



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년05월24일
 (11) 등록번호 10-1036711
 (24) 등록일자 2011년05월17일

(51) Int. Cl.

G06F 7/52 (2006.01)

(21) 출원번호 10-2005-0107885
 (22) 출원일자 2005년11월11일
 심사청구일자 2010년10월15일
 (65) 공개번호 10-2006-0069256
 (43) 공개일자 2006년06월21일
 (30) 우선권주장 11/015,148 2004년12월17일 미국(US)
 (56) 선행기술조사문헌 KR10200600083125 A

(73) 특허권자

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이

(72) 발명자

투, 쉐지에

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
 웨이마이크로소프트 코포레이션 내

말바르, 엔리케 사르멘토

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
 웨이마이크로소프트 코포레이션 내

스리니바산, 스리드하르

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
 웨이마이크로소프트 코포레이션 내

(74) 대리인

주성민, 이중희, 백만기

전체 청구항 수 : 총 35 항

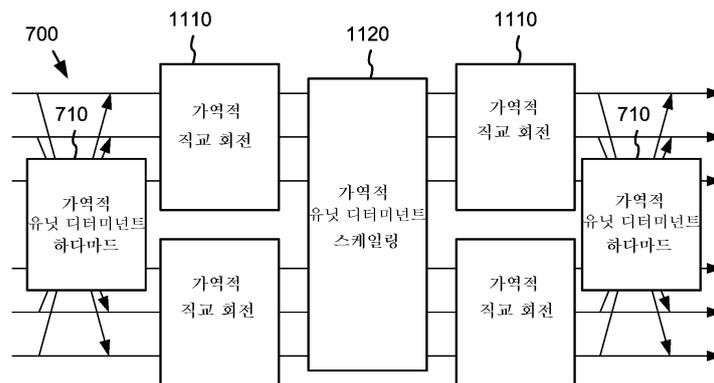
심사관 : 최정권

(54) 효과적인 무손실 데이터 압축을 위한 가역적 오버랩 연산자

(57) 요약

유닛 디터미넌트 컴포넌트 행렬(unit determinant component matrices)로 구성된 전필터들(pre-filters)과 후필터들(post-filters)(또는 가역적 오버랩 연산자들(reversible overlap operators))을 사용하여 효과적인 겹침 변환(lapped transform)이 실현된다. 전필터들과 후필터들은 평면 회전 변환들(planar rotational transforms)과 유닛 디터미넌트 평면 스케일링 변환들(planar scaling transforms)의 연속으로서 실현될 수 있다. 평면 스케일링 변환들은 평면 쉬어(planar shears) 또는 리프팅(lifting) 단계들을 사용하여 구현될 수 있다. 또한, 평면 회전과 평면 쉬어는 가역적/무손실(lossless) 연산들로서 구현되어, 결과로서, 가역적 오버랩 연산자를 제공한다.

대표도 - 도11



특허청구의 범위

청구항 1

블록들로 타일링된(tiled) 픽처 데이터(picture data)를 디코딩하는 방법으로서, 상기 방법은 컴퓨터 프로세서에 의해 수행되며,

압축된 비트스트림(compressed bitstream)으로부터 픽처 데이터의 블록들을 검색하는 단계;

상기 블록들에 가역적 블록 변환을 적용하는 단계;

블록들 간의 경계들 중 적어도 일부에 걸쳐 가역적 오버랩 연산자(reversible overlap operator)를 적용하는 단계 - 상기 가역적 오버랩 연산자는 유닛 디터미넌트(unit determinant)임 - ; 및

디스플레이에 표시(presentation)하기 위해 상기 픽처 데이터로부터 픽처를 생성하는 단계를 포함하는, 디코딩 방법.

청구항 2

제1항에 있어서,

상기 가역적 오버랩 연산자는 각각이 유닛 디터미넌트인 복수 개의 컴포넌트로 구조화되는, 디코딩 방법.

청구항 3

제2항에 있어서,

상기 가역적 오버랩 연산자를 적용하는 단계는,

가역적 하다마드 네트워크(reversible Hadamard network)를 적용하는 단계;

가역적 블록 회전(reversible block rotation)을 적용하는 단계;

가역적 스케일링 연산자(reversible scaling operator)를 적용하는 단계;

다른 가역적 블록 회전을 적용하는 단계; 및

가역적 인버스 하다마드 네트워크(reversible inverse Hadamard network)를 적용하는 단계를 포함하는, 디코딩 방법.

청구항 4

제3항에 있어서,

상기 가역적 스케일링 연산자를 적용하는 단계는,

제1 값과, 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 비율을 제2 값에 곱한 값을 합하는 제1 리프팅 단계(lifting step)를 수행하는 단계;

상기 제2 값과, 상기 제1 합에 상기 파라미터 값을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;

상기 제1 합과, 상기 제2 합에 상기 파라미터 값의 음수(negative)를 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계; 및

상기 제2 합과, 상기 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 상기 비율의 음수를 상기 제3 합에 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계를 포함하는, 디코딩 방법.

청구항 5

제3항에 있어서,

상기 가역적 스케일링 연산자를 적용하는 단계는,

제2 값과 상기 제1 값을 합하는 제1 리프팅 단계를 수행하는 단계;

상기 제1 값과, 상기 제1 합에 $\frac{1}{2}\left(\frac{s}{1+c}-1\right)$, 여기서 $s^2+c^2=1$, 을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;

상기 제1 합과, 상기 제2 합에 $\frac{2s}{c}$ 을 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계;

상기 제2 합과, 상기 제3 합에 $\frac{1}{2}\left(\frac{s}{1+c}+1\right)$ 을 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계; 및

상기 제3 합과 상기 제4 합의 음수를 합하는 제5 리프팅 단계를 수행하는 단계를 포함하는 5-리프팅 단계 프로시저를 수행하는 단계를 포함하는, 디코딩 방법.

청구항 6

제5항에 있어서,

상기 리프팅 단계들 중 한 개 이상은 상기 가역적 오버랩 연산자의 다른 단계와 결합되거나 또는 상기 다른 단계에 의해 취소되는(cancelled), 디코딩 방법.

청구항 7

제5항에 있어서,

상기 가역적 스케일링 연산자를 적용하는 단계는 2-포인트 스케일링의 캐스케이드(cascade)를 사용하여 s_1 에서 s_M 까지 M 개의 스케일링 인자들에 의해 M 데이터 경로들을 스케일링하는 단계를 포함하는, 디코딩 방법.

청구항 8

제2항에 있어서,

상기 가역적 오버랩 연산자를 적용하는 단계는,

가역적 하다마드 네트워크를 적용하는 단계;

가역적 블록 회전을 적용하는 단계;

가역적 스케일링 연산자를 적용하는 단계; 및

가역적 인버스 하다마드 네트워크를 적용하는 단계를 포함하는, 디코딩 방법.

청구항 9

픽처 인코더 및 디코더 장치로서,

인코드 및 디코드될 픽처 데이터를 저장하는 데이터 저장 버퍼; 및

프로세서를 포함하고,

상기 프로세서는,

상기 픽처 데이터를 블록들로 타일링하고,

블록들 간의 경계들 중 적어도 일부에 걸쳐 가역적 오버랩 연산자를 적용하고 - 상기 가역적 오버랩 연산자는 유닛 디터미넌트임 -,

상기 블록들에 가역적 블록 변환을 적용하도록 프로그램되고,

상기 가역적 오버랩 연산자와 상기 가역적 블록 변환의 적용은 상기 픽처 데이터의 인코딩과 디코딩 간에 역으로 되는, 픽처 인코더 및 디코더 장치.

청구항 10

제9항에 있어서,

상기 가역적 오버랩 연산자는 각각이 유닛 디터미넨트인 복수 개의 컴포넌트 변환들로 구조화되는, 픽처 인코더 및 디코더 장치.

청구항 11

제10항에 있어서,

상기 프로세서는,

가역적 하다마드 네트워크를 적용하고,

가역적 블록 회전을 적용하고,

가역적 스케일링 연산자를 적용하고,

다른 가역적 블록 회전을 적용하고,

가역적 인버스 하다마드 네트워크를 적용함으로써

상기 가역적 오버랩 연산자를 실현하는, 픽처 인코더 및 디코더 장치.

청구항 12

제11항에 있어서,

상기 가역적 스케일링 연산자의 적용시에 상기 프로세서는,

제1 값과, 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 비율을 제2 값에 곱한 값을 합하는 제1 리프팅 단계를 수행하는 단계;

상기 제2 값과, 상기 제1 합에 상기 파라미터 값을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;

상기 제1 합과, 상기 제2 합에 상기 파라미터 값의 음수를 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계; 및

상기 제2 합과, 상기 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 상기 비율의 음수를 상기 제3 합에 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계

를 포함하는 4-리프팅 단계 프로시저를 수행하는, 픽처 인코더 및 디코더 장치.

청구항 13

제11항에 있어서,

상기 가역적 스케일링 연산자의 적용시에 상기 프로세서는,

제1 값과 제2 값을 합하는 제1 리프팅 단계를 수행하는 단계;

상기 제1 값과, 상기 제1 합에 $\frac{1}{2}\left(\frac{s}{1+c}-1\right)$, 여기서 $s^2+c^2=1$, 을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;

상기 제1 합과, 상기 제2 합에 $\frac{2s}{c}$ 를 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계;

상기 제2 합과, 상기 제3 합에 $\frac{1}{2}\left(\frac{s}{1+c}+1\right)$ 을 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계; 및

상기 제3 합과, 상기 제4 합의 음수를 합하는 제5 리프팅 단계를 수행하는 단계

를 포함하는 5-리프팅 단계 프로시저를 수행하는, 픽처 인코더 및 디코더 장치.

청구항 14

제13항에 있어서,

상기 리프팅 단계들 중 한 개 이상은 상기 가역적 오버랩 연산자의 다른 단계와 결합되거나 또는 상기 다른 단계에 의해 취소되는, 픽처 인코더 및 디코더 장치.

청구항 15

제13항에 있어서,

상기 프로세서는 2-포인트 스케일링의 캐스케이드를 사용하여 s_1 에서 s_M 까지 M 개의 스케일링 인자들에 의해 M 개의 데이터 경로들을 스케일링하여 상기 가역적 스케일링 연산자를 실현하는, 픽처 인코더 및 디코더 장치.

청구항 16

제10항에 있어서,

상기 프로세서는,

가역적 하다마드 네트워크를 적용하는 것;

가역적 블록 회전을 적용하는 것;

가역적 스케일링 연산자를 적용하는 것; 및

가역적 인버스 하다마드 네트워크를 적용하는 것

에 의해 상기 가역적 오버랩 연산자를 실현하는, 픽처 인코더 및 디코더 장치.

청구항 17

컴퓨터-실행가능한 픽처 프로세싱 프로그램을 기록한 적어도 한 개의 컴퓨터-판독가능한 기록 매체로서, 상기 프로그램이 컴퓨터 프로세서에 의해 실행되는 경우 컴퓨터로 하여금 블록들에 타일링된 픽처 데이터를 프로세싱 하는 방법을 수행하게 하며, 상기 방법은,

블록들 간의 경계들 중 적어도 일부에 걸쳐 가역적 오버랩 연산자를 적용하는 단계 - 상기 가역적 오버랩 연산자는 유닛 디터미넌트임 - ; 및

상기 블록들에 가역적 블록 변환을 적용하는 단계를 포함하고,

상기 가역적 오버랩 연산자와 상기 가역적 블록 변환의 적용은 상기 픽처 데이터의 인코딩과 디코딩 간에 역으로 되는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 18

제17항에 있어서,

상기 가역적 오버랩 연산자는 각각이 유닛 디터미넌트인 복수 개의 컴포넌트들로 구조화되는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 19

제18항에 있어서,

상기 가역적 오버랩 연산자를 적용하는 상기 단계는,

가역적 하다마드 네트워크를 적용하는 단계;

가역적 블록 회전을 적용하는 단계;

가역적 스케일링 연산자를 적용하는 단계;

다른 가역적 블록 회전을 적용하는 단계; 및

가역적 인버스 하다마드 네트워크를 적용하는 단계를 포함하는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 20

제19항에 있어서,
 상기 가역적 스케일링 연산자를 적용하는 상기 단계는,
 제1 값과, 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 비율을 제2 값에 곱한 값을 합하는 제1 리프팅 단계를 수행하는 단계;
 상기 제2 값과, 상기 제1 합에 상기 파라미터 값을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;
 상기 제1 합과, 상기 제2 합에 상기 파라미터 값의 음수를 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계;
 및
 상기 제2 합과, 상기 파라미터 값과 (1 - 상기 파라미터 값의 제곱)의 상기 비율의 음수를 상기 제3 합에 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계를 포함하는 4-리프팅 단계 프로시저를 수행하는 단계를 포함하는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 21

제19항에 있어서,
 상기 가역적 스케일링 연산자를 적용하는 상기 단계는,
 제1 값과 제2 값을 합하는 제1 리프팅 단계를 수행하는 단계;
 상기 제1 값과, 상기 제1 합에 $\frac{1}{2}\left(\frac{s}{1+c}-1\right)$, 여기서 $s^2+c^2=1$, 을 곱한 값을 합하는 제2 리프팅 단계를 수행하는 단계;
 상기 제1 합과, 상기 제2 합에 $\frac{2s}{c}$ 을 곱한 값을 합하는 제3 리프팅 단계를 수행하는 단계;
 상기 제2 합과, 제3 합에 $\frac{1}{2}\left(\frac{s}{1+c}+1\right)$ 을 곱한 값을 합하는 제4 리프팅 단계를 수행하는 단계; 및
 상기 제3 합과, 상기 제4 합에 곱한 값을 합하는 제5 리프팅 단계를 수행하는 단계를 포함하는 5-리프팅 단계 프로시저를 수행하는 단계를 포함하는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 22

제21항에 있어서,
 상기 가역적 스케일링 연산자를 적용하는 상기 단계는 2-포인트 스케일링의 캐스케이드를 사용하여 s_1 에서 s_M 까지 M 개의 스케일링 인자들에 의해 M 개의 데이터 경로들을 스케일링하는 단계를 포함하는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 23

제18항에 있어서,
 상기 가역적 오버랩 연산자를 적용하는 단계는,
 가역적 하다마드 네트워크를 적용하는 단계;
 가역적 블록 회전을 적용하는 단계;

가역적 스케일링 연산자를 적용하는 단계; 및
 가역적 인버스 하다마드 네트워크를 적용하는 단계
 를 포함하는, 적어도 한 개의 컴퓨터-판독가능한 기록 매체.

청구항 24

압축을 위해 이미지 데이터를 코딩하는 방법으로서, 상기 방법은 컴퓨터 프로세서에 의해 수행되며,
 상기 이미지 데이터의 블록들에 적용된 겹침 변환(lapped transform)을 이용하여 상기 이미지 데이터를 프로세싱하는 단계 -

상기 겹침 변환을 이용하는 상기 프로세싱하는 단계는 부분적으로,

제1 신호 경로 내지 제4 신호 경로에 대한 연산들을 수행하는 것에 의해 구현된 4 포인트 오버랩 연산자를 이용하여 블록 경계에 걸쳐있는(straddling) 이미지 데이터의 4 포인트들의 세트를 프로세싱하는 단계를 포함하고,

상기 연산들은,

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 1의 계수를 갖는 제1 리프팅 단계(lifting steps);

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-1/2$ 의 계수를 갖는 제2 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $1/2$ 의 계수를 갖는 제3 리프팅 단계;

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-29/32$ 의 계수를 갖는 제4 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $3/16$ 의 계수를 갖는 제5 리프팅 단계; 및

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제1 신호 경로까지에 대한 $3/32$ 의 계수를 갖는 제6 리프팅 단계

를 적어도 포함함 - ;

상기 겹침 변환을 이용하는 상기 프로세싱 단계에 의해서 생성된 계수들을 비트스트림으로 인코딩하는 단계; 및
 상기 비트스트림을 출력하는 단계
 에 인코더를 사용하는, 방법.

청구항 25

제24항에 있어서,

상기 4 포인트 오버랩 연산자의 상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 회전 연산(rotation operation)을 더 포함하는, 방법.

청구항 26

제24항에 있어서,

상기 4 포인트 오버랩 연산자의 상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 스케일링 연산(scaling operation)을 더 포함하는, 방법.

청구항 27

제24항에 있어서,

상기 4 포인트 오버랩 연산자의 상기 연산들은 적어도,
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계;
 상기 제3 신호 경로에서 상기 제4 신호 경로까지에 대한 $3/8$ 의 계수를 갖는 리프팅 단계; 및
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계를 더 포함하는, 방법.

청구항 28

압축된 비트스트림으로부터 이미지 데이터를 디코딩하는 방법으로서, 상기 방법은 컴퓨터 프로세서에 의해 수행되며,

상기 압축된 비트스트림으로부터 겹침 변환 계수들을 디코딩하는 단계;

이미지 데이터의 블록들을 재구성하기 위해 인버스 블록 변환을 이용하여 상기 겹침 변환 계수들을 프로세싱하는 단계 -

상기 인버스 블록 변환을 이용하는 상기 프로세싱 단계는 부분적으로,

제1 신호 경로 내지 제4 신호 경로에 대한 연산들의 역순으로 인버스들(inverses)을 수행하는 것에 의해 구현된 4 포인트 인버스 오버랩 연산자를 이용하여 블록 경계에 걸쳐있는 변환 계수들의 4 포인트들의 세트를 프로세싱하는 단계를 포함하고,

상기 연산들은,

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 1의 계수를 갖는 제1 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-1/2$ 의 계수를 갖는 제2 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $1/2$ 의 계수를 갖는 제3 리프팅 단계;

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-29/32$ 의 계수를 갖는 제4 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $3/16$ 의 계수를 갖는 제5 리프팅 단계; 및

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제1 신호 경로까지에 대한 $3/32$ 의 계수를 갖는 제6 리프팅 단계

를 적어도 포함함 - ; 및

디스플레이에 표시하기 위해 상기 이미지 데이터로부터 이미지를 생성하는 단계에 디코더를 사용하는, 방법.

청구항 29

제28항에 있어서,

상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 회전 연산을 더 포함하는, 방법.

청구항 30

제28항에 있어서,

상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 스케일링 연산을 더 포함하는, 방법.

청구항 31

제28항에 있어서,
 상기 연산들은 적어도,
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계;
 상기 제3 신호 경로에서 상기 제4 신호 경로까지에 대한 $3/8$ 의 계수를 갖는 리프팅 단계; 및
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계를 더 포함하는, 방법.

청구항 32

압축된 비트스트림으로부터 이미지 데이터를 디코딩하기 위한 디코더 장치로서,
 상기 압축된 비트스트림으로부터 겹침 변환 계수들을 디코딩하고 이미지 데이터의 블록들을 재구성하기 위해 인버스 블록 변환을 이용하여 상기 겹침 변환 계수들을 프로세스하도록 동작하는 디지털 신호 프로세서 -

상기 인버스 블록 변환을 이용하여 프로세싱하는 것은 부분적으로,

제1 신호 경로 내지 제4 신호 경로에 대한 연산들의 역순으로 인버스들을 수행하는 것에 의해 구현된 4 포인트 인버스 오버랩 연산자를 이용하여 블록 경계에 걸쳐있는 상기 변환 계수들의 4 포인트들의 세트를 프로세싱하는 것을 포함하고,

상기 연산들은,

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 1의 계수를 갖는 제1 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-1/2$ 의 계수를 갖는 제2 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $1/2$ 의 계수를 갖는 제3 리프팅 단계;

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제2 신호 경로까지에 대한 $-29/32$ 의 계수를 갖는 제4 리프팅 단계;

상기 제1 신호 경로에서 상기 제4 신호 경로까지, 그리고 상기 제2 신호 경로에서 상기 제3 신호 경로까지에 대한 $3/16$ 의 계수를 갖는 제5 리프팅 단계; 및

상기 제4 신호 경로에서 상기 제1 신호 경로까지, 그리고 상기 제3 신호 경로에서 상기 제1 신호 경로까지에 대한 $3/32$ 의 계수를 갖는 제6 리프팅 단계

를 적어도 포함함 - ; 및

상기 재구성된 이미지 데이터의 블록들로부터 이미지를 표시하기 위한 디스플레이를 포함하는, 디코더 장치.

청구항 33

제32항에 있어서,
 상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 회전 연산을 더 포함하는, 디코더 장치.

청구항 34

제32항에 있어서,
 상기 연산들은 상기 제3 신호 경로와 상기 제4 신호 경로 사이에 스케일링 연산을 더 포함하는, 디코더 장치.

청구항 35

제32항에 있어서,
 상기 연산들은 적어도,
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계;
 상기 제3 신호 경로에서 상기 제4 신호 경로까지에 대한 $3/8$ 의 계수를 갖는 리프팅 단계; 및
 상기 제4 신호 경로에서 상기 제3 신호 경로까지에 대한 $-3/16$ 의 계수를 갖는 리프팅 단계를 더 포함하는, 디코더 장치.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- [0035] 본 발명은, 일반적으로, 겹침 변환(lapped transform)들을 사용하는 디지털 미디어(예를 들어, 비디오와 이미지) 압축에 관한 것이다.
- [0036] <겹침 변환들(Lapped Transforms)>
- [0037] 겹침 변환은 데이터 압축에 사용되는 강력한 신호 처리 기술이다. *예로서*, 1992년, Artech House, MA, Boston, *Signal Processing with Lapped Transforms*, H.S. Malvar를 참조한다. 그러나, 현재까지, 선형 위상을 갖는 효율적인 겹침 변환들은 체계화되지 않았고, 이는 데이터의 무손실(가역적) 압축에 적용되지도 않았다.
- [0038] 아래 더 상세히 논의되는 것처럼, 겹침 변환은 데이터 변환이 뒤에 오는 전필터(및 후필터가 뒤에 오는 역 데이터 변환(inverse data transform)으로서 그것의 역(inverse))로서 체계화될 수 있다는 것이 알려진다. *예로서*, 1987년 6월, Sweden, Stockholm, *Proc. Picture Coding Symposium*에서 "A pre- and post-filtering technique for the reduction of blocking effects", H. S. Malvar; 2003년 6월, *IEEE Trans. on Signal Processing*, 제51권, 6호, "Lapped Transform via Time-Domain Pre- and Post-Filtering", T. D. Tran, J. Liang, 및 C. Tu를 참조한다. 무손실 데이터 변환이 이 공식에 사용되어 양호한 가역성을 얻도록 할 수 있다. 지금까지는, 가역성을 위해 어느 정도 다양성이 제한된 전필터와 후필터들만 선택될 수 있다고 믿어졌다. 이 제한된 집합은 그것의 압축(비율(rate) 대 왜곡(distortion), 또는 R-D) 성능에 있어서 매우 한정된다. 최근의 한 논문(2003년 10월, *IEEE Trans. on Signal Processing*, 제51권, 페이지 2568-2581, "Regularity-constrained pre- and post-filtering for block DCT-based systems", W. Dai 및 T. Tran)에서, 대부분의 엘리먼트들이 가역적이고 양호한 압축 속성들을 갖는 구성이 제시된다.
- [0039] 오디오 압축에서, 가역적 겹침 변환들에 대한 여러 구성들이 도입된다. *예로서*, 2002년 5월, FL, Orlando, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*에서, "IntMDCT - A link between perceptual and lossless audio coding", R. Geiger, J. Herre, J. Koller, 및 K. Brandenburg; 및 2004년 5월, Canada, Montreal, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*에서, "Reversible FFT and MDCT via matrix lifting", J. Li를 참조한다. 그러나, 이러한 구성들은, 그 기저 함수들(basis functions)이 직교(orthogonal)이고 대칭이 아닌(즉, 기저 함수들은 선형 위상이 아닌), 또한 MDCT(modified discrete cosine transform)로서 알려진, MLT(modulated lapped transform)에만 적용가능하다. 이들 변환들은, 디지털 픽처 압축과 같은, 선형 위상(대칭) 함수들이 요구되는 데이터 압축 애플리케이션들에 적용될 수 없다.
- [0040] 픽처(이미지) 압축을 위해, R-D 성능의 견지에서 최고-수행 변환(best-performing transforms)들 중의 하나는 LBT(lapped biorthogonal transform)이다. 1998년 4월, *IEEE Trans. on Signal Processing*, 제46권, 페이지 1043-1053, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and rising artifacts", H. S. Malvar를 참조한다. MLT와는 달리, LBT 기저 함수들은 대칭적이고, 정확하게 직교하지는 않다(LBT에서, 분석 기저 함수들은 합성(synthesis) 기저 함수들에 직교이므로 쌍직교(biorthogonal)라는 용어를 사용함). LBT들은 이미지 압축 애플리케이션들에서 성공적으로 사용되어 왔지만, 정수-가역적(integer-reversible) 구성들이 알려지지 않았으므로, 무손실 이미지 압축에는 아직 사용되지 않고

있다.

[0041] <블록 변환-기반 코딩의 개관>

[0042] 변환 코딩은 많은 오디오, 이미지, 및 비디오 압축 시스템들에서 사용되는 압축 기술이다. 비압축 디지털 이미지와 비디오는 통상적으로 2-차원(2D) 그리드(grid)에 배치된 이미지 또는 비디오 프레임의 위치들에서 픽처 엘리먼트들이나 색들의 샘플들로서 표현되거나 또는 캡처된다. 이것은 이미지 또는 비디오의 공간-영역 표현으로서 일컬어진다. 예를 들어, 이미지에 대한 통상적 형식은 그리드로서 배치된 24-비트 색 픽처 엘리먼트 샘플들의 스트림으로 구성된다. 각 샘플은, 다른 것들 중에서, RGB 또는 YIQ와 같은, 색 공간 내에 있는 그리드의 픽셀 위치에서 색 컴포넌트들을 나타내는 숫자이다. 다양한 이미지와 비디오 시스템들은 샘플링의 다양한 다른 색, 공간, 및 시간 해상도들을 사용할 수 있다. 유사하게, 디지털 오디오는 통상적으로 시간-샘플링된 오디오 신호 스트림으로서 표현된다. 예를 들어, 통상적 오디오 형식은 정기적 시간 간격에서 취해진 오디오 신호의 16-비트 진폭 샘플들의 스트림으로 구성된다.

[0043] 비압축된 디지털 오디오, 이미지, 및 비디오 신호들은 상당한 저장과 전송 용량을 소비할 수 있다. 변환 코딩은 신호의 공간-영역 표현을 주파수-영역(또는 다른 유사 변환 영역) 표현으로 변환하고, 그 다음, 변환-영역 표현 중 일반적으로 적게 인지되는 주파수 컴포넌트들의 해상도를 감소시킴으로써 디지털 오디오, 이미지, 및 비디오의 크기를 감소시킨다. 이로 인해, 일반적으로 공간 영역 내의 이미지나 비디오의 색이나 공간 해상도를 감소시키는 것에 비해 디지털 신호의 성능 저하, 또는 시간 영역 내의 오디오의 성능 저하가 훨씬 적게 인지된다.

[0044] 더 구체적으로, 도 1에 도시된 전형적인 블록 변환-기반 코덱(100)은 비압축 디지털 이미지의 픽셀들을 고정-크기의 2 차원 블록들(X_1, \dots, X_n)로 분리하고, 각 블록은 다른 블록들과 오버랩될 수 있다. 공간-주파수 분석을 행하는 선형 변환(120 내지 121)은 각 블록에 적용되고, 이것은 블록 내의 이격된 샘플들을, 블록 간격에 걸쳐 대응 주파수 대역들 내의 디지털 신호의 세기를 일반적으로 나타내는 주파수(또는 변환) 계수들의 집합으로 변환한다. 압축을 위해, 변환 계수들은 선택적으로 양자화(quantization)되고(130)(즉, 계수 값들 중의 가장 중요하지 않은 비트들을 버리거나, 그렇지 않으면 고해상도 숫자 집합 내의 값들을 저해상도로 매핑하는 등, 해상도가 감소됨), 또한 압축 데이터 스트림으로 엔트로피 또는 가변-길이 코드화된다(130). 디코딩에서, 변환 계수들은 원래의 색/공간 샘플링 이미지/비디오 신호를 거의 재구성하기 위해 역(inverse)으로 변환할 것이다(170 내지 171)(재구성된 블록들 $\hat{X}_1, \dots, \hat{X}_n$).

[0045] 블록 변환(120 내지 121)은 크기 N의 벡터 x에 수학적 연산으로서 정의될 수 있다. 가장 빈번하게, 연산은 선형 곱셈이고, 변환 영역 출력 $y = Mx$ 를 생성하는데, 여기서 M은 변환 행렬이다. 입력 데이터가 제멋대로 긴 경우, 그것은 N 크기의 벡터들로 세그먼트화되고, 각 세그먼트(segment)에 블록 변환이 적용된다. 데이터 압축을 위해, 가역적 블록 변환들이 선택된다. 다시 말해, 행렬 M은 인버터블하다. 다중 차원(즉, 이미지와 비디오에 대해)에서, 블록 변환들은 통상적으로 분리가능한 연산들로서 구현된다. 행렬 곱셈은 데이터의 각 차원에 따라(즉, 행과 열 모두) 개별적으로 적용된다.

[0046] 압축을 위해, 변환 계수들(벡터 y의 컴포넌트)은 선택적으로 양자화될 수 있고(즉, 계수 값들 중의 가장 중요하지 않은 비트들을 버리거나, 그렇지 않으면 고해상도 숫자 집합의 값들을 저해상도로 매핑하는 등, 해상도가 감소됨), 또한 압축 데이터 스트림으로 엔트로피 또는 가변-길이 코드화된다.

[0047] 도 1에 도시된 바와 같이, 디코더(150)에서의 디코딩에서, 이들 연산들의 역(inverse)(비양자화(dequantization)/엔트로피 디코딩(160) 및 역 블록 변환(170 내지 171))이 디코더(150) 측에서 적용된다. 데이터를 재구성하는 동안, 역행렬 M^{-1} (역 변환(170 내지 171))는 변환 영역 데이터에 곱셈기로서 적용된다. 변환 영역 데이터에 적용될 때, 역변환은 원래 시간-영역 또는 공간 영역 디지털 미디어를 거의 재구성한다.

[0048] 다수의 블록 변환-기반 코딩 애플리케이션들에서, 변환은 양자화 인자에 따라 손실과 무손실 압축 모두를 지원하기 위해 가역적인 것이 바람직하다. 예를 들어, 양자화가 없이(일반적으로 1의 양자화 인자로서 표현됨), 가역적 변환을 사용하는 코덱은 디코딩에 입력 데이터를 정확하게 재생성할 수 있다. 그러나, 이들 애플리케이션들에서 가역성의 요구는 코덱이 디자인될 수 있는 변환들의 선택을 제한한다.

[0049] 다른 것들 중에서, MPEG와 윈도우즈 미디어와 같은, 많은 이미지와 비디오 압축 시스템들은 DCT(Discrete Cosine Transform)에 기초한 변환들을 사용한다. DCT는 근접-최적(near-optimal) 데이터 압축의 결과를 가져

오는 유익한 에너지 압축 속성들을 갖는 것으로 알려진다. 이들 압축 시스템들에서, 역 DCT(IDCT)는 개별 이미지 블록들을 재구성하는 압축 시스템의 인코더와 디코더 모두의 재구성 루프들에서 채택된다. DCT는 1974년 1월, *IEEE Transactions on Computers*, C-23, 페이지 90-93, "DCT(Discrete Cosine Transform)", N. Ahmed, T. Natarajan, 및 K. R. Rao에 의해 기재된다. IDCT의 전형적인 구현은 1990년 12월 6일, *IEEE Std. 1180-1990*, "IEEE Standard Specification for the Implementations of 8 x 8 Inverse Discrete Cosine Transform"에 기재된다.

[0050] 스틸 이미지(still image)(또는 비디오 시퀀스에서 인트라 코딩된 프레임)를 압축하는 동안, MPEG-2, MPEG-4, 및 윈도우즈 미디어와 같은 가장 일반적 표준들은 이미지를 정방향 파일들로 분리하여, 각 이미지 파일에 블록 변환을 적용한다. 주어진 분할(일반적으로 블록으로 알려짐)에서 변환 계수들은 블록 내의 미가공(raw) 데이터 컴포넌트들에 의해서만 영향을 받는다. 양자화와 같은 인코더 측의 비가역적 또는 손실 연산들은 디코딩된 이미지에 아티팩트들(artifacts)이 나타나도록 한다. 이들 아티팩트들은 블록들에서 독립적이고, 블록킹 효과로서 알려진 가시적으로 성가신 효과를 생성한다. 유사하게, 오디오 데이터에서, 논-오버랩(non-overlapping) 블록들이 독립적으로 변환 코딩되었을 때, 양자화 오류들은 디코더에서 오디오 신호의 재구성 시에 블록 경계들에서 신호에 불연속성을 생성할 것이다. 오디오에서, 주기적 클리핑 효과가 들릴 것이다.

[0051] 여러 기술들이 블록킹 효과를 극복하기 위해 사용된다 - 이들 중 가장 인기 있는 것들은 블록 에지 경계 사이를 스무스하게 하는 디블록킹 필터(deblocking filter), 및 미가공 입력 데이터와 이웃 블록 에지들로부터의 예측 간의 차이를 인코딩하는 공간 보간(spatial extrapolation)이다. 이들 기술들에 결함들이 없는 것은 아니다. 예를 들어, 디블록킹 필터 접근법은 "열린 루프(open loop)", 즉, 순변환 프로세스(forward transform process)는 디블록킹이 디코더 측의 재구성 전에 수행될 것이라는 사실을 염두에 두지 않는다. 더욱이, 이들 기술들 모두는 계산상 비경제적이다.

[0052] 블록킹 효과를 최소화하기 위해, 블록 간 상관관계가 이용될 수 있다. 블록 간 상관관계를 얻는 한 방법은, 1992년, Norwood MA, Artech House, "Signal Processing with Lapped Transforms", H. Malvar에 기재된 것처럼 겹침 변환을 사용해서이다. 겹침 변환은, 그 입력이, 현재 블록의 데이터 엘리먼트들과 더불어, 이웃 블록들 내의 몇 개의 인접 엘리먼트들을 스패н(span)하는 변환이다. 유사하게, 재구성 측에서, 역변환은 현재 블록 내의 모든 데이터 포인트들 및 이웃 블록들 내의 몇 개의 데이터 포인트들에 영향을 미친다.

[0053] 2 차원 데이터의 경우에, 랩트 2D 변환은, 왼쪽, 오른쪽, 위, 아래 및 왼쪽 위, 오른쪽 위, 왼쪽 아래, 및 오른쪽 아래로 블록들의 선택 엘리먼트들과 함께, 현재 블록의 함수이다. 현재 변환을 계산하기 위해 사용되는 이웃 블록들 내의 데이터 포인트들의 수는 오버랩으로서 일컬어진다.

[0054] <공간 영역 겹침 변환의 개관>

[0055] 겹침 변환은 종래 블록이 변환한 후에 변환 영역 양들을 병합하는 단계로서, 변환 영역에서 구현될 수 있다. 그 외에, 오버랩의 범위 내의 픽셀들에 적용되는 전처리 단계에 의해 공간-영역에서 구현될 수 있다. 이들 2 개의 구현들은 수학적으로 관련이 있고, 그러므로 동등하다.

[0056] 도 2는 종래의 공간-영역 겹침 변환의 일 예를 도시한다. 도시된 예에서, 오버랩은 2 픽셀이고, 도시된 2 개의 인접 블록들로부터의 각 2 개의 픽셀들은 전처리 단계(210)에서 전처리된다. 2 개의 전처리된 출력들은 도 1에서 처럼 코텍(100)에 의한 블록 변환-기반 코딩을 위해 블록들의 각각에 전송된다. 전처리 단계의 역(inverse)은 디코딩 후에 후처리 단계(220)에 적용된다. 전처리와 블록 변환의 신중한 선택으로, 광범위한 겹침 변환들이 실현될 수 있다.

[0057] 겹침 변환의 공간 영역 실현의 핵심 이점은, 현존하는 블록 변환-기반 코텍이 전처리와 후처리 단계로 개선되어, 현존하는 코텍 프레임워크를 사용하여, 겹침 변환의 이득들, 즉, 감소된 블록 효과와 더 나은 압축을 유도할 수 있다는 것이다. 전처리(210)와 후처리는 도 3에 도시된 바와 같이 행렬 곱셈으로서 표현될 수 있다. 종래에, 전처리와 후처리 행렬들은 서로 역행렬들이다, 즉, 전처리 행렬(P_T)과 그 역행렬이나 후처리 행렬(P_I)의 곱은 단위 행렬 I와 같다.

[0058] <정의들>

[0059] 일반적으로, 변환의 길이 N 은 특정 변환 블록 내의 변환 계수들의 수이다.

[0060] 변환의 썬포트(support) K 는 변환 블록의 계수들에 영향을 미치는 입력 데이터 포인트들의 수이다. 유사하게, 그것은 역변환의 프로세스에 의해, 각 변환 계수에 의해 영향받는 출력 데이터 포인트들의 수이다.

[0061] DCT와 같은 전형적 블록 변환들에서, 길이와 씨포트는 동일하다. 그러나, 겹침 변환들(LTs)은 씨포트 K가 길이 N보다 더 큰 중요한 부류의 변환들이다. 표기법 $K \times N$ 은 겹침 변환의 씨포트와 길이를 나타내기 위해 사용된다. ($K < N$ 은 확장적인 변환으로, 따라서 데이터 압축에 사용되지 않음)

[0062] 예(300)로서, 도 3에 도시된 6×4 LT(310)는 6 개 입력과 4 개 출력을 갖는 변환이다. 변환이 인버터블이므로, 입력들 중의 2 개는 인접 변환 블록들과 공유된다. ILT(inverse lapped transform; 320)는 그것의 4 개의 입력들로부터 6 개 출력들을 생성한다. 블록 경계 근처의 출력 데이터 포인트들(이 경우, 블록의 각 끝에서 하나의 포인트)은 2 개의 인접 역변환 블록들의 대응하는 응답들을 합하여 재구성된다.

[0063] <압축 시스템들에 사용되는 겹침 변환들에 대한 제한사항들>

[0064] 수학적 관점에서, 입력과 출력 신호들은 물론 중간 계산 결과들을 실수로 간주할 때, 겹침 변환들은 인버터블한 구조들이다. 무한한 정확성이 달성될 수 있으면, 입력 데이터는 그것의 겹침 변환 계수들로부터 완전히 회복될 수 있다. 그러나, 무한한 정확성은 실제로 가능하지 않다; 데이터의 무손실 압축을 위한, 요구사항은, 정수나 고정된-정확성(fixed-precision) 산술 연산에서 동작하고, 데이터를 완전히 재구성하여 전수로 표현된 변환 계수들이 주어지는 변환을 디자인하는 것이다. 이것은 수학적 가역성(invertibility)보다 더 강한 조건이고, 그런 변환은 본 명세서에서 "무손실" 변환으로서 일컬어진다. 더욱이, 무손실 변환은 또한 데이터 압축(손실 및 무손실 모두)에 효과적이도록 요구된다. 그 효율성은 변환된 데이터의 엔트로피에 의해 측정될 수 있고, 엔트로피가 낮을수록, 컨텍스트-기반 산술 연산 코딩 또는 적응성 연속 길이 코딩(adaptive run-length coding)과 같은, 표준 엔트로피 코딩 기술들에 의해 더 많은 변환된 데이터가 압축될 수 있다.

발명이 이루고자 하는 기술적 과제

[0065] 정수 산술 연산에서 가역적이며, 효율적인 무손실 데이터 압축/압축해제 시스템에 기초하여 사용될 수 있는 효율적인 겹침 변환에 대한 다양한 실현 형태들이 본 명세서에 기재된다.

[0066] 가장 효율적인 무손실 변환 디자인들(즉, 변환된 데이터의 최소 엔트로피를 갖는 것들)은 변환 행렬이 유닛 디터미넌트(즉, 변환 행렬의 디터미넌트는 ± 1 임)이도록 요구한다는 것이 보여질 수 있다. 다음 설명에서, 데이터 라운딩(rounding)과 같은 부차적인 비선형적인 현상들이 있을 수 있다는 것이 인식되지만, 변환은 행렬 곱셈으로서 표현될 수 있다고 가정된다. 그러므로, 디터미넌트를 언급할 때, 잘라버림(truncation) 또는 라운딩 양태들은 고려되지 않는다.

[0067] 효율적인 겹침 변환은 본 명세서에서 "오버랩 연산들"로서 일컬어지는 전필터와 후필터를 사용하여 실현된다. 이 실현은 가역적이고, 매우 R-D 효율적이다. 다른 애플리케이션들 중에서, 이러한 새로운 오버랩 연산자들은, 무손실 이미지 압축을 위해 사용될 수 있는, 가역적 LBT들의 구현을 허용한다. 전필터와 후필터는 가역적 연산들을 사용한다. 또한, 설명된 오버랩 연산자들은 계산적 효율성을 위해 단순화한 것들을 포함한다.

[0068] 평면적 회전 변환들과 유닛 디터미넌트 평면적 스케일링 변환들의 연속으로 전필터링과 후필터링 연산들이 실현된다. 더욱이, 평면적 회전들과 평면적 쉬어들(sheers)은 가역적/무손실 연산들로 구현되어, 결과로서, 가역적 오버랩 연산자를 제공한다.

[0069] 전형적인 애플리케이션은 가역적 오버랩 연산자들에 대해 계산상 효율적 근사치들을 사용하여 실현되는 $8 \times 4 \times 1$ 차원 겹침 변환이다.

[0070] 본 발명의 추가 특징들과 이점들은 동반하는 도면들을 참조하여 진행되는 다음 실시예들의 상세한 설명으로부터 명백해질 것이다.

발명의 구성 및 작용

[0071] 다음 설명은, 겹침 변환에 대해 가역적 오버랩 연산자를 사용하는, 디지털 미디어 압축 시스템 혹은 코덱에 관한 것이다. 설명을 위해, 가역적 오버랩 연산자를 병합하는 압축 시스템의 일 실시예는 이미지나 비디오 압축 시스템이다. 다른 경우, 가역적 오버랩 연산자는 다른 2D 데이터에 대해 압축 시스템이나 코덱으로 병합될 수 있다. 가역적 오버랩 연산자는, 디지털 미디어 압축 시스템이 특정 코딩 형식의 압축된 디지털 미디어 데이터를 인코딩하는 것을 요구하지 않는다.

[0072] 1. 인코더/디코더

[0073] 도 4와 도 5는 가역적 오버랩 연산자를 사용하는 겹침 변환에 기초한 대표적인 2 차원 데이터 인코더(400)와 디

코더(500)에 채택되는 프로세스들의 일반적 다이어그램이다. 다이어그램들은, 2D 데이터 인코더와 디코더를 병합하는 압축 시스템에서 이 가역적 오버랩 연산자의 사용과 애플리케이션의 일반화된 혹은 단순화된 설명을 제공한다. 이 가역적 오버랩 연산자에 기초한 다른 인코더들에서, 이 대표적 인코더와 디코더에 도시되는 것들 외에 추가적인 또는 그 보다 더 적은 프로세스들이 2D 데이터 압축에 사용될 수 있다. 예를 들어, 일부 인코더들/디코더들은 색 변환, 색 형식, 스케일가능한 코딩, 무손실 코딩, 매크로블록(macroblock) 모드 등도 포함할 수 있다. 압축 시스템(인코더와 디코더)은, 무손실에서 손실까지 다양한 양자화 파라미터에 기초하는 양자화에 의존하여, 2D 데이터의 무손실 및/또는 손실 압축을 제공할 수 있다.

[0074] 2D 데이터 인코더(400)는 인코더에 입력으로서 제공되는 2D 데이터(410)의 더욱 압축된 표현(전형적인 입력에 대해)인 압축 비트스트림(420)을 생성한다. 예를 들어, 2D 데이터 입력은 이미지, 비디오 시퀀스의 프레임, 또는 2차원을 갖는 다른 데이터일 수 있다. 2D 데이터 인코더는, 이 대표적 인코더에서 크기가 16 x 16 픽셀인, 매크로블록들로 입력 데이터를 타일링한다(430). 2D 데이터 인코더는 각 매크로블록을 4 x 4 블록들(432)로 추가로 타일링한다. "포워드 오버랩(forward overlap)" 연산자(440)는 블록들 간의 각 변에 적용된 후, 각 4 x 4 블록은 블록 변환(450)을 사용하여 변환된다. 이 블록 변환(450)은, 본 명세서에 참조로서 그 개시 내용이 병합되고, 이것과 함께 동시 출원된, "Improved Reversible Transform For Lossy And Lossless 2-D Data Compression"라는 발명의 명칭의 미국특허출원서로서, Srinivasan에 의해 기재된 가역적, 스케일프리(scale-free) 2D 변환일 수 있다. 다른 경우, DCT 또는 다른 블록 변환들은 본 명세서에 기재된 가역적 오버랩 연산자와 사용될 수 있다. 변환에 후속하여, 각각의 4 x 4 변환 블록의 DC 계수(460)는 유사한 프로세싱 체인(타일링, 포워드 오버랩, 그 다음에, 4 x 4 블록 변환)에 따른다. 결과적 DC 변환 계수들과 AC 변환 계수들은 양자화되고(470), 엔트로피 코딩되고(480), 및 패킷화된다(490).

[0075] 디코더는 그 반대 프로세스를 수행한다. 디코드 측에서, 변환 계수 비트들이 그들의 각각의 패킷들로부터 추출되고(510), 계수들은 그들 자체가 디코드되고(520) 비양자화된다(530). DC 계수들(540)은 역변환을 적용하여 재생성되고, DC 계수들의 평면은 DC 블록 예지들에 걸쳐 적용되는 적절한 스무딩 연산자(smoothing operator)를 사용하여 "인버스 오버랩(inverse overlap)"된다. 후속적으로, 전체 데이터는 비트스트림으로부터 디코드된 DC 계수들과 AC 계수들(542)에 4 x 4 역변환(550)을 적용하여 재생성된다. 최종적으로, 결과적 이미지 평면들에서 블록 예지들은 인버스 오버랩 필터링된다(560). 이것은 재구성된 2D 데이터 출력을 생성한다.

[0076] 2. 오버랩 연산자들을 사용하여 실현된 겹침 변환

[0077] 더 일반적으로, 인코더(400)(도 4)의 오버랩 연산자(440)와 블록 변환(450)은, 도 6에 도시된 바와 같이, 전 필터링 연산(610)으로 분해되고 그 후에 블록 데이터 변환(620)이 올 수 있는 겹침 변환들(600)의 큰 클래스의 일 예이다. 도 6은 그런 분해되는 겹침 변환들의 일반화된 예를 도시한다. 이러한 경우에서, 도 3에 도시된 6 x 4 겹침 변환(310)은 전필터 연산(610)과 블록 변환(620) 단계들로 분해된다. 전필터링 연산(610)과 블록 변환(620)은 데이터 포인트들에 걸쳐 균일하게 배치된다. 설명된 6 x 4 겹침 변환(600) 예에서, 각 전필터는 인접 블록들에 걸쳐있는 데이터 포인트들의 길이 2 변환이다. 디코더 측에서, 블록 경계들을 가로질러 역 블록 변환(630) 후에 후필터(640)가 적용된다. 유사하게, 일반적 K x N의 경우, 전필터는 블록 경계에 인접한 각 블록의 (K - N)/2 데이터 포인트들에 적용된다.

[0078] 가역성을 위해, 전필터(610)와 후필터(640)는 서로 인버터블하다. 그러나, 무손실 겹침 변환을 실현하기 위해, 이 조건은 충분하지가 않다. 이것은 또한, 블록(코어(core)) 변환(620)이 무손실 방식으로 실현되도록 하는 것에 추가하여, 전필터와 후필터들(610, 640)이 무손실 변환들이 되도록 제한한다. DCT는, 다른 것들 중에서, 래더(ladder), 래티스(lattice), 또는 리프팅(lifting)-기반 방법들을 사용하여, 무손실 방식으로 실현될 수 있다. 예로서, 1992년, *IEEE J. Selected Areas Communications*, 제10권, 1호, "New networks for perfect inversion and perfect reconstruction", A. A. M. L. Bruekens와 A.W.M. van den Enden; 및 1998년, *J. Fourier Anal. Appl.*, 제4권, 페이지 247-269, "Factoring wavelet transform into lifting steps", I. Daubechies와 W. Sweldens를 참조한다. 가역적 스케일프리 2 차원 변환은 또한, 본 명세서에 참조로서 병합되고 이것과 함께 동시 출원된, "Improved Reversible Transform For Lossy And Lossless 2D Data Compression"라는 발명의 명칭의 미국특허출원서, Srinivasan에 의해 기재된다. 또한, 1 차원에서 DCT으로의 리프팅-기반 가역적 근사법이 알려졌다. 예로서, 2001년 12월, *IEEE Trans. Signal Processing*, 제49권, 페이지 3032-3044, "Fast multiplierless approximations of the DCT with the lifting scheme", J. Liang과 T.D. Tran을 참조한다.

[0079] 효율적 가역성은, 양 단계들, 즉, 전/후 필터와 블록 변환이 유닛 디터미넌트이도록 더 요구한다.

- [0080] 3. 가역적 오버랩 연산자
- [0081] 인코더(400)/디코더(500)(도 4와 도 5)가 기초로 하는 무손실 겹침 변환(600)의 전필터(610)(도 6)로서 사용하기 위한 효율적인 가역적 오버랩 연산자는, 도 7에 도시된 구조(700)로 분해되는, 선형 위상 전필터로서 실현될 수 있다. 이 전필터의 역(inverse)(즉, 후필터(640))은 동일 구조를 갖지만 다른 계수들을 갖는다.
- [0082] 이 선형 위상 필터 구조(700)는, 그것의 입력과 출력에 교차 하다마드 네트워크(cross-over Hadamard network;710)를 포함하는, 다중 직교 컴포넌트들을 갖는다. 도시된 하다마드 네트워크(710)의 내부 화살표들은 이 도면에서 음수(negation)를 나타낸다. 구조(700)는 직교 행렬들 U1, U2, V1, 및 V2를 더 포함한다. 이들 컴포넌트들은 래티스/리프팅 기반 방법들을 사용하여 무손실 방식으로 구현될 수 있다.
- [0083] 추가로, 구조(700)는 s_1 에서 s_M 까지 0이 아닌 스케일 인자들을 가진다. 유닛 디터미넌트 제한사항은 $\prod_i s_i = \pm 1$ 를 내포한다. 모든 스케일 인자들이 ± 1 일 때, 전/후 필터들은 컴포넌트 행렬들 U1, U2, V1, 및 V2가 무손실 래티스/리프팅 단계들로서 구현되는 무손실 변환으로서 실현될 수 있다. 그러나, 스케일 인자들이 모두가 ± 1 이 아닐 때, 무손실 실현은 아래 더 완전히 논의되는 것처럼 해결되는 문제점으로 남는다.
- [0084] 이 선형 위상 전필터 구조(700)로, 무손실 전/후 필터 쌍을 실현하는 문제는 다음 3 단계들로 감소된다:
- [0085] 1. 직교 행렬들 U1, U2, V1, 및 V2에 대해, 필터 F를 다음 형태로 분해한다:

수학식 1

$$F = \begin{pmatrix} I & J \\ J/2 & -I/2 \end{pmatrix} \begin{pmatrix} U_1 & 0 \\ 0 & V_1 \end{pmatrix} \begin{pmatrix} s_1 & & \\ & s_2 & \\ & & \ddots \\ & & & s_M \end{pmatrix} \begin{pmatrix} U_2 & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} I/2 & J \\ J/2 & -I \end{pmatrix}$$

$$J = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & & \ddots \\ 1 & 0 & 0 \end{pmatrix}$$

- [0086]
- [0087] 여기서, I는 단위 행렬이다.
- [0088] 2. U1, U2, V1, 및 V2에 대해 무손실 실현 형태들을 유도한다.
- [0089] 3. 스케일링 행렬에 대해 무손실 실현 형태를 유도한다.
- [0090] 단계 1에서, 2 포인트 하다마드 변환들을 정의하는, 오른쪽의 처음과 최종 행렬들은 일부 항들에 1/2의 인자들을 포함하여 이들 단계들을 유닛 디터미넌트인 것으로 만든다. 나머지는, F의 선형 차원들의 반의 각각인, 2개의 블록들을 갖는 블록 대각선 형태로 재배치된다. 각 블록의 SVD(singular value decomposition)는 직교 행렬들 U1, U2, V1, 및 V2, 및 스케일들을 제공한다.
- [0091] 컴포넌트 행렬들의 무손실 실현 형태들은, 1992년, IEEE 제이. 선택된 지역 통신(IEEE J. Selected Areas Communications), 제10권, 1호, "완전 인버전과 완전 재구성을 위한 새로운 네트워크들(New networks for perfect inversion and perfect reconstruction)", 에이.에이.엠.엘. 브루켄스(A.A.M.L. Bruekens)와 에이.더블유.엠. 반 덴 엔덴(A.W.M. van den Enden)에 의해 기재된 것들과 같은, 표준 리프팅-기반 기술들을 사용하여 단계 2에서 유도될 수 있다.
- [0092] 단계 3의 스케일링 행렬의 무손실 실현 형태는 다음과 같이 해결된다. 단순성을 위해, 우리가 (a) 무손실이고 (b) 제1 컴포넌트에 대해 $s(0 < s < 1)$ 에 의해 및 제2 컴포넌트에 대해 $1/s$ 에 의해 스케일링하는 것을 실현하는 (다른 경우들은 한 개 또는 두 개 모두의 출력 신호들의 부호를 전환하여 얻어질 수 있음) 특정한 2 입력 2 출력 컴포넌트를 갖는다고 가정하자. 부연하여, 우리는 다음 식에 의해 주어진 입력-출력 관계를 가진다:

수학식 2

$$y = \begin{pmatrix} s & 0 \\ 0 & 1/s \end{pmatrix} x$$

[0093]

[0094] 수학식 2에서 변환 행렬의 디터미넌트(determinant)는 $s/s = 1$ 이다. 이 행렬은 도 8과 도 9에 도시된 바와 같이 4 개의 리프팅 단계 프로시저(800) 또는 5 개의 리프팅 단계 프로시저(900)에서 실현될 수 있다. 우리는 보통 $y = (a \cdot x + r) \gg b$ 의 형태로 모든 리프팅 단계들을 근사하고, 여기서 x 는 입력이고, y 는 출력이고, a , b , 및 r 은 정수들이고, r 은 분리없는 정수 구현을 얻기 위한 라운딩 오류 제어를 위해 사용된다. 수학식 2에 의해 정의된 변환은, 스케일링 변환으로서 축약되는, 본 명세서에서 유닛 디터미넌트 스케일링 변환으로서 일컬어진다.

[0095] 흥미롭게도, 스케일링 변환은, 아래 정의된 바와 같이, 쉬어 연산과 가깝게 관련이 있다:

수학식 3

$$y = \begin{pmatrix} a & b \\ b & a \end{pmatrix} x$$

[0096]

[0097] 제한사항 $a^2 - b^2 = 1 (a > 0, b \geq 0)$ 하에, 쉬어 연산은 유닛 디터미넌트이며, 3 개의 리프팅 단계들로 실현될 수 있다:

수학식 4

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} = \begin{pmatrix} 1 & \sqrt{\frac{a-1}{a+1}} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} 1 & \sqrt{\frac{a-1}{a+1}} \\ 0 & 1 \end{pmatrix}$$

[0098]

[0099] 그러므로,

수학식 5

$$\begin{aligned} \begin{bmatrix} a+b & 0 \\ 0 & a-b \end{bmatrix} &= \begin{bmatrix} 1 & 1/2 \\ -1 & 1/2 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & -1/2 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \left(\sqrt{\frac{a-1}{a+1}} + 1 \right) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2b & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \left(\sqrt{\frac{a-1}{a+1}} - 1 \right) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

[0100]

[0101] 여기서, 쉬어 행렬을 사이에 둔 행렬들의 스케일링 인자들 1/2과 2는 쉬어 리프팅 단계들로 분산되고, 제1 행렬의 최종 리프팅 단계는 제1 쉬어 리프팅 단계와 결합되고, 한편 최종 행렬의 제1 리프팅 단계는 제1 쉬어 리프팅 단계와 결합된다. 도 9에 도시된 스케일링 변환의 프로시저(900)로서 5 개의 단계 실현은 수학식 5에 기초한다. 구조에 있어서의 단순화는, 수학식 1의 3 개의 그룹들, 즉, 하다마드 네트워크, 직교 행렬들, 및 스케일링 연산들(이것은 하다마드와 쉬어 연산들로 분해될 수 있음) 사이에서, 가능한 경우에, 역 연산들을 취소함으로써 가능해 질 수 있다.

[0102] 더 구체적으로, 프로시저(800)로서 무손실 스케일링의 4 개의 리프팅 단계 실현의 효과적 변환 행렬은 $T = \begin{pmatrix} c^2 & 0 \\ 0 & c^2 \end{pmatrix}$ 이고, 여기서, $c^2 = 1 - s^2$ 이다. 한편, 프로시저(900)의 5 개의 리프팅 단계 실현의 효과적 변환 행렬은 다음과 같다.

$$T = \begin{pmatrix} \frac{1+s}{c} & 0 \\ 0 & \frac{1-s}{c} \end{pmatrix}, \text{ 여기서, } c^2 = 1 - s^2.$$

[0103]

[0104] 도 8에 도시된 스케일링 프로시저(800)가 도 9의 것보다 한 개 적은 리프팅 단계를 갖지만, 전 프로시저가 4 개인데 반해 나중 프로시저(900)는 단지 3 개의 논-트리비얼(non-trivial) 리프팅 단계들만을 가진다. 위의 문단에 기재된 이유에서, 도 9의 처음이나 최종 트리비얼 리프팅 단계는 특정 조건 하(예를 들어, U1, U2, 및 V1 이 단위 행렬들일 때)에서 이전 또는 후속 변환 단계들(예를 들어, 도 7의 어느 한쪽에서 하다마드 네트워크(710)와 병합될 수 있다.

[0105] 스케일링 프로시저는 더 큰 행렬들로 쉽게 확장될 수 있다. 이것은, s_1 에서 s_M 까지 M 개의 다를 수 있는 스케일

인자들이 스케일링 변환들의 캐스케이드(1000)로서 M 데이터 경로들에 적용되는, 도 10에 도시된다. 가역적 방식으로 이것을 성취하기 위해, M - 1 가역적 스케일링 변환들이 일반적으로 필요하다.

[0106] 한 가지 유용한 특별 경우는 s_1 에서 s_M 까지 M 개의 스케일 인자들이 $(s, 1/s)$ 형태의 M/2 그룹들로 그룹화될 수 있을 때이다. 이 경우, 단지 M/2 가역적 스케일링 변환들이 필요하다. 한 가지 예는 $s_1 = s_2 = \dots = s_{M/2} = s$ 및 $s_{M/2+1} = s_{M/2+2} = \dots = s_M = 1/s$. 선호되는 그룹화 방법은 중앙축을 가로질러 대칭을 이루도록 하는 것이고, 부연하면, 각 그룹은 계수들 s_i 와 s_{M+1-i} 를 스케일링한다. M이 홀수이면, 그룹화되지 않은 한 개의 스케일 인자는, 축을 따라 데이터 경로에 대응하여, 1이다.

[0107] 전/후 필터들이 신호 외부로 확장할 필요가 있는 신호 경계들에서, 한 가지 해결책은 신호를 대칭적으로 확장하여, 전/후 필터들을 적용하는 것이다. 이것은 스케일링 때문에 일반적으로 무손실 연산이 아니다. 다른 해결책은 경계들에서 전/후 필터링을 스킵(skip)하는 것이다. R-D 성능 및 인지적 품질(예를 들어, 손실 이미지/비디오 압축을 위해 사용되면)로서 2 개의 해결책들 간에 주목할 만한 차이가 없다.

[0108] 이제 도 11의 참조에서, 그 다음, 바람직한 R-D 효율적(즉, 유닛 디터미넌트) 특성을 갖는 가역적 오버랩 연산자는 가역적 유닛 디터미넌트 하다마드 네트워크들(710), 가역적 직교 회전들(1110)(컴포넌트 행렬들 U1, U2, V1, 및 V2에 대해), 및 가역적 유닛 디터미넌트 스케일링(1120)(즉, 리프팅 단계 프로시저들(800, 900) 또는 캐스케이드(1100)를 사용하여)을 포함하는 선형 위상 전필터 구조(700)(도 7)로서 실현된다. 후필터는, 역순으로 역 리프팅 단계들이지만, 전필터와 유사하고, 동일 구성을 사용하여 빌드(build)된다. 이것은, 블록의 데이터 값들 M의 수가 일반적으로 임의의 자연수인, 도 7에 도시된다. 설명이 짝수 값인 M에 대한 것이지만, 홀수 값들도 중앙 데이터 값의 "1 포인트 하다마드" 변환이 그 자체임을 주목하여 또한 가능하다. 이 프로시저는 고차원 데이터로 일반화될 수 있다.

[0109] 요약하면, 가역적 오버랩 연산자의 연산이 도 12에 도시된다. 제1 단계(1210)에서, 입력 2-차원 디지털 미디어 데이터는 블록들로 타일링된다(도 4의 인코더(400)에 대해 또한 도시된 바와 같이). 가역적 오버랩 연산자는 단계(1220)에서 인접 타일들을 가로질러 하다마드 네트워크(710)를 적용한다. 그 다음, 연산자는 단계(1230)에서 합들과 차들에 가역적 회전들을 적용하고, 그 다음 단계(1240)에서 가역적 스케일링 연산자가 뒤따른다. 이것 뒤에 다른 가역적 블록 회전(단계(1250))과 가역적 인버스 하다마드 네트워크(단계(1260))가 뒤따른다.

[0110] 이제 도 13의 참조에서, 가역적 블록 회전들과 스케일링 연산자들의 행렬 표현들은 예를 들어 수학식 1에 기재된 산술 연산을 사용하는 원하는 랩트 연산자에 따른다. 도 13은, 가역적 블록 변환(이 경우, 4 포인트 하다마드 변환)이 선행되는, 도 7과 도 11에 도시된 구조(700)를 갖는 후필터의 일 예를 도시한다. 후필터의 트랜스퍼 함수(transfer function)는 다음과 같다:

수학식 6

$$T = \begin{pmatrix} 0.9885 & 0.1553 & -0.1553 & 0.2183 \\ -0.1499 & 0.9885 & 0.2183 & 0.1499 \\ 0.1502 & 0.2167 & 0.9884 & -0.1502 \\ 0.2167 & -0.1556 & 0.1556 & 0.9884 \end{pmatrix}$$

[0111] 하다마드의 지역필터 컴포넌트는 도 14의 그래프에 도시된 임펄스 응답을 생성한다.

[0112]

4. 컴퓨팅 환경

[0113] 위에 기재된 가역적 오버랩 연산자를 사용하는 결합 변환에 기초한 코덱은, 다른 예들 중에도, 컴퓨터, 이미지와 비디오 녹음, 전송, 및 수신 장치, 휴대용 비디오 플레이어, 비디오 회의, 및 기타 등등을 포함하는, 디지털 미디어 신호 처리가 수행되는 다양한 디바이스들 중의 임의의 것에서 수행될 수 있다. 디지털 미디어 코딩 기술들은, 도 15에 도시된 바와 같이, 컴퓨터 또는 다른 컴퓨팅 환경 내에서 실행하는 디지털 미디어 프로세싱 소프트웨어 및 하드웨어 회로에서 구현될 수 있다.

[0114] 도 15는 기재된 실시예들이 구현될 수 있는 적절한 컴퓨팅 환경(1500)의 일반 예를 도시한다. 본 발명이 다양한 일반 목적 또는 특수 목적 컴퓨팅 환경에 구현될 수 있으므로, 컴퓨팅 환경(1500)은 본 발명의 사용이나 기능의 범위에 대해 임의의 제한을 제안하려고 의도되지는 않는다.

[0115] 도 15의 참조에서, 컴퓨팅 환경(1500)은 적어도 한 개의 프로세싱 유닛(1510)과 메모리(1520)를 포함한다. 도

15에서, 이 가장 기본 구성(1530)은 점선 안에 포함된다. 프로세싱 유닛(1510)은 컴퓨터-실행가능 명령들을 실행하고, 실제 또는 가상 프로세서일 수 있다. 멀티프로세싱 시스템에서, 멀티프로세싱 유닛들은 컴퓨터-실행가능한 명령들을 실행하여 프로세싱 능력을 증가시킨다. 메모리(1520)는 휘발성 메모리(예를 들어, 레지스터, 캐쉬(cache), RAM), 비휘발성 메모리(예를 들어, ROM, EEPROM, 플래쉬 메모리 등), 또는 그 둘의 어떤 조합일 수 있다. 메모리(1520)는 기재된 인코더/디코더와 변환들을 구현하는 소프트웨어(1580)를 저장한다.

[0117] 컴퓨팅 환경은 추가적 특징들을 가질 것이다. 예를 들어, 컴퓨팅 환경(1500)은 저장 장치(1540), 한 개 이상의 입력 디바이스들(1550), 한 개 이상의 출력 디바이스들(1560), 및 한 개 이상의 통신 접속들(1570)을 포함한다. 버스, 제어기, 또는 네트워크와 같은 상호접속 메카니즘(도시 안됨)은 네트워크 환경(1500)의 컴포넌트들을 상호접속한다. 통상적으로, 운영 시스템 소프트웨어(도시 안됨)는 컴퓨팅 환경(1500)에서 실행하는 다른 소프트웨어를 위한 운영 환경을 제공하고, 컴퓨팅 환경(1500)의 컴포넌트들의 활동들을 조정한다.

[0118] 저장 장치(1540)는 분리형이나 비분리형일 수 있고, 자기 디스크, 자기 테이프나 카세트, CD-ROM, CD-RW, DVD, 또는 정보를 저장하기 위해 사용될 수 있고 컴퓨팅 환경(1500) 내에 액세스될 수 있는 임의의 다른 매체를 포함한다. 저장 장치(1540)는 가역적 오버랩 연산자를 사용한 겹침 변환에 기초하여 코덱을 구현하는 소프트웨어(1580)를 위한 명령들을 저장한다.

[0119] 입력 디바이스(들)(1550)는 키보드, 마우스, 펜, 또는 트랙볼, 음성 입력 장치, 스캐닝 디바이스, 또는 컴퓨팅 환경(1500)에 입력을 제공하는 다른 디바이스와 같은 터치(touch) 입력 디바이스일 것이다. 오디오에서, 입력 디바이스(들)(1550)는 아날로그나 디지털 형태로 오디오 입력을 받는 사운드 카드나 유사한 디바이스, 또는 컴퓨팅 환경에 오디오 샘플들을 제공하는 CD-ROM 판독기일 수 있다. 출력 디바이스(들)(1560)는 디스플레이, 프린터, 스피커, CD-기록기, 또는 컴퓨팅 환경(1500)으로부터 출력을 제공하는 다른 디바이스일 수 있다.

[0120] 통신 접속(들)(1570)은 다른 컴퓨팅 개체로 통신 매체를 통한 통신을 가능하게 한다. 통신 매체는 컴퓨터-실행가능한 명령들, 압축된 오디오나 비디오 정보, 또는 변조 데이터 신호의 다른 데이터와 같은 정보를 전달한다. 변조 데이터 신호는 신호에 정보를 인코딩하는 방식에서 그것의 한 개 이상의 특성들이 설정되거나 변경되는 신호이다. 예를 들어, 하지만 이에 제한되지는 않고, 통신 매체는 전기, 광학, RF, 적외선, 음향, 또는 다른 반송파로 구현되는 유선이나 무선 기술들을 포함한다.

[0121] 본 명세서에서 디지털 미디어 프로세싱 기술들은 컴퓨터-판독가능 매체의 일반 컨텍스트로 기재될 수 있다. 컴퓨터-판독가능 매체는 컴퓨팅 환경 내에서 액세스될 수 있는 임의의 이용가능한 매체이다. 예를 들어, 하지만 이에 제한되지는 않고, 컴퓨팅 환경(1500)에서, 컴퓨터-판독가능 매체는 메모리(1520), 저장 장치(1540), 통신 매체, 및 위의 것들 중의 임의의 것들의 조합들을 포함한다.

[0122] 본 명세서에서 디지털 미디어 프로세싱 기술들은 컴퓨팅 환경에서 타겟(target) 실제나 가상 프로세서에서 실행되는, 프로그램 모듈들에 포함된 것들과 같은, 컴퓨터-실행가능 명령들의 일반 컨텍스트로 기재될 수 있다. 일반적으로, 프로그램 모듈들은 특정 작업들을 수행하거나 특정 추상 데이터 유형들을 구현하는 루틴, 프로그램, 라이브러리, 객체, 클래스, 컴포넌트, 데이터 구조 등을 포함한다. 프로그램 모듈들의 기능은 다양한 실시예들에서 바람직한 바와 같이 프로그램 모듈들 간에 조합되거나 분리될 수 있다. 프로그램 모듈들을 위한 컴퓨터-실행가능한 명령들은 로컬이나 분산 컴퓨팅 환경 내에서 실행될 수 있다.

[0123] 표현을 목적으로, 상세한 설명은 컴퓨팅 환경에서 컴퓨터 동작들을 설명하기 위해 "결정한다(determine)", "생성한다(generate)", "조정한다(adjust)", 및 "적용한다(apply)"와 같은 용어들을 사용한다. 이들 용어들은 컴퓨터에 의해 수행되는 동작들에 대한 고수준 추상화(high-level abstraction)이고, 사람에게 의해 수행되는 동작들과 혼동되어서는 안 된다. 이들 용어들에 대응하는 실제 컴퓨터 동작들은 구현에 따라 달라진다.

[0124] 5. 가역적 오버랩 연산자의 변형과 확장

[0125] 위에 기재된 가역적 오버랩 연산자의 다양한 수정과 확장이 만들어질 수 있다. 위에 제공된 설명들이 1 차원 데이터에 대한 것이지만, 동일 프로시저가 복수 개의 데이터 차원들에 개별적으로, 또는 비개별적으로 적용될 수 있다.

[0126] 위에 기재된 가역적 오버랩 연산자 구현의 직교 회전들은 그것들의 근사치들에 의해, 또는 직교가 아닐 수 있는 다른 변환들에 의해 대체될 수 있다.

[0127] 더욱이, 위의 설명의 제1 포커스(focus)는 입력 데이터의 무손실 복구에 있지만, 동일 변환이 손실 데이터 압축을 위해 또한 사용될 수 있다. 이 경우, 양자화 프로세스에서, 또는 전필터나 후필터의 제한된 정확성/근사 구

현 때문에, 또는 다른 부정확성들 또는 복수 개의 인자들의 조합 때문에, 손실이 발생할 수 있다.

- [0128] 본 명세서에서 기재된 가역적 오버랩 연산자는 데이터-압축의 외부 영역들에 적용될 수 있다. 가역적 오버랩 연산자를 사용한 겹침 변환은 그 자체가 확장적일 수 있다.
- [0129] 가역적 오버랩 연산자는 멀티레이트 필터 뱅크(multirate filter banks), 웨이브릿(wavelets), 2 블록 너비 이상 서포트 스페닝(support spanning)을 갖는($K > 2N$) 겹침 변환을 구현하기 위해 적절히 수정된 형태로 적용될 수 있다.
- [0130] 가역적 오버랩 연산자는, 오버랩 필터의 범위와 형상이 데이터의 공간적 범위에 걸쳐 달라질 수 있는, 공간적으로 변하는 방식으로 적용될 수 있다.
- [0131] 본 발명의 원칙들이 적용될 수 있는 다수의 가능한 실시예들의 관점에서, 우리는 다음 청구범위와 그것의 동등물들의 범위와 취지 내에 올 수 있는 모든 그런 실시예들을 본 발명으로서 청구한다.

발명의 효과

- [0132] 정수 산술 연산에서 가역적이고, 효율적이고 무손실 데이터 압축/압축해제 시스템의 기초로서 사용될 수 있는 효율적 겹침 변환에 대한 다양한 실현 형태들이 본 명세서에 기재된다. 가장 효율적 무손실 변환 디자인들(즉, 변환된 데이터의 최소 엔트로피를 갖는 것들)은 변환 행렬이 유닛 디터미넌트(즉, 변환 행렬의 디터미넌트는 ± 1 임)이도록 요구된다. 효율적인 겹침 변환은 본 명세서에서 "오버랩 연산들"로서 일컬어지는 전필터와 후필터를 사용하여 실현된다. 이 실현은 가역적이고, 매우 R-D 효율적이다.

도면의 간단한 설명

- [0001] 도 1은 종래 기술에서 종래 블록 변환-기반 코덱(block transform-based codec)의 블록도이다.
- [0002] 도 2는 도 1 또한 종래 기술의 블록 변환-기반 코덱과 조합하여 전처리(pre-processing) 및 후처리(post-processing) 연산들로서 구현된 공간-영역 겹침 변환(spatial-domain lapped transform)의 블록도이다.
- [0003] 도 3은 1 차원 데이터에 겹침 변환과 역(inverse) 겹침 변환 쌍을 도시하는 블록도이다.
- [0004] 도 4는 가역적 오버랩 연산기(reversible overlap operator)를 사용하는 겹침 변환에 기초하는 인코더의 흐름도이다.
- [0005] 도 5는 겹침 변환에 기초한 디코더의 흐름도이다.
- [0006] 도 6은 블록 변환과 관련하여 전필터링(prefiltering)과 후필터링(postfiltering) 연산들(또는 가역적 오버랩 연산자)을 사용하여 1 차원 데이터에 겹침 변환과 역 겹침 변환 쌍을 도시하는 블록도이다.
- [0007] 도 7은 도 6의 겹침 변환의 가역적 오버랩 연산자로서 사용하는 선형 위상(linear phase) 전필터(또는 후필터)의 구조를 도시하는 신호 흐름 그래프이다.
- [0008] 도 8은 가역적 오버랩 연산자에서 사용하는 4 개의 리프팅(lifting) 단계들로서 무손실 스케일링(lossless scaling)의 신호 흐름 그래프이다.
- [0009] 도 9는 가역적 오버랩 연산자에서 사용하는 5 개의 리프팅 단계들로서 무손실 스케일링의 신호 흐름 그래프이다.
- [0010] 도 10은 무손실 유닛 디터미넌트(unit determinant) 스케일링을 실현하기 위해 더 큰 차원 행렬에 적용되는 2-포인트 스케일링의 캐스케이드(cascade)의 신호 흐름 그래프이다.
- [0011] 도 11은 도 7에 도시된 구조를 가지며 도 10의 무손실 유닛 디터미넌트 스케일링을 사용하는 가역적 오버랩 연산자(또는 전필터/후필터)의 신호 흐름도이다.
- [0012] 도 12는 도 11의 가역적 오버랩 연산자의 동작의 흐름도이다.
- [0013] 도 13은 도 11의 가역적 오버랩 연산자를 사용하여 가역적 겹침 변환 구현의 일 예를 도시하는 신호 흐름도이다.
- [0014] 도 14는 도 13의 겹침 변환의 예의 DC 계수의 임펄스 응답 그래프이다.
- [0015] 도 15는 도 4와 도 5의 향상된 공간-영역 겹침 변환을 갖는 블록 변환-기반 코덱을 구현하는 적절한 컴퓨팅 환

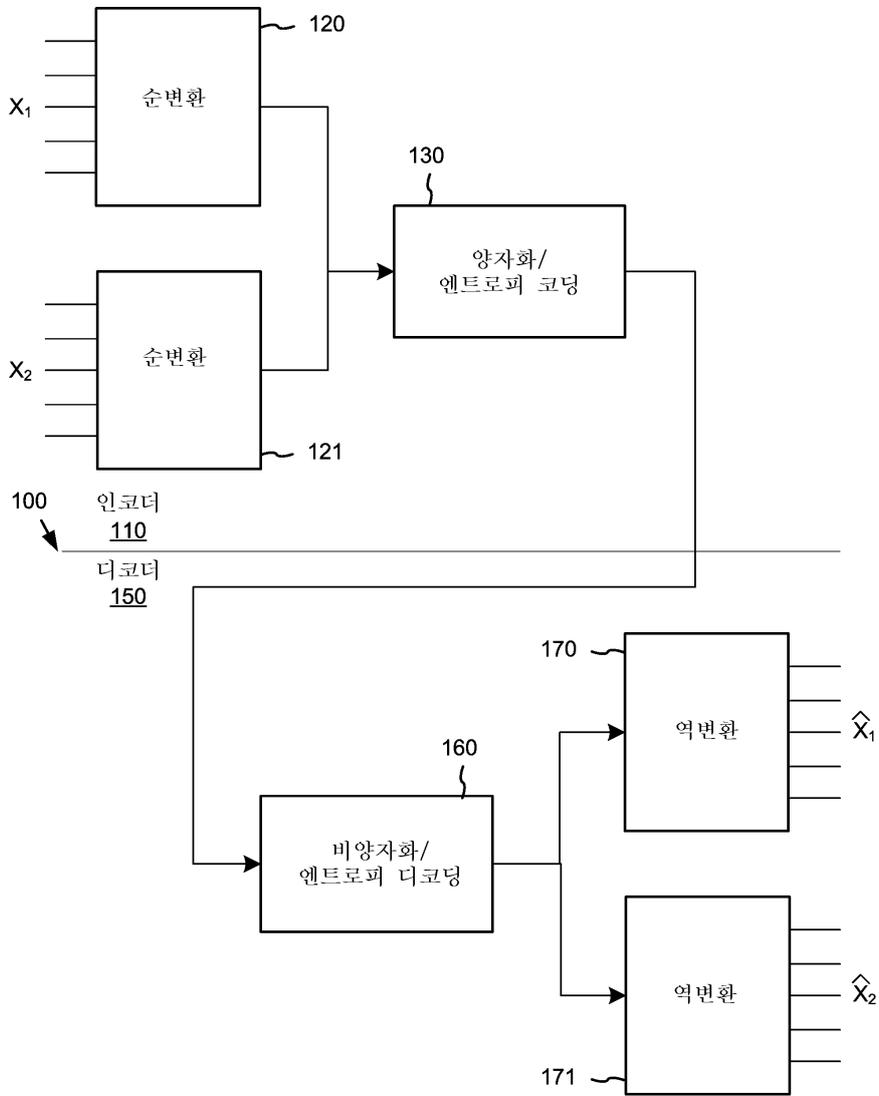
경의 블록도이다.

- [0016] <도면의 주요 부분에 대한 부호의 설명>
- [0017] 410 : 2D 입력 데이터
- [0018] 430 : 타일링
- [0019] 440 : 포워드 오버랩
- [0020] 450 : 블록 변환
- [0021] 460 : DC 계수들
- [0022] 462 : AC 계수들
- [0023] 470 : 양자화
- [0024] 480 : 엔트로피 코딩
- [0025] 490 : 패킷화
- [0026] 420 : 압축된 비트스트림
- [0027] 510 : 추출
- [0028] 520 : 디코딩
- [0029] 530 : 비양자화
- [0030] 542 : AC 계수들
- [0031] 540 : DC 계수들
- [0032] 550 : 역 블록 변환
- [0033] 560 : 인버스 오버랩(inverse overlap)
- [0034] 590 : 2D 데이터

도면

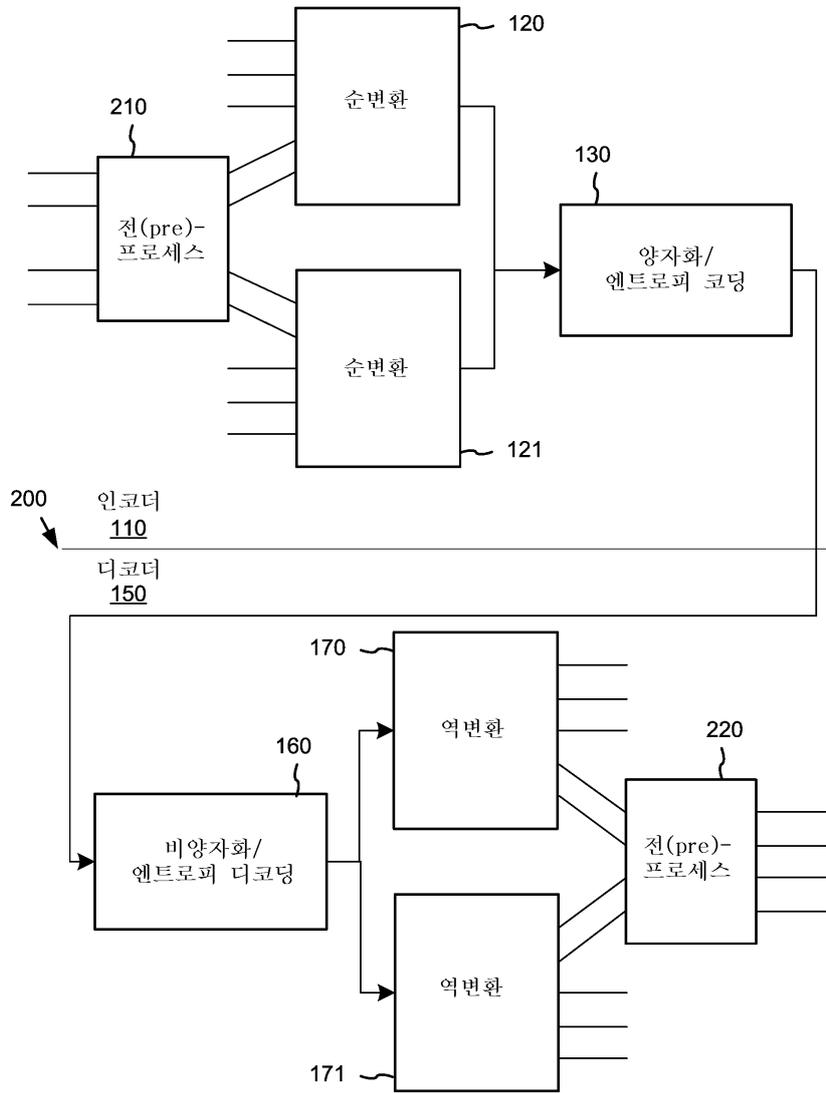
도면1

(종래 기술)

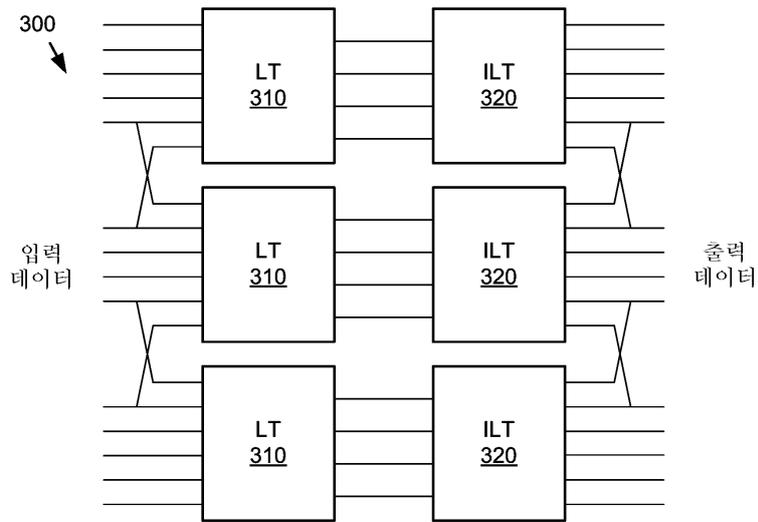


도면2

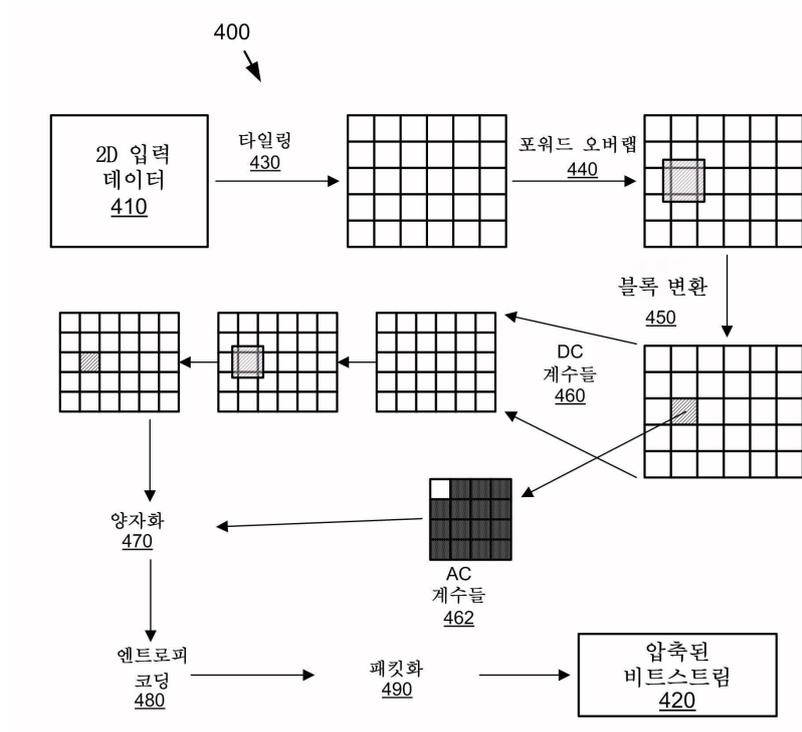
(종래 기술)



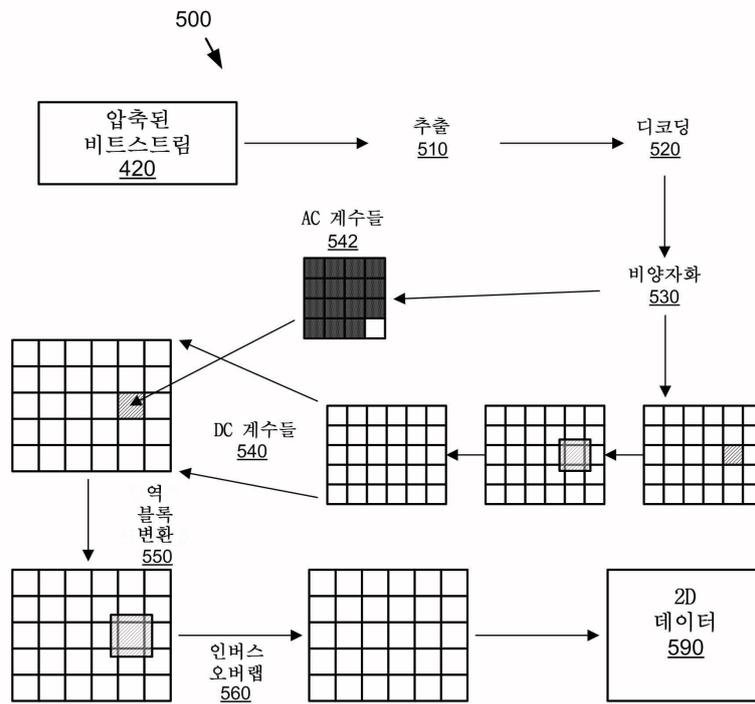
도면3



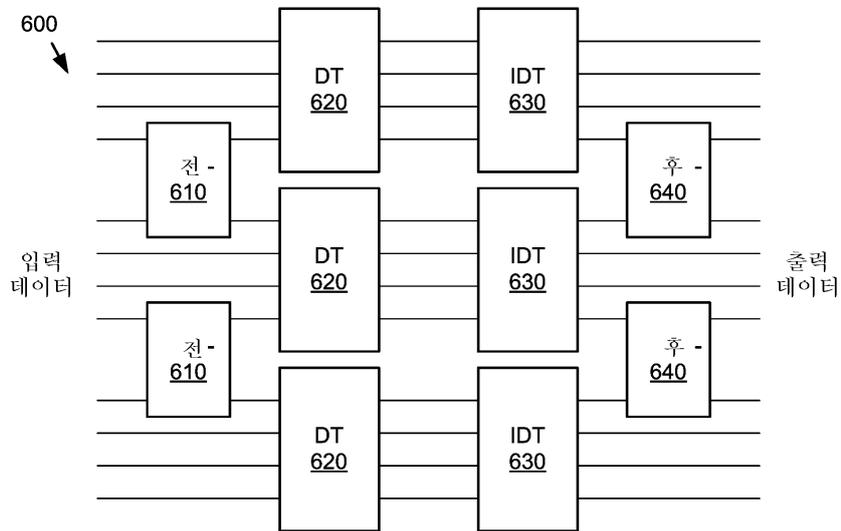
도면4



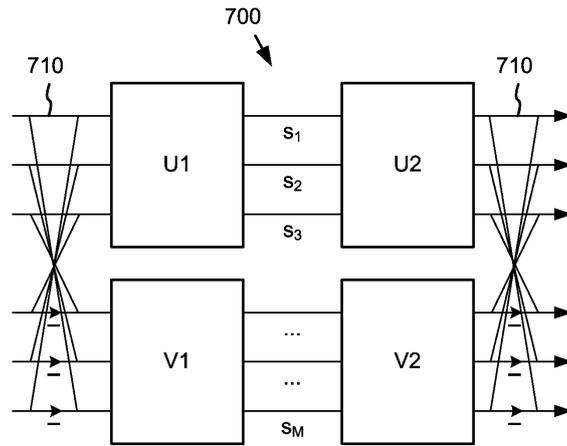
도면5



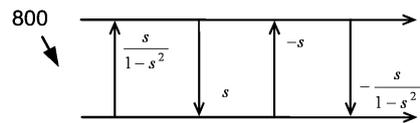
도면6



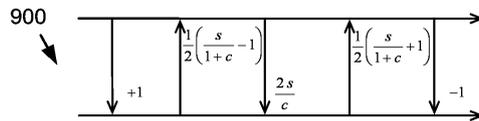
도면7



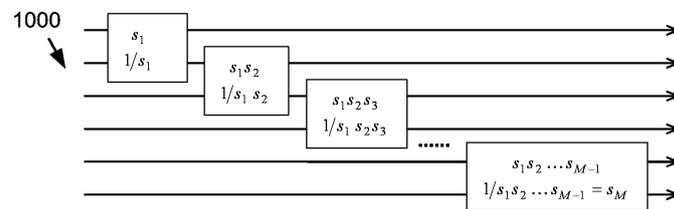
도면8



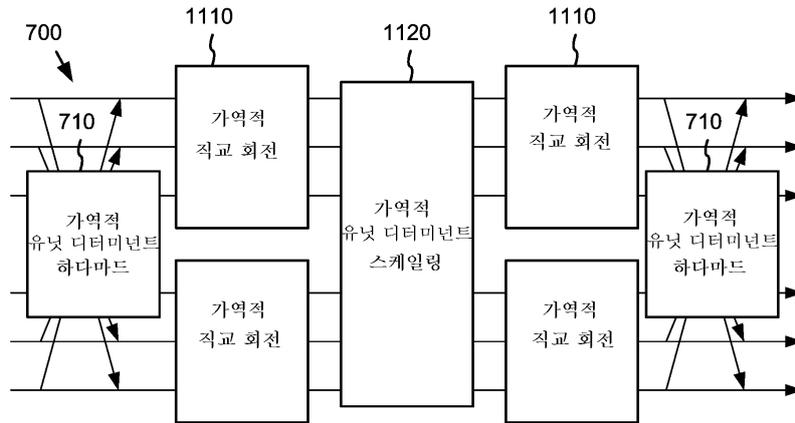
도면9



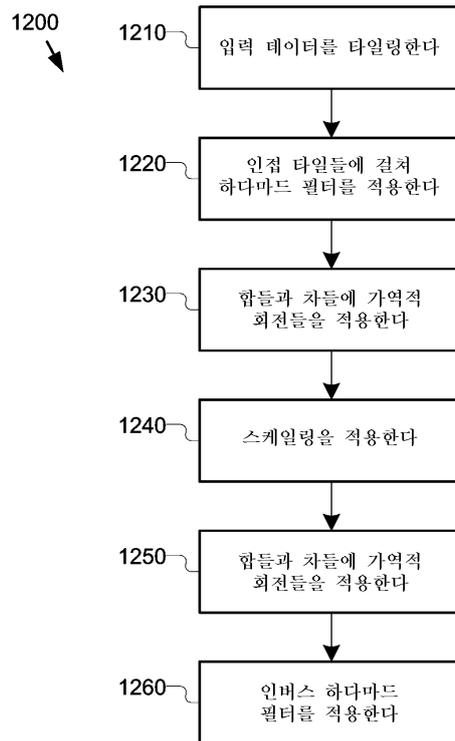
도면10



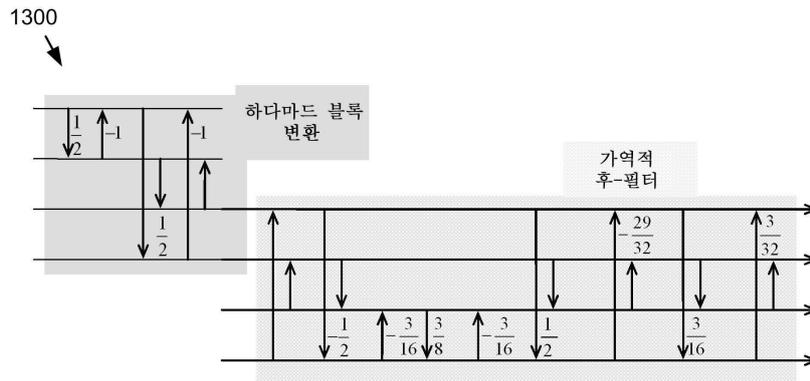
도면11



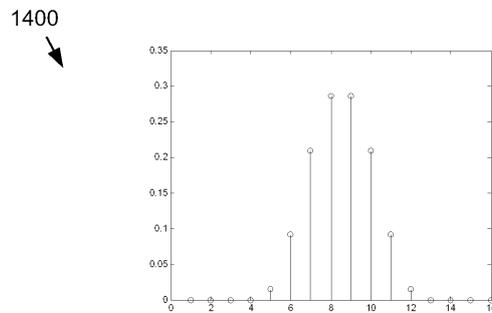
도면12



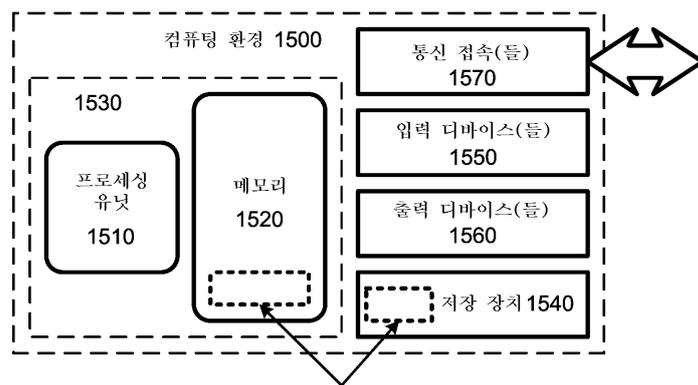
도면13



도면14



도면15



가역적 오버랩 연산자를 구현하는 소프트웨어