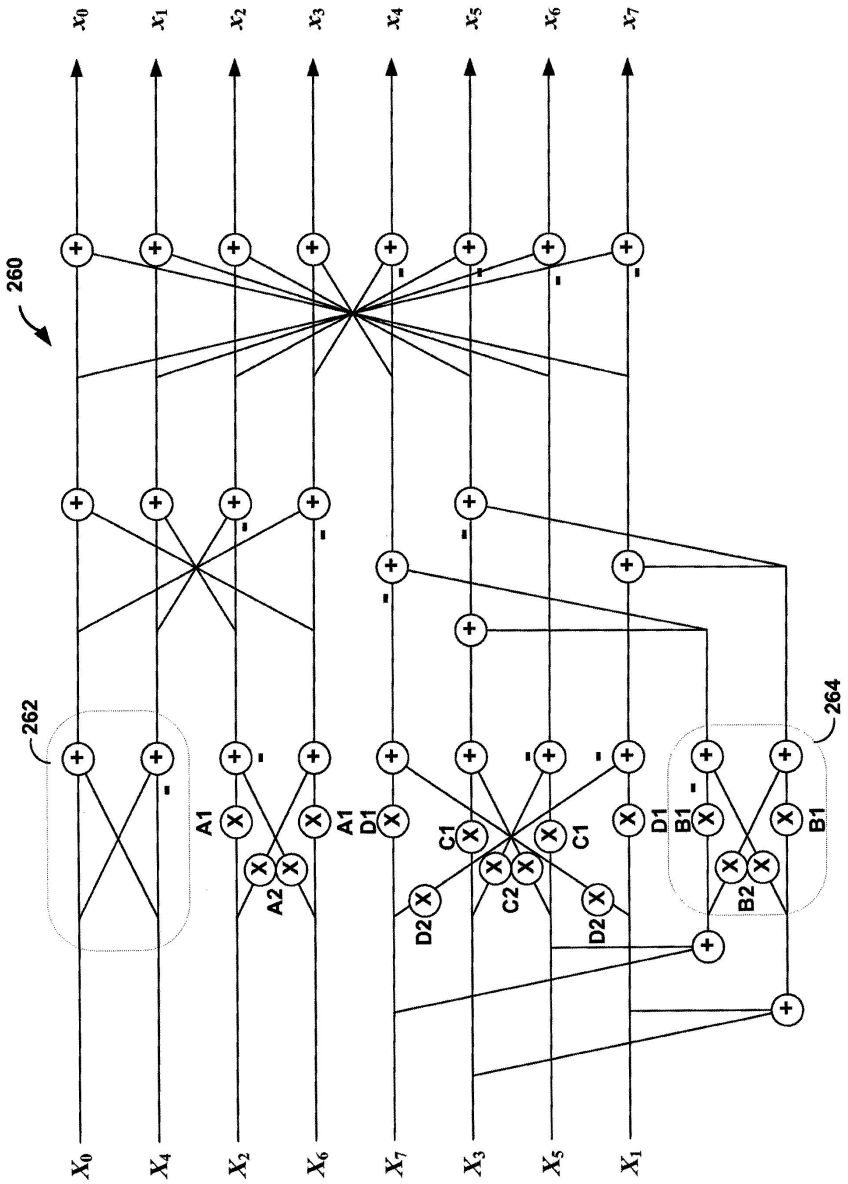
도면8


순방향 변환 모듈
26
스케일링 모듈
216
순방향 벡터
변환 모듈
214
계수 바이어싱
모듈
218
좌측-시프트 모듈
212
우측-시프트 모듈
220
입력 모듈
210
출력 모듈
222
성분 추출 모듈
24
양자화 모듈
28

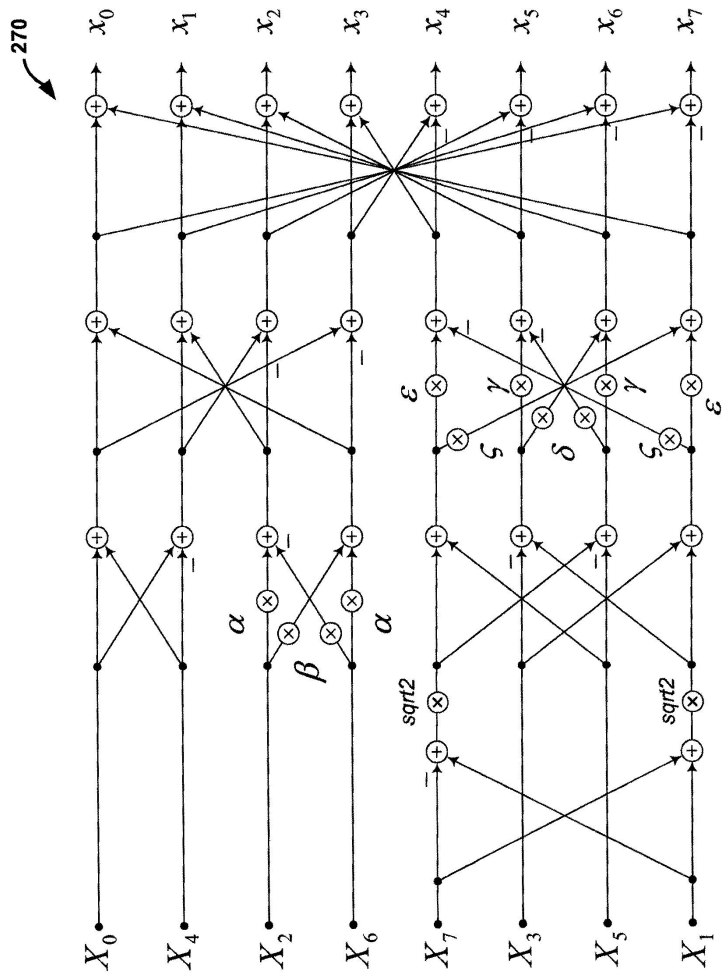
도면9


230
픽셀 성분값의 행렬 수신
232
계수 행렬에서의 각 계수 좌측-시프트
234
열 카운터가 최대 열 카운터 미만인가?
예
236
열 벡터에 대해 1-차원 순방향 변환 컴퓨팅
238
열 카운터 증분
아니오
240
행 카운터가 최대 행 카운터 미만인가?
예
242
행 벡터에 대해 1-차원 순방향 변환 컴퓨팅
244
행 카운터 증분
아니오
246
계수의 행렬에서의 각각의 값 스케일링
248
계수의 행렬에서의 계수에 바이어스 값 가산
250
계수의 행렬에서의 계수 우측-시프트
252
조정된 계수의 행렬 출력

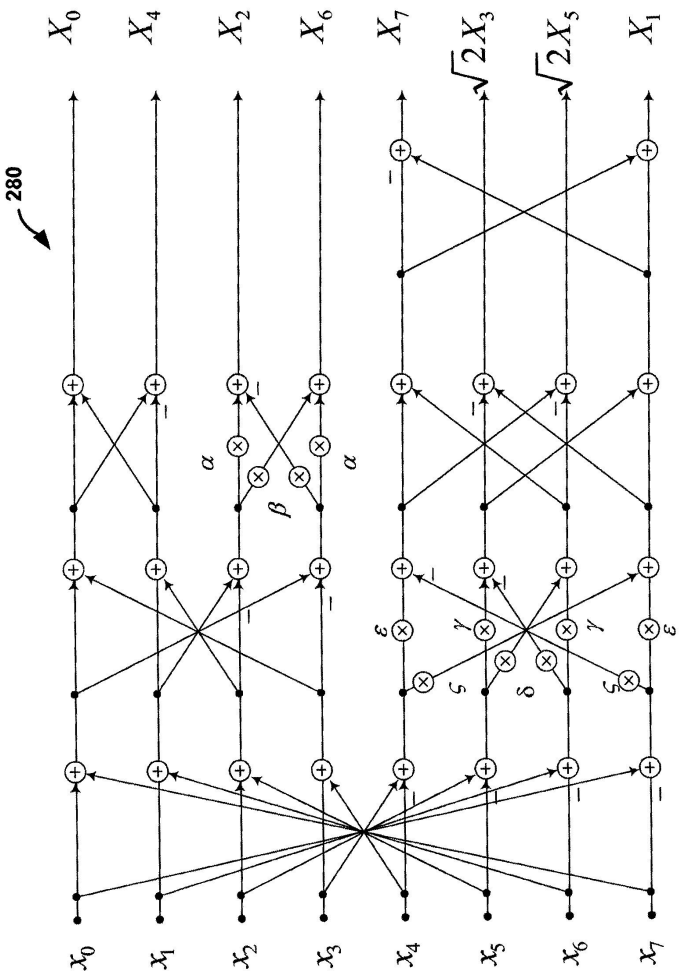
도면10


x0
x1
x2
x3
x4
x5
x6
x7
260
262
264
A1
A2
D1
C1
C2
D2
B1
B2
X0
X4
X2
X6
X7
X3
X5
X1

도면11


270
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7
X_0 X_4 X_2 X_6 X_7 X_3 X_5 X_1
α β α
ε γ δ γ ε
ζ ζ
sqrt2 sqrt2

도면12


280
X_0
X_4
X_2
X_6
X_7
$\sqrt{2}X_3$
$\sqrt{2}X_5$
X_1
x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
α
β
γ
δ
ε
ζ