



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0090331
(43) 공개일자 2016년07월29일

(51) 국제특허분류(Int. Cl.)
H04N 19/625 (2014.01) G06T 1/20 (2006.01)
G06T 9/00 (2006.01) H04N 19/159 (2014.01)
(52) CPC특허분류
H04N 19/625 (2015.01)
G06T 1/20 (2013.01)
(21) 출원번호 10-2016-7016587
(22) 출원일자(국제) 2014년11월20일
심사청구일자 2016년06월21일
(85) 번역문제출일자 2016년06월21일
(86) 국제출원번호 PCT/US2014/066662
(87) 국제공개번호 WO 2015/077476
국제공개일자 2015년05월28일
(30) 우선권주장
14/087,568 2013년11월22일 미국(US)

(71) 출원인
구글 인코포레이티드
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 (우:94043)
(72) 발명자
한, 정녕
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 구글 인코포레이티드 (우:94043)
수, 야오우
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 구글 인코포레이티드 (우:94043)
무커지, 데바르가
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이
1600 구글 인코포레이티드 (우:94043)
(74) 대리인
특허법인 남앤드남

전체 청구항 수 : 총 20 항

(54) 발명의 명칭 하이브리드 변환 코딩 방식을 위한 구현 설계

(57) 요약

방법 및 시스템은 비디오 코덱을 사용하여 비디오 데이터 블록을 식별하고, 버터플라이 ADST(asymmetric discrete sine transform)의 변환 커널을 파이프라인의 비디오 데이터 블록에 적용할 수 있다.

대표도 - 도1

100



(52) CPC특허분류

G06T 9/007 (2013.01)

H04N 19/159 (2015.01)

명세서

청구범위

청구항 1

실행될 때, 프로세서로 하여금, 프로세스를 수행하게 하기 위한 명령들을 저장하는 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 명령들은,

비디오 코덱을 사용하여 비디오 데이터 블록을 식별하기 위한 명령들; 및

버터플라이 비대칭 이산 사인 변환(ADST; asymmetric discrete sine transform)의 변환 커널을 파이프라인의 비디오 데이터 블록에 적용하기 위한 명령들

을 포함하는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 2

제 1 항에 있어서,

상기 변환 커널은 비대칭 기반을 가지는 버터플라이 ADST 변환 및 이산 코사인 변환(DCT; discrete cosine transform)의 조합을 포함하는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 커널은 $\sin \frac{(2n-1)(2k-1)\pi}{4N}$ 의 형태이고, 여기서 N은 블록 디멘션(block dimension)이고, n 및 k는 각각 시간 인덱스 및 주파수 인덱스를 나타내고, n과 k는 양자 모두 1 내지 N의 범위인,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 4

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 버터플라이 ADST는 정수 포맷으로 수행되는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 5

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 프로세서는 단일 명령 메모리 및 제어 프로세서와 통신하는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 6

제 1 항 내지 제 5 항 중 어느 한 항에 있어서,

상기 명령들은, 예측 방향에 의존하여 1-차원 변환들의 세트를 선택하기 위한 명령들을 더 포함하는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 7

제 6 항에 있어서,

상기 명령들은, 상기 1-차원 변환들의 세트를 사용하여 2-차원 변환을 형성하기 위한 명령들을 더 포함하는,
비-일시적 컴퓨터-판독가능 저장 매체.

청구항 8

제 3 항에 있어서,

상기 버터플라이 ADST는 상기 비디오 데이터 블록으로부터 생성된 잔차 픽셀 블록(residual pixel block)에 적용되는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 9

제 1 항 내지 제 8 항 중 어느 한 항에 있어서,

상기 변환 커널은 단일 명령 다중 데이터 동작들을 통한 병렬 컴퓨팅을 제공하는,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 10

장치로서,

비디오 데이터 블록을 식별하도록 구성된 비디오 코덱; 및

하이브리드 변환 인코더

를 포함하고,

상기 하이브리드 변환 인코더는,

비대칭 기반을 가지는 버터플라이 비대칭 이산 사인 변환(ADST) 및 이산 코사인 변환(DCT)의 조합을 사용하여
상기 비디오 데이터 블록에 인트라-프레임 코딩 방식을 제공하고, 그리고

상기 버터플라이 ADST를 사용하여, 상기 인트라-프레임 코딩의 버터플라이 구조 구현(butterfly structured implementation)을 산출하도록

구성되는,

장치.

청구항 11

제 10 항에 있어서,

상기 커널은 $\sin \frac{(2n-1)(2k-1)\pi}{4N}$ 의 형태이고, 여기서 N은 블록 디멘션이고, n 및 k는 각각 1 내지 N 범위의 시간 인덱스 및 주파수 인덱스를 나타내는,

장치.

청구항 12

제 10 항 또는 제 11 항에 있어서,

상기 버터플라이 구조 구현은 정수 포맷으로 수행되는,

장치.

청구항 13

제 10 항 내지 제 12 항 중 어느 한 항에 있어서,
상기 프로세서는 단일 명령 메모리 및 제어 프로세서와 통신하는,
장치.

청구항 14

제 10 항 내지 제 13 항 중 어느 한 항에 있어서,
예측 방향에 의존하여 1-차원 변환들의 세트를 선택하도록 구성된 선택기를 더 포함하는,
장치.

청구항 15

제 14 항에 있어서,
상기 선택기는 상기 세트를 사용하여 2-차원 변환을 형성하도록 추가로 구성되는,
장치.

청구항 16

제 10 항 내지 제 15 항 중 어느 한 항에 있어서,
상기 버터플라이 ADST는 상기 비디오 데이터 블록으로부터 생성된 잔차 픽셀 블록에 적용되는,
장치.

청구항 17

제 10 항 내지 제 17 항 중 어느 한 항에 있어서,
상기 버터플라이 구조 구현은 단일 명령 다중 데이터 동작들을 통한 병렬 컴퓨팅을 제공하는,
장치.

청구항 18

프로세서에 의해 실행될 때, 프로세스를 수행하게 하기 위한 명령들을 저장하는 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 명령들은,

비대칭 기반을 가지는 오리지널 비대칭 이산 사인 변환(ADST)에 근사한 커널을 갖는 버터플라이 ADST 및 이산 코사인 변환(DCT)의 조합을 사용하여 비디오 데이터 블록에 인트라-프레임 코딩 방식을 제공하기 위한 명령들;

상기 버터플라이 ADST를 사용하여 상기 인트라-프레임 코딩의 버터플라이 구조 구현을 산출하기 위한 명령들을 포함하고,

상기 커널은 $\sin \frac{(2n-1)(2k-1)\pi}{4N}$ 의 형태이고, 여기서 N은 블록 디멘션이고, n 및 k는 각각 시간 인덱스 및 주파수 인덱스를 나타내고, n 및 k는 양자 모두 1 내지 N 범위인,

비-일시적 컴퓨터-판독가능 저장 매체.

청구항 19

제 18 항에 있어서,
상기 명령들은 예측 방향에 의존하여 1-차원 변환들의 세트를 선택하기 위한 명령들을 더 포함하는,
비-일시적 컴퓨터-판독가능 저장 매체.

청구항 20

제 18 항 또는 제 19 항에 있어서,

상기 명령들은 상기 1-차원 변환들의 세트를 사용하여 2-차원 변환을 형성하기 위한 명령들을 더 포함하는,
비-일시적 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2013년 11월 22일자로 출원된 미국 정식 특허 출원 제 14/087,568호의 연속출원이고 이를 우선권으로 주장하며, 상기 정식 특허 출원의 개시내용은 전체가 인용에 의해 본원에 포함된다.

배경 기술

[0002] 본 개시내용은 이미지 및 비디오 압축 코딩에 관한 것이다. 이미지 또는 비디오 압축 코딩에서, 인트라-프레임 코딩(intra-frame coding)은 단일 프레임 내의 비디오 또는 이미지들을 압축하고, 비디오 시퀀스의 임의의 다른 프레임에 대해서는 압축하지 않는 압축 기술을 지칭한다.

[0003] 변환 코딩(transform coding)은 비디오 및 이미지 압축에 있어 중심 컴포넌트이다. 변환 커널(transform kernel)의 최적화는 압축 이득들을 위한 신호 상관을 활용하기 위해 사용될 수 있다. 그러나, 하드웨어 설계 측면에서, 변환 모듈은 코텍 계산 복잡도에 상당부 기여할 수 있다. 이에 따라, 본 기술의 부족한 부분을 해결하고 코딩 효율을 개선하기 위한 시스템들, 방법들 및 장치들에 대한 필요성이 존재한다.

발명의 내용

[0004] 일반적 양상에서, 비-일시적 컴퓨터-판독가능 저장 매체는, 실행될 때, 프로세서로 하여금 프로세스를 수행하게 하기 위한 명령들을 저장하며, 명령들은, 비디오 코텍을 사용하여 비디오 데이터 블록을 식별하기 위한 명령들 및 버터플라이 ADST(butterfly asymmetric discrete sine transform)의 변환 커널을 파이프라인의 비디오 데이터 블록에 적용하기 위한 명령들을 포함한다.

[0005] 구현들은 하기의 특징들 중 하나 이상의 것을 포함할 수 있다. 예를 들어, 변환 커널은 비대칭 기반을 가지는 버터플라이 ADST 변환 및 DCT(discrete cosine transform)의 조합을 포함할 수 있다. 커널은

$$\sin \frac{(2n-1)(2k-1)\pi}{4N}$$

의 형태일 수 있으며, 여기서 N은 블록 디멘션이며, 여기서 n 및 k는, 각각, 시간 인덱스 및 주파수 인덱스를 나타내며, 여기서 n 및 k 양자 모두 1 내지 N 범위이다. 버터플라이 ADST는 정수 포맷으로 수행될 수 있다. 프로세서는 단일 명령 메모리 및 제어 프로세서와 통신할 수 있다. 예를 들어, 명령들은 예측 방향에 의존하여 1-차원 변환들의 세트를 선택하기 위한 명령들을 더 포함할 수 있다. 예를 들어, 명령들은 1-차원 변환들의 세트를 사용하여 2-차원 변환을 형성하기 위한 명령들을 더 포함할 수 있다. 버터플라이 ADST는 비디오 데이터 블록으로부터 생성된 잔차(residual) 픽셀 블록에 적용될 수 있다. 변환 커널은 단일 명령 다중 데이터 동작들을 통한 병렬 컴퓨팅을 제공할 수 있다.

[0006] 다른 일반적 양상에서, 장치는, 비디오 데이터 블록을 식별하도록 구성된 비디오 코텍, 및 비대칭 기반을 가지는 버터플라이 ADST(asymmetric discrete sine transform) 변환 및 DCT(discrete cosine transform)의 조합을 사용하여 비디오 데이터 블록에 인트라-프레임 코딩 방식을 제공하며, 버터플라이 ADST를 사용하여, 인트라-프레임 코딩의 버터플라이 구조 구현(butterfly structured implementation)을 산출하도록 구성된 하이브리드 변환 인코더를 포함한다.

[0007] 구현들은 하기 특징들 중 하나 이상의 것을 포함할 수 있다. 예를 들어, 커널은 $\sin \frac{(2n-1)(2k-1)\pi}{4N}$ 의 형태일 수 있으며, 여기서 N은 블록 디멘션이며, 여기서 n 및 k는, 각각, 1 내지 N 범위의 시간 인덱스 및 주파수 인덱스를 나타낸다. 버터플라이 구조 구현은 정수 포맷으로 수행될 수 있다. 프로세서는 단일 명령 메모리 및 제어 프로세서와 통신할 수 있다. 장치는, 예측 방향에 의존하여 1-차원 변환들의 세트를 선택하도록 구성된 선택기를 더 포함할 수 있다. 선택기는 추가로, 상기 세트를 사용하여 2-차원 변환을 형성하도록 구성될 수 있다.

다. 버터플라이 ADST는 비디오 데이터 블록으로부터 생성된 잔차 픽셀 블록에 적용될 수 있다. 버터플라이 구조 구현은 단일 명령 다중 데이터 동작들을 통한 병렬 컴퓨팅을 제공할 수 있다.

[0008] [0008] 또 다른 일반적 양상에서, 비-일시적 컴퓨터-관독가능 저장 매체는, 실행될 때, 프로세서로 하여금, 프로세서를 수행하게 하기 위한 명령들을 저장하며, 명령들은, 비대칭 기반을 가지는 오리지널 ADST(asymmetric discrete sine transform)에 근사한 커널을 갖는 버터플라이 ADST 변환 및 DCT(discrete cosine transform)의 조합을 사용하여 비디오 데이터 블록에 인트라-프레임 코딩 방식을 제공하기 위한 명령들 및 버터플라이 ADST를 사용하여, 인트라-프레임 코딩의 버터플라이 구조 구현을 산출하기 위한 명령들을 포함한다. 커널은
$$\sin \frac{(2n-1)(2k-1)\pi}{4N}$$
의 형태이며, 여기서 N은 블록 디멘션이고, 여기서 n 및 k는, 각각, 시간 인덱스 및 주파수 인덱스를 나타내며, 여기서 n 및 k 양자 모두 1 내지 N 범위이다.

[0009] [0009] 구현들은 하기의 특징들 중 하나 이상의 것을 포함할 수 있다. 예를 들어, 명령들은 예측 방향에 의존하여 1-차원 변환들의 세트를 선택하기 위한 명령들을 더 포함할 수 있다. 예를 들어, 명령들은 1-차원 변환들의 세트를 사용하여 2-차원 변환을 형성하기 위한 명령들을 더 포함할 수 있다.

[0010] [0010] 본 양상의 다른 구현들은, 컴퓨터 저장 디바이스들 상에 인코딩된 방법들의 동작들을 수행하도록 구성되는 대응하는 시스템들, 장치, 방법 및 컴퓨터 프로그램들을 포함한다.

[0011] [0011] 본원에 설명된 시스템들 및 방법들의 장점들은, 예를 들어, 효율적인 하드웨어 활용 및 코드 스피드-업(code speed-up)을 포함한다.

[0012] [0012] 하나 이상의 구현들의 상세사항들은 하기 설명 및 첨부되는 도면들에서 제시된다. 다른 특징들은 설명 및 도면들로부터 그리고 청구항들로부터 명백할 것이다.

도면의 간단한 설명

[0013] [0013] 도 1은 하이브리드 변환 코딩에 사용될 수 있는 예시적 시스템을 예시하는 블록도이다.

[0014] [0014] 도 2a는 하이브리드 변환 인코딩에 사용될 수 있는 예시적 이미징 디바이스를 예시하는 블록도이다.

[0015] [0015] 도 2b는 SIMD 동작들의 예를 예시하는 블록도이다.

[0016] [0016] 도 3a는 파이프라인에서 하이브리드 변환 코딩을 위해 사용될 수 있는 예시적 프로세스를 예시하는 흐름도이다.

[0017] [0017] 도 3b는 하이브리드 변환 코딩을 위해 사용될 수 있는 예시적 프로세스를 예시하는 흐름도이다.

[0018] [0018] 도 3c는 하이브리드 변환 코딩을 위해 사용될 수 있는 하드웨어 동작 흐름의 예를 예시하는 도면이다.

[0019] [0019] 도 4는 CPU 사이클들과 관련하여 본원에서 설명되는 시스템들 및 방법들의 계산 복잡도(computational complexity)를 예시하는 블록도이다.

[0020] [0020] 도 5는 본원에서 설명되는 시스템들에 사용될 수 있는 예시적 디코딩 프로세스를 예시하는 흐름도이다.

[0021] [0021] 도 6은, 본 개시내용의 양상들에 따라, 본원에서 설명되는 하나 이상의 시스템들 및 방법들을 구현하기 위해 사용될 수 있는 예시적 또는 대표적 컴퓨팅 디바이스들 및 연관된 엘리먼트들을 예시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0014] [0022] 본 개시내용의 양상들에 따라, 이미지 또는 비디오 압축 성능을 개선하기 위한 방법 및 시스템이 본원에서 제시된다. 개시된 방법 및 시스템은 (단일 프레임 내에서) 인트라-프레임 코딩 및 비디오 코덱에서 사용되는 예측을 구현하기 위한 방법을 제공한다.

[0015] [0023] 블록 코더(block coder)를 채용하기 위한 하나의 동기는 비디오 프레임을 별개의 영역들로 분할하는 것이며, 별개의 영역들 각각은, 국부적으로 그 자신의 정상 신호(stationary signal) 통계들을 가질 수 있다. 최대 압축 효율을 달성하기 위해, 인트라-프레임 코딩은, 블록 바운더리(boundary)들에서 이용가능한, 이전에 인코딩된 이웃 픽셀들로부터 각각의 블록에 대한 공간적 예측을 통해, 국부적 이방성(local anisotropy)(이를테면, 프레임 내에서 공간적 패턴들의 발생)을 활용한다.

[0016] [0024] 하이브리드 변환 코딩 방식은, 바운더리 정보의 품질 및 이용가능성에 의존하여, DST(discrete sine

transform) 및 비대칭 DST에서 선택하는 것을 허용한다. 하이브리드 변환 코딩 방식은 동일한 PSNR(peak signal-to-noise ratio)에서 비트 절감들을 달성할 수 있다. 전체 인트라 코딩 방식은, 예측 방향 및 바운더리 정보에 의존하여, 비대칭 DST(ADST)로 불리는 DST의 이러한 변형물과, 통상적 DCT 사이에서 전환될 수 있다. 이러한 적응식 예측-변환 접근방식(또한 하이브리드 변환 코딩 방식이라 불림)은, DCT-기반 인트라-프레임 예측-변환 코딩을 능가할 수 있다. 또한, 하드웨어 설계 측면에서, 변환 모듈은 코텍 계산 복잡도에 상당부 기여할 수 있다.

[0017] [0025] 고속 푸리에 변환은 N 넘버들의 시퀀스에 대한 이산 푸리에 변환을 계산하기 위한 효율적 알고리즘이다. 고속 푸리에 변환 알고리즘들의 맥락에서, 버터플라이는, 더 작은 이산 푸리에 변환들(DFT들)의 결과들을 더 큰 DFT로, 또는 더 큰 이산 푸리에 변환들(DFT들)의 결과들을 더 작은 DFT로 결합하는, 예를 들어, 더 큰 DFT를 서브변환들(subtransforms)로 분해하는 계산의 부분이다.

[0018] [0026] 본원에 설명되는 방법 및 시스템은, 비대칭 이산 정현 변환 그리고 하이브리드 변환 코딩에 대해, 인트라-프레임 변환을 예측하기 위해 버터플라이 비대칭 이산 정현 변환을 산출하는 것을 개선시킨다. 단일 명령 다중 데이터(SIMD; single instruction multiple data) 동작을 통한 병렬 컴퓨팅을 허용하는 개선된 버터플라이 구조 구현이 설명된다.

[0019] [0027] 8×8 또는 이를 초과하는 변환 디멘션에 대해, DCT에 가깝게 근사하는 직교 행렬 및 작은 정수들인 엘리먼트들을 갖는 직교 행렬을 찾는 것은 어려울 수 있다. 더 큰 블록 크기 변환들은 정상 신호에 대한 더 높은 변환 코딩 이득들을 제공하며, 다양한 비디오 코덱들에서 압축 효율에 기여하는 것으로 실험적으로 증명되었다. 고속 ADST의 설계 및 이에 따른, 임의의 블록 크기들의 하이브리드 변환 코딩 방식에서는 문제점들이 발생한다.

$$\sin \frac{n(2k-1)\pi}{2N+1}$$

오리지널 ADST 커널은 $\sin \frac{n(2k-1)\pi}{2N+1}$ 로 유도되었으며, 여기서 N은 블록 디멘션이며, n 및 k는, 각각, 양자 모두 범위가 1 내지 N인 시간 인덱스 및 주파수 인덱스를 나타낸다. 다른 한편으로, DCT 커널은 $\cos \frac{(2n-1)(k-1)\pi}{2N}$ 형태이다. 정현과 변환들(sinusoidal transforms)의 버터플라이 구조 구현들은, 커널 인수(kernel argument)의 분모(denominator), 즉 ADST에 대해서는 $(2N+1)$ 그리고 DCT에 대해서는 2N이 합성수인 경우(그리고 작은 정수들의 곱으로 분해될 수 있는 경우) 존재한다. 이런 이유로, 다수의 블록-기반 비디오(및 이미지) 코덱들은, DCT 변환의 효율적 계산을 위해, 블록 크기의 2 거듭제곱(예를 들어, N=4, 8, 16 등)이 되게 설계된다. 그러나, 이는 오리지널 ADST가 고속 구현을 할 수 없게 만든다. 예를 들어, N = 8일 때, $(2N + 1)$ 은 17이 되며, 이는 버터플라이 구조의 가능성을 배제하는 소수(prime number)이다.

[0020] [0028] 본원에 설명된 방법들 및 시스템들은 DCT와 오리지널 ADST 간의 이러한 내재적 충돌을 해결하는데, 이는 $\sin \frac{(2n-1)(2k-1)\pi}{4N}$ 이들이 커널인 ADST의 새로운 변형을 포함하기 때문이다. 커널 인수의 분모 4N은, 2N이 2 거듭제곱이라면, 4N이라는 점에서 DCT의 것과 일치한다. 따라서, 이는 버터플라이 구조로 구현될 수 있으며, 또한 본원에서 btf-ADST로 지칭될 수 있다.

[0021] [0029] btf-ADST는 오리지널 ADST와 유사한 비대칭 특성을 기반으로 한다. 시스템은 하이브리드 변환 코딩 방식에서 오리지널 ADST를 대체하기 위해 이 btf-ADST를 사용할 수 있다. 전체 방식은 예측 방향에 의존하여 btf-ADST 및 DCT 중에서 적당한 1-차원(1-D) 변환을 선택하여 2-차원(2-D) 변환을 형성한다. 예를 들어, 시스템은 행렬의 각각의 컬럼 벡터에 수직 방향의 변환을 적용할 수 있다.

[0022] [0030] 도 1은, 본 개시내용의 양상들에 따른, 하이브리드 변환 코딩을 위한 예시적 시스템(100)을 예시하는 블록도이다. 도 1의 예에서, 시스템(100)은 디바이스(104)와 연관될 수 있는 하이브리드 변환 코딩 시스템을 포함하며, 이로써 디바이스(104)가 하이브리드 변환 코딩을 구현하도록 설계된 특수 용도 머신으로 변환된다. 이런 의미에서, 디바이스(104)는, 마이크로프로세서(110), 메모리(112), 예컨대 비-일시적 컴퓨터-판독가능 저장 매체, 파워(power), 주변기기들을 비롯한 다양한 엘리먼트들, 및 도 1에 명확히 도시되지 않은 다양한 다른 엘리먼트들을 포함할 수 있다. 메모리(112)는 하나 이상의 구역들, 예컨대 제 1 구역(112a) 및 제 2 구역(112b)을 포함할 수 있다. 디바이스(104)는 이미지들(예를 들어, 정지 이미지들 및/또는 비디오 이미지들)을 디스플레이하는데 사용될 수 있는 디스플레이(140)(예를 들어, 모니터 또는 어떤 다른 디스플레이)를 포함할 수 있다. 일부 예시들에서, 디바이스(104)를 구현하는데 유용할 수 있는 디바이스(104)의 다양한 다른 엘리먼트들이, 본 개시내용의 범주를 벗어남 없이, 추가되거나 포함될 수 있다.

- [0023] [0031] 일부 구현들에서, 시스템(100)은 이미지 또는 이미지들을 캡처하고 캡처된 이미지 또는 이미지들의 미압축(uncompressed) 이미지 데이터를 생성하도록 구성된 이미지저(114)를 포함할 수 있다. 이미지저(114)는 선택적 이미지를 전자 신호로 컨버팅하는 임의의 디바이스일 수 있다. 일 예에서, 이미지저(114)는 순차적 이미지들을 캡처하고 순차적으로 캡처된 이미지들의 미압축 이미지 데이터를 생성하도록 구성된 비디오 카메라일 수 있다. 이미지저(114)는, 이미지 또는 이미지들을 표현하는 이미지 신호들을 캡처하기 위한 하나 이상의 센서들(예를 들어, 초점면 배열과 같은 임의의 타입의 검출기)를 포함할 수 있다. 이미지저(114)의 센서들은 이미지 또는 이미지들의 캡처된 이미지 신호를 (예를 들어, 센서의 일부로서 포함된 또는 이미징 시스템(100)의 일부로서 센서와는 별개인 디지털-아날로그 컨버터를 통해) 디지털 데이터로 표현 또는 컨버팅하기 위해 제공된다.
- [0024] [0032] 일 예에서, 이미지저(114)는 단일 정지 이미지를 캡처하고 캡처된 이미지의 미압축 이미지 데이터를 생성하도록 구성된 정지 카메라(static camera)를 포함할 수 있다. 다른 예에서, 이미지저(114)는 복수의 순차적 이미지들을 캡처하고 순차적으로 캡처된 이미지들의 미압축 이미지 데이터를 생성하도록 구성된 비디오 카메라를 포함할 수 있다.
- [0025] [0033] 시스템(100)은 미압축 이미지 데이터를 포함하는 외부 소스로부터의 이미지 데이터를 수신하도록 구성된 입력 인터페이스(116)를 포함할 수 있다. 입력 인터페이스(116)는 다른 디바이스로부터의 스트리밍된 이미지 데이터(예를 들어, 정지 이미지들 또는 입력 비디오)를 수신하도록 구성될 수 있다. 입력 인터페이스(116)는 이미지 또는 이미지들을 표현하는 이미지 신호들을 수신하기 위한 하나 이상의 입력들을 포함할 수 있다. 입력 인터페이스(116)의 하나 이상의 입력들은 (예를 들어, 입력 인터페이스(116)의 일부로서 포함된 또는 디바이스(104)의 일부로서 입력 인터페이스(116)와는 별개인 아날로그-디지털 컨버터를 통해) 디지털 데이터로서 이미지 또는 이미지들의 이미지 신호들을 수신하기 위해 제공될 수 있다.
- [0026] [0034] 시스템(100)은 미압축 이미지 데이터를 수신하고 미압축 이미지 데이터의 하이브리드 변환 코딩을 수행하도록 구성된 인코더(120)를 포함할 수 있다.
- [0027] [0035] 인코더(120)는, 도 2a-4와 관련하여 아래에서 보다 상세히 논의되는 인트라-프레임 코딩을 제공하도록 구성된 하이브리드 변환 코더(122)를 포함할 수 있다.
- [0028] [0036] 시스템(100)은 또한, 이미지 데이터를 디코딩하도록 구성된 디코더(130)를 포함할 수 있다. 디코더(130)는 인코더(120)에 의해 행해진 인코딩을 디코딩함으로써 동작할 수 있으며, 이에 따라, 예를 들어, 오리지널 이미지 또는 비디오 정보가 리트리브될 수 있다. 다양한 구현들에서, 디코더(130)는 별개의 시스템에, 예컨대 디바이스(104)와 별개인 다른 컴퓨터 디바이스(미도시)에 포함될 수 있다. 이런 경우, 디바이스(104)는 인코더(120)를 사용하여 인코딩을 수행할 수 있고, 별개의 디바이스는 인코더(120)에 의해 행해진 인코딩을 디코딩할 수 있고, 이에 따라, 예로써, 오리지널 이미지가 별개의 디바이스에 의해 리트리브될 수 있다.
- [0029] [0037] 디바이스(104)의 구현에서, 이미지저(114)는 이미지 또는 이미지들을 캡처하고 캡처된 이미지 또는 이미지들의 미압축 이미지 데이터를 생성하도록 구성될 수 있다. 이 예시에서, 인코더(120)는 이미지저(114)로부터 미압축 이미지 데이터를 수신하고 메모리(112)의 제 1 구역(112a)에 미압축 이미지 데이터를 저장하도록 구성될 수 있다. 일 예에서, 이미지저(114)는 단일 정지 이미지를 캡처하고 캡처된 이미지의 미압축 이미지 데이터를 생성하도록 구성된 정지 카메라를 포함할 수 있다. 다른 예에서, 이미지저(114)는 복수의 순차적 이미지들을 캡처하고 순차적으로 캡처된 이미지들의 미압축 이미지 데이터를 생성하도록 구성된 비디오 카메라를 포함할 수 있다.
- [0030] [0038] 디바이스(104) 구현에서, 이미지저(114)는 이미지 또는 이미지들을 캡처하고 캡처된 이미지 또는 이미지들의 미압축 이미지 데이터를 생성하도록 구성될 수 있다. 이 예시에서, 마이크로프로세서(110)는 이미지저(114)로부터 미압축 이미지 데이터를 수신하고 미압축 이미지 데이터를 인코더(120)에 제공하도록 구성될 수 있다.
- [0031] [0039] 디바이스(104)의 구현에서, 입력 인터페이스(116)는 이미지 또는 이미지들의 이미지 데이터를 수신하도록 구성될 수 있다. 이 예시에서, 인코더(120)는 입력 인터페이스(116)로부터 이미지 데이터를 수신하고 메모리(112)의 제 1 구역(112a)에 이미지 데이터를 저장하도록 구성될 수 있다.
- [0032] [0040] 디스플레이(140)는 캡처된 이미지의 미압축 이미지 데이터를 디스플레이할 수 있다. 디스플레이(140)는 이미지 디스플레이 디바이스(예를 들어, LCD(liquid crystal display)) 또는 일부 다양한 다른 타입의 비디오 디스플레이 또는 모니터를 포함할 수 있다. 마이크로프로세서(110)는 디스플레이(140) 상에 이미지 데이터를 디스플레이하도록 적응될 수 있다. 마이크로프로세서(110)는 메모리(112)로부터 이미지 데이터를 리트리브 및 획득하고 디스플레이(140) 상에 임의의 리트리브된 이미지 데이터를 디스플레이하도록 구성될 수 있다. 디스플레이

레이(140)는, 이미지 데이터를 디스플레이하기 위해 마이크로프로세서(110)에 의해 활용될 수 있는 디스플레이 전자장치를 포함할 수 있다. 디스플레이(140)는 마이크로프로세서(110)를 통해 이미지(114)로부터 직접 이미지 데이터를 수신할 수 있고, 또는 마이크로프로세서(110)를 통해 메모리(112)로부터 이미지 데이터가 전송될 수 있다.

[0033] [0041] 도 1의 예에서, 시스템(100)은 다양한 컴포넌트들을 사용하여 예시된다. 그러나, 이러한 예시는 명확성 및 편의성을 위해 제공되며, 따라서, 일부 예시들에서, 다양한 컴포넌트들은 설명된 블록(들) 또는 모듈(들) 내에서 결합되거나 오버랩될 수 있고, 혹은 도 1의 예에 명확히 예시되지 않은 하나 이상의 블록(들) 또는 모듈(들)에 의해 구현될 수도 있다.

[0034] [0042] 도 2a는, 본 개시내용의 양상들에 따라, 하이브리드 변환 인코딩을 위해 사용될 수 있는 이미징 디바이스(200)의 예를 예시하는 블록도이다. 도 2a에 도시된 것처럼, 디바이스(200)는 이미지(210), 단일 명령 메모리 및 제어 프로세서(212)를 포함하는 인코더(214), 메모리(216), 디코더(204), 및 디스플레이(202)를 포함할 수 있다. 다양한 구현들에서, 디바이스(200)는, 모바일 폰들, 스마트폰들, 랩톱들, 텔레비전들, 또는 데스크탑 컴퓨터들을 비롯한 다양한 모바일 디바이스들 중 하나 이상의 것을 표현할 수 있다. 다양한 구현들에서, 이미지(210)는 디바이스(200)에 원격으로 연결될 수 있으며, 이미지는 다양한 모바일 디바이스들, 예컨대 모바일 폰들, 컨슈머 비디오 카메라들(예를 들어, 콤팩트, DSLR-비디오 등), 보안 비디오 카메라들, 또는 비디오를 디지털 방식으로 기록하고 저장하는 임의의 이미지 디바이스를 표현할 수 있다.

[0035] [0043] 이미지(210)는 이미지를 캡처하고 캡처된 이미지의 미압축 이미지 데이터를 생성할 수 있다. 이미지(210)는 또한 카메라 센서 또는 이미징 센서로 지칭될 수 있다. 이미지(210)는 복수의 순차적 이미지들을 캡처하고 순차적으로 캡처된 이미지들의 미압축 이미지 데이터를 생성하는 비디오 이미지를 포함할 수 있다. 다양한 예들에서, 이미지(210)는 하나 이상의 이미지들을 캡처하고 캡처된 이미지들의 미압축 이미지 데이터를 생성할 수 있다. 이미지(210)는 광학 이미지를 전자 신호로 컨버팅하는 임의의 디바이스일 수 있다. 이를테면, 이미지(210)는 복수의 순차적 이미지들을 캡처하고 순차적으로 캡처된 이미지들의 미압축 이미지 데이터를 생성하는 비디오 카메라일 수 있다.

[0036] [0044] 이미지(210)는 미압축 이미지 데이터를 인코더(214)에 제공할 수 있다. 인코더는 단일 명령 메모리 및 제어 프로세서(212)와 함께 동작할 수 있다. 단일 명령 메모리 및 제어 프로세서(212)는 인코더(214)가 버터플라이 구조 구현의 변환을 채용하게 할 수 있다. SIMD(single instruction multiple data) 동작들을 통한 병렬 컴퓨팅을 허용하는 버터플라이 구조 구현의 변환이 유리할 수 있고, 이는 도 3a-4와 관련하여 아래에서 보다 상세히 논의된다. SIMD 컴퓨터들, 예컨대 디바이스(200)는, 도 2b와 관련하여 아래에서 보다 상세히 논의되는 것처럼, 병렬로 다수의 데이터 아이템들에 동일한 동작들을 적용함으로써 데이터-레벨 병렬화(data-level parallelism)를 이용할 수 있다. 단일 명령 메모리 및 제어 프로세서(212)는 명령들을 페치하고(fetch) 그리고 디스패치한다(dispatche). 일 구현에서, 도 2a의 예를 참조로, 단일 명령 메모리 및 제어 프로세서(212)는 (예를 들어, 인코더(214)를 통해) 이미지(210)로부터 미압축 이미지 데이터 또는 이미지 신호들을 수신할 수 있고, 단일 명령 메모리 및 제어 프로세서(212)는 (예를 들어, 프로세싱된 이미지 데이터를 제공하기 위해) 수신된 이미지 신호들을 프로세싱하고, 인코딩된 비트스트림을 메모리(216)에 제공할 수 있다. 본원에 설명된 실시예들은, 예로써, 64-비트 플랫폼상에서 실행되는 SSE2(streaming SIMD extension 2)를 활용할 수 있다. 인코더(214)는, 도 3a-4와 관련하여 아래에서 보다 상세히 논의되는 것처럼, 미압축 이미지 데이터의 하이브리드 변환 코딩을 수행할 수 있다.

[0037] [0045] 일부 구현들에서, 디바이스는 또한, 메모리(216)로부터 리트리브된 잔차 픽셀 블록 데이터에 역변환(inverse transform)을 적용할 수 있는 디코더(204)를 포함할 수 있다. 따라서, 디코더(204)는 인코딩을 디코딩한 다음, 디스플레이(202)에 오리지널 이미지 데이터를 공급할 수 있다.

[0038] [0046] 도 2의 예에서, 시스템(200)은 다양한 컴포넌트들을 사용하는 것으로 예시된다. 그러나 이러한 예시는 명확성 및 편의성을 위해 제공되는 것이며, 따라서 일부 예시들에서, 다양한 컴포넌트들은 설명된 블록(들) 또는 모듈(들) 내에서 결합되거나 오버랩될 수 있고, 혹은 도 2의 예에 명확히 예시되지 않은 하나 이상의 블록(들) 또는 모듈(들)에 의해 구현될 수도 있다.

[0039] [0047] 도 2b는, 예를 들어, 인코딩 동작들과 함께 도 2a에 도시된 단일 명령 메모리 및 제어 프로세서(212)에 의해 수행될 수 있는 SIM 동작들의 예를 예시하는 블록도이다. 단일 명령 스트림, 다중 데이터 스트림(SIMD) 프로세서들에서, 하나의 명령은 몇 개의 프로세싱 엘리먼트들(PE들)을 사용함으로써 몇 개의 데이터 아이템들을 동시에 작동시키며, 이 몇 개의 프로세싱 엘리먼트들 모두는 도 2b에 예시된 것과 동일한 동작을 수행할 수 있

다. 단일 명령(218)은 PE들(222A 내지 222n)을 사용함으로써 동시에 몇 개의 데이터 아이템들(220)에 작동한다. PE들(222A 내지 222n)은, 예를 들어, ASIC(application specific integrated circuit)로 구현될 수 있다. SIMD 프로세서는, 단일 프로그램 카운터가 가리키는 명령들을 판독하고, 이들을 디코딩하고 제어 신호들을 PE들에 전송하는 단일 제어 유닛을 갖는다. 데이터, 예컨대 데이터 아이템들(220)은, PE들이 있는 것만큼 많은 데이터 경로들을 통해 PE들(222A 내지 222n)에 공급되고 이들로부터 유도될 수 있다.

[0040] [0048] 도 3a는 파이프라인에서 하이브리드 변환 코딩을 위해 사용될 수 있는 예시적 프로세스를 예시하는 흐름도이다. 도 3a의 동작들은, 예를 들어, 도 1에 도시된 인코더(120) 또는 도 2에 도시된 인코더(214)를 포함하는 시스템(100)의 이미징 디바이스(104)에 의해 수행될 수 있다. 프로세스는 비디오 데이터 블록을 식별하는 것(302)을 포함한다. 시스템은 잔차 값(residual value)들이 생성하기 위해 블록 예측을 구현(304)할 수 있다. 예를 들어, 시스템은 현재 픽셀 블록을 예측하기 위해 이전에 재구성된 픽셀들을 사용할 수 있다. 시스템은 잔차 픽셀 블록을 생성하기 위해 현재 픽셀 블록으로부터 상기 예측을 차감할 수 있다. 시스템은, 예를 들어 도 3b와 관련하여 아래에서 논의되는 것처럼 파이프라인에 하이브리드-변환을 적용(306)할 수 있다. 일 예로서, 시스템은 잔차 픽셀 블록에 버터플라이 ADST의 조합 DCT를 적용할 수 있고, 변환 계수들 및 코드 양자화 레벨들을 정량화할 수 있다. 이후 시스템은, 디코더, 예를 들어 도 1에 도시된 디코더(130)에 의한 사용을 위해 압축 결과를 생성(308)할 수 있다.

[0041] [0049] 도 3b는 하이브리드 변환 코딩을 위해 사용될 수 있는 예시적 프로세스를 예시하는 흐름도이다. 도 3의 동작들은, 예를 들어, 도 1에 도시된 시스템(100) 또는 이미징 디바이스(104)에 의해 수행될 수 있다.

[0042] [0050] 도 3b에 도시된 것처럼, 시스템(예를 들어, 도 1에 도시된 마이크로프로세서(110))은, 비디오 코덱을 사용하여 비디오 데이터 블록을 식별(330)할 수 있다. 시스템은, 커널이 원칙면에서(basis-wisely) 오리지널 ADST(asymmetric discrete sine transform)와 근사(즉, 비대칭을 기반으로 함)하며 DCT(discrete cosine transform) 커널과 일치하는 ADST 변환의 변형 및 DCT의 결합을 사용하여 비디오 데이터 블록에 인트라-프레임 코딩 방식(332)을 제공할 수 있다. 시스템은, (DCT를 포함할 수 있는) ADST의 변형을 사용하여, 인트라-프레임 코딩의 버터플라이 구조 구현을 산출(334)할 수 있다.

[0043] [0051] 공간 예측 및 변환 코딩

[0044] [0052] 부분적 예측 바운더리가 주어지면, ADST 변환의 변형을 유도하기 위해, 1-D 1차 가우스-마코프 모델(first-order Gauss-Markov model)의 맥락에서 오리지널 ADST를 유도한 수학적 이론이 검토된다.

[0045] [0053] 제로-평균, 단위 분산, 1차 가우스-마코프 시퀀스

[0046]
$$x_k = \rho x_{k-1} + e_k \quad (1)$$

[0047] 를 고려한다. 여기서 ρ 는 상관 계수이며, e_k 는 분산 $1 - \rho^2$ 을 갖는 백색 가우스 잡음 프로세스이다.

$\underline{x} = [x_1, x_2, \dots, x_N]^T$ 는, x_0 가 이용가능한 (편과적인) 바운더리로서 주어지면, 랜덤 벡터가 인코딩될 것임을 표시한다고 하자. 윗첨자 T 는 행렬 전치(matrix transposition)를 나타낸다. (1) 순환(recursion)은, 하기 세트의 식들로,

$$\begin{aligned} x_1 &= \rho x_0 + e_1 \\ x_2 - \rho x_1 &= e_2 \\ &\vdots \\ x_N - \rho x_{(N-1)} &= e_N, \end{aligned} \quad (2)$$

[0048]

[0049] 또는 컴팩트한 표기법(compact notation)으로는

[0050]
$$Q\underline{x} = \underline{b} + \underline{e} \quad (3) \text{로 전환되며,}$$

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ -\rho & 1 & 0 & 0 & \dots \\ 0 & -\rho & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & -\rho & 1 \end{pmatrix}, \quad (4)$$

여기서, 이다.

[0054] 그리고 $\underline{b} = [\rho x_0, 0, \dots, 0]^T$ 및 $\underline{e} = [e_1, e_2, \dots, e_N]^T$ 는, 각각, 바운더리 정보 및 이노베이션 프로세스 (innovation process)를 캡처한다. Q 는 도치가능하며, 따라서,

[0055] $\underline{x} = Q^{-1}\underline{b} + Q^{-1}\underline{e}$, (5) 임을 알 수 있다. 여기서, 뒹첨자 -1은 행렬 역변환(matrix inversion)을 나타낸다. 예상대로, (5)에서의 "바운더리 응답" 또는 예측 $Q^{-1}\underline{b}$ 은

$$Q^{-1}\underline{b} = [\rho x_0, \rho^2 x_0, \dots, \rho^N x_0]^T \quad (6)$$

[0056] 을 충족한다.

[0057] 예측 잔차

[0058] $\underline{y} = Q^{-1}\underline{e}$ (7) 는, 압축되어 송신될 것이며, 이는 그의 KLT의 유도를 유발한다.

$\underline{y}$ 의 자기상관 행렬은,

$$R_{\underline{y}\underline{y}} = E\{\underline{y}\underline{y}^T\} = Q^{-1}E\{\underline{e}\underline{e}^T\}(Q^T)^{-1} = (1 - \rho^2)Q^{-1}(Q^T)^{-1} \quad (8)$$

[0057] 로 주어진다.

[0059] 따라서, y 에 대한 KLT는 $Q^{-1}(Q^T)^{-1}$ 를 대각화하고 그리고 이에 따라 또한 더 편리한

$$P_1 = Q^T Q = \begin{pmatrix} 1 + \rho^2 & -\rho & 0 & 0 & \dots \\ -\rho & 1 + \rho^2 & -\rho & 0 & \dots \\ 0 & -\rho & 1 + \rho^2 & -\rho & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & -\rho & 1 + \rho^2 & -\rho \\ 0 & \dots & 0 & -\rho & 1 \end{pmatrix}. \quad (9)$$

[0060] 인 유니터리 행렬(unitary matrix)이다.

[0061] P_1 은 테플리츠(Toeplitz)지만, 하단 우측 코너에서의 엘리먼트는 주 대각선(principal diagonal)상의 모든 다른 엘리먼트들과 상이한데, 즉, 이는 $1 + \rho^2$ 가 아니다.

[0062] 이러한 불일치성(irregularity)은 P_1 의 고유값들 및 고유벡터들의 분석적 유도(analytic derivation)를 복잡하게 한다. 서브터퓨즈(subterfuge)로서, P_1 이

$$\hat{P}_1 = \begin{pmatrix} 1+\rho^2 & -\rho & 0 & 0 & \dots \\ -\rho & 1+\rho^2 & -\rho & 0 & \dots \\ 0 & -\rho & 1+\rho^2 & -\rho & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & -\rho & 1+\rho^2 & -\rho \\ 0 & \dots & 0 & -\rho & 1+\rho^2-\rho \end{pmatrix} \quad (10)$$

로 근사화되며, 이는 하단-우측 코너 엘리먼트를 $1-\rho^2-\rho$ 로 교체함으로써 얻어진다. 근사는 $\rho \rightarrow 1$ 에 대해 명확히 유지되며, 이는 사실상 비디오/이미지 신호들의 공간적 상관을 설명하는 근사이다. $\hat{P}$ 를 대각화하는 유니터리 행렬 T_s 및 이에 따라 $\mathbf{y}$ 의 요구되는 KLT에 대한 근사는, 공통 DST와 관련하여 다음과 같이 도출될 수 있다:

$$[T_s]_{j,i} = \left(\frac{2}{\sqrt{2N+1}} \sin \frac{(2j-1)i\pi}{2N+1} \right) \quad (11)$$

여기서, $j, i \in \{1, 2, \dots, N\}$ 는 각각 변환 커널의 주파수 인덱스 및 시간 인덱스이다.

[0062] ADST의 버터플라이 구조 변형

[0063] 위에서 유도된 ADST에 관한 관찰 결과(observation)는 T_s 의 로우(row)들(즉, 변환의 기본 함수들)이 초반에 더 작은 값들(인지된 바운더리에 더 가까움)을 갖고, 다른 단쪽에서는 더 큰 값들을 갖는다는 점이다. 이를테면, $j=1$ 인 로우(즉, 최저 주파수를 갖는 기본 함수)를 고려한다. $N \gg 1$ 인 경우, 제 1 샘플($i=1$)은 $\frac{2}{\sqrt{2N+1}} \sin \frac{\pi}{2N+1} \approx 0$ 인 반면, 마지막 샘플($i=N$)은 최대값 $\frac{2}{\sqrt{2N+1}} \sin \frac{N\pi}{2N+1} \approx \frac{2}{\sqrt{2N+1}}$ 를 취한다.

[0064] 이는, 인지된 바운더리에 더 가까운 픽셀들이 더 잘 예측되고 이에 따라 원단(far end)에서의 것들보다 통계적으로 더 작은 분산을 갖는다는 사실을 효율적으로 활용한다. ADST의 압축 성능과 유사한 유니터리 정현파 변환(unitary sinusoidal transform)은 ADST의 버터플라이 설계 및 이에 따른 병렬 컴퓨팅을 위한 하이브리드 변환 코딩의 복잡성을 해결할 수 있다.

[0065] DST의 새로운 변형은,

$$[T_{btf}]_{j,i} = \left(\sqrt{\frac{2}{N}} \sin \frac{(2j-1)(2i-1)\pi}{4N} \right) \quad (12)$$

로 제시되며, 여기서 $j, i \in \{1, 2, \dots, N\}$ 는, 각각 주파수 인덱스 및 시간 인덱스를 나타낸다. 이는 또한, 비대칭 기반 함수의 특성을 지니지만, DCT의 것과 일치하는 커널 인수의 분모 $4N$ 을 가지며, 이로써 버터플라이 구조 구현이 가능하다. 본 발명자들은 이를 btf-ADST로 지칭한다.

[0066] 실제로, 이들 모든 계산들은 정수 포맷으로 수행될 수 있으며, 이는 각각의 모든 스테이지에 걸쳐 누적되는 라운딩 효과들(rounding effects)을 초래할 수 있다. 라운드-트립 에러(round-trip error)를 최소화시키거나 감소시키기 위해, 구조는, 초기 스테이지들이 더 많은 곱셈들(multiplication)로 구성되게 변형되어, 라운딩 에러들이 덜 확대되게 된다. [6]에서 사용되는 규칙(convention)들을 지키는데 있어, $\bar{I}$ 및 D_N 이 재배열(re-ordering) 동작들을 나타낸다고 하자:

$$I = \begin{bmatrix} & & & 1 \\ & & \cdots & \\ & \cdots & & \\ 1 & & & \end{bmatrix}$$

$$D_N = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & \cdots & \\ & & & -1 \end{bmatrix} \quad (13)$$

P_J 는, 벡터 엔트리들의 제 1 절반부가 짝수 포지션으로 이동하고, 엔트리들 제 2 절반부가 홀수 포지션으로 그러나 역순(reversed order)으로 이동하는 치환 행렬이라고 하자:

$$P_J = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & \cdots & & 0 & 1 \\ 0 & 1 & \cdots & & \\ 0 & \cdots & & 1 & 0 \\ \cdots & \cdots & & & \cdots \end{bmatrix} \quad (14)$$

여기서 J는 행렬의 높이이다. 이는 제 2 치환을 포물레이팅한다:

$$H_N = P_N \begin{bmatrix} P_{N/2} & \\ & P_{N/2} \end{bmatrix} \cdots$$

$$\begin{bmatrix} P_4 & & & \\ & \bar{I}_4 P_4 \bar{I}_4 & & \\ & & \cdots & \\ & & P_4 & \\ & & & \bar{I}_4 P_4 \bar{I}_4 \end{bmatrix} \quad (15)$$

유사하게, 치환 연산자 Q_J 는 홀수 엔트리들을 역순이 되게 이동시킨다:

$$Q_J = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & \cdots & & 0 & 1 \\ 0 & 0 & 1 & \cdots & \\ 0 & \cdots & & 1 & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad (16)$$

$J = \log_2 N$ 라 하고, 본 발명자들은 버퍼플라이 구조를 포물레이팅하는 하기의 빌딩 블록들을 정의한다.

타입 1 전환 연산(translational Operation): 행렬들 $U_N(j)$, $j = 1, 2, \dots, J-1$,

$$U_N(j) = \begin{bmatrix} B(j) & & & \\ & B(j) & & \\ & & \cdots & \\ & & & B(j) \end{bmatrix} \quad (17)$$

로 정의되며, 여기서 $B(j) = \begin{bmatrix} I_{2^j} & I_{2^j} \\ I_{2^j} & -I_{2^j} \end{bmatrix} \quad (18)$

타입 2 회전 연산(Rotational Operation): $V_N(j)$, $j = 1, 2, \dots, J-1$ 은 블록 대각선 행렬이다:

$$V_N(j) = \begin{bmatrix} I_{2j} & & & \\ & E(j) & & \\ & & I_{2j} & \\ & & & \dots \\ & & & & E(j) \end{bmatrix} \quad (19)$$

여기서 $E(j) = \text{diag}\{T_{1/2j+1}, T_{5/2j+1}, \dots, T_{(2j+1-3)/2j+1}\}$ 이고,

$$T_r = \begin{bmatrix} \cos r\pi & \sin r\pi \\ \sin r\pi & -\cos r\pi \end{bmatrix} \quad (20)$$

[0071] 특별한 경우로서,

$$V_N(J) = \begin{bmatrix} T_{1/4N} & & & \\ & T_{5/4N} & & \\ & & \dots & \\ & & & T_{(2N-3)/4N} \end{bmatrix} \quad (21)$$

[0072] 상기 확립된 빌딩 블록들이 주어지면, btf-ADST는,

$$T_{btf} = D_N \cdot H_N^T \cdot V_N(1) \cdot U_N(1) \cdot V_N(2) \dots U_N(J-1) \cdot V_N(J) \cdot Q_N \cdot I_N, \quad (22)$$

로 분해될 수 있고, 이는 버터플라이 그래프로 직접 전환된다. btf-ADST는, 병렬 컴퓨팅을 사용하여 파이프라인에 버퍼플라이 구조 구현을 가능하게 하며, 이는 예를 들어 도 2b와 관련하여 앞서 설명된 것과 같은 프로세싱 엘리먼트들(222A 내지 222n)과 같은 다양한 프로세싱 엘리먼트들에 의해 수행될 수 있다. 예를 들어 도 2b와 관련하여 앞서 설명된 것처럼, SIMD 동작들과 함께 이러한 버터플라이 구조 구현은, 행렬 곱셈을 통해 연산되는 오리지널 ADST와 비교할 때 스피드-업(speed-up)을 제공한다.

[0073] 도 3c는 하이브리드 변환 코딩을 위해 사용될 수 있는 하드웨어 동작 흐름의 예를 예시하는 도면이다. 도 3c에 도시된 예는, 예로써, 변형 ADST의 8×8 버터플라이 구조 데이터 흐름을 도시한다. 도 3c에 도시된 예는 본원에서 설명된 변환 커널의 하드웨어 구현에 대한 하나의 프로토타입으로서의 역할을 할 수 있다. 버터플라이 엘리먼트들의 순서를 치환함으로써 동일 변환 커널의 상이한 구현을 산출하는 것이 가능하다.

[0074] 평균 CPU 사이클들의 면에서, 오리지널 ADST/DCT 하이브리드 변환 방식들과 비교한 btf-ADST/DCT의 런타임이 도 4에 도시된다. 도 4에 도시된 것처럼, y-축(410)은 다수의 CPU 사이클들을 묘사한다. x-축(420)은 변환 디멘션을 묘사한다. 구현은 SSE2(streaming SIMD extension 2)를 사용했으며 실험들은 64-bit 플랫폼상에서 실행되었다. 도 4에 도시된 것처럼, 라인(430)은 ADST/DCT 하이브리드 변환 방식 런타임들을 플롯팅하며, 라인(440)은 btf-ADST/DCT 하이브리드 변환 방식 런타임들을 플롯팅한다. 따라서, btf-ADST 구현은, 오리지널 ADST의 압축 이득들과 근접하게 유사하면서도, 효율적인 하드웨어 활용 및 이에 따른 상당한 코덱 스피드-업을 허용한다.

[0075] 이에 따라, 커널이 원칙면에서 오리지널 ADST(asymmetric discrete sine transform)와 근사한 ADST 변환의 이러한 변형은 DCT 커널과 일치하며, 이로써 버터플라이 구조 구현이 가능하다. 설명된 방식은, 하이브리드 변환 코딩 방식의 압축 성능을 유지하면서, 상당한 코덱 스피드-업을 위한 효율적 하드웨어 활용을 허용한다.

[0076] 도 5는 본원에 설명된 시스템에 사용될 수 있는 예시적인 디코딩 프로세스를 예시하는 흐름도이다. 도 5에 도시된 프로세스는, 예를 들어, 도 1에 도시된 디코더(130)와 같은 디코더에 의해 부분적으로 수행될 수 있다. 프로세스는 잔차 픽셀 블록 데이터를 식별(510)하는 것을 포함할 수 있다. 데이터 블록은, 예를 들어, 2D

잔차 픽셀 블록일 수 있다. 디코더는 잔차 픽셀 블록 데이터에 역변환을 적용(520)할 수 있다. 디코딩 프로세스는 역변환을 사용하는데, 이는 역순으로 식(22)에 도시된 각각의 스테이지(또는 행렬)의 전치(transpose)이다. 디코더는 역변환을 기초로 디코딩된 데이터를 생성(530)할 수 있다. 2D 잔차 픽셀 블록에 적용되는 변환의 예로써, 시스템은 2개의 변환들 T_a 및 T_c 및 잔차 블록 R 을 가질 수 있다. R 의 수직 방향에 T_a 를 적용하고, 수평 방향에 T_c 를 적용하는 것은 $T_a * R * T_c^t$ 로 전환되며, 여기서 t 는 행렬 전치를 나타낸다. 앞서 언급된 것처럼, 본원에 개시된 버터플라이 구조 하이브리드-변환은 SIMD 동작들을 통한 병렬 컴퓨팅을 사용하여 비트스트림의 고속 압축을 달성하도록 구현될 수 있다.

[0098] [0077] 도 6은, 본 개시내용의 양상들에 따라, 본원에 설명된 하나 이상의 시스템들 및 방법들을 구현하는데 사용될 수 있는 예시적인 또는 대표적인 컴퓨팅 디바이스들 및 연관된 엘리먼트들을 예시하는 블록도이다.

[0099] [0078] 구현에서, 도 6은, 본원에 설명되고 제공되는 것과 같은 양상들, 방법들 및 기술들에 따라 사용될 수 있는 컴퓨터 디바이스(600) 및 모바일 컴퓨터 디바이스(650)(예를 들어, 저전력 모바일 통신 디바이스를 포함하는 모바일 통신 디바이스, 예를 들어, 모바일 폰, 셀룰러 폰 등)의 예를 도시한다. 컴퓨팅 디바이스(600)는, 다양한 형태들의 디지털 컴퓨터들, 예컨대, 퍼스널 컴퓨터들, 랩톱들, 테블릿들, 데스크탑들, 워크스테이션들, 개인 정보 단말기들(personal digital assistants), 서버들, 블레이드 서버들, 메인프레임들 및 다른 적절한 컴퓨터들을 표현할 수 있다. 컴퓨팅 디바이스(650)는 다양한 형태들의 모바일 디바이스들, 예컨대 개인 정보 단말기들, 셀룰러 휴대폰들, 스마트 폰들, 및 다른 유사한 컴퓨팅 디바이스들을 표현할 수 있다. 본원에 도시된 컴포넌트들, 이들의 연결들 및 관계들 및 이들의 기능들은 단지 예시적인 것을 의미하며, 본 개시내용에 청구되는 그리고/또는 본원에 설명되는 구현들을 제한하는 것을 의미하지 않는다.

[0100] [0079] 컴퓨팅 디바이스(600)는 하나 이상의 프로세서들(602), 메모리(604), 저장 디바이스(606), 메모리(604)와 고속 확장 포트들(610)에 연결되는 고속 인터페이스(608), 그리고 저속 버스(614) 및 저장 디바이스(606)에 연결되는 저속 인터페이스(612)를 포함할 수 있다. 컴포넌트들(602, 604, 606, 608, 610, 및 612) 중 하나 이상의 것은 다양한 버스들을 사용하여 상호연결되며, 공통 마더보드 상에 또는 적절한 경우 다른 방식으로 장착될 수 있다. 구현에서, 프로세서(602)는 고속 인터페이스(608)에 커플링된 디스플레이(616)와 같은 외부 입/출력 디바이스 상에 GUI에 대한 그래픽 정보를 디스플레이하기 위해 메모리(604)에 또는 저장 디바이스(606) 상에 저장된 명령들을 비롯하여, 컴퓨팅 디바이스(600)내에서의 실행을 위한 명령들을 프로세싱하도록 구성될 수 있다. 다른 구현들에서, 적절한 경우, 다수의 메모리들 및 다수의 타입들의 메모리와 함께, 다수의 프로세서들 및/또는 다수의 버스들이 활용될 수 있다. 추가로, 다수의 컴퓨팅 디바이스들(600)은 (예를 들어, 서버 뱅크, 블레이드 서버들의 그룹, 또는 다중-프로세서 시스템처럼) 필요한 동작들의 부분들을 제공하는 디바이스와 연결될 수 있다.

[0101] [0080] 메모리(604)는 컴퓨팅 디바이스(600) 내에 정보를 저장하도록 구성될 수 있다. 구현에서, 메모리(604)는 하나 이상의 휘발성 메모리 유닛들을 포함할 수 있다. 다른 구현에서, 메모리(604)는 하나 이상의 비-휘발성 메모리 유닛들을 포함할 수 있다. 메모리(604)는, 자기 또는 광학 디스크와 같은 다른 형태의 비-일시적 컴퓨터-판독가능 매체를 포함할 수 있다.

[0102] [0081] 저장 디바이스(606)는 컴퓨팅 디바이스(600)에 대용량 저장부(mass storage)를 제공하도록 구성될 수 있다. 구현에서, 저장 디바이스(606)는, 스토리지 에어리어 네트워크(storage area network)에서의 사용을 위해 구성된 디바이스들 또는 다양한 다른 구성들을 비롯하여, 비-일시적 컴퓨터-판독가능 매체, 예컨대 플로피 디스크 디바이스, 하드 디스크 디바이스, 광학 디스크 디바이스, 또는 테이프 디바이스, 플래시 메모리, 또는 다른 유사한 고체 상태 메모리 디바이스, 또는 디바이스들의 어레이를 포함할 수 있다. 일부 구현들에서, 컴퓨터 프로그램 물건은 정보 캐리어에서 유형적으로 구체화될 수 있다. 컴퓨터 프로그램 물건은, 실행될 때, 하나 이상의 방법들, 예컨대 본원에서 설명된 것들을 수행하는 명령들을 포함할 수 있다. 다른 구현에서, 정보 캐리어는 비-일시적 컴퓨터-판독가능 매체 또는 비-일시적 머신-판독가능 매체, 예컨대 메모리(604), 저장 디바이스(606), 또는 프로세서(602) 상의 메모리를 포함할 수 있다.

[0103] [0082] 고속 제어기(608)는 컴퓨팅 디바이스(600)에 대한 대역폭-집약적(bandwidth-intensive) 동작들을 관리하도록 구성될 수 있는 반면, 저속 제어기(612)는 낮은 대역폭-집약적 동작들을 관리하도록 구성될 수 있다. 이러한 기능들의 할당은 단지 예시적일 수 있다. 구현에서, 고속 제어기(608)는 (예를 들어, 그래픽 프로세서 또는 가속기를 통해) 메모리(604), 디스플레이(616)에, 그리고/또는 다양한 확장 카드들(미도시)을 수락하도록 구성될 수 있는 고속 확장 포트들(610)에 커플링될 수 있다. 구현에서, 저속 제어기(612)는 저장 디바이스(606) 및/또는 저속 확장 포트(614)에 커플링될 수 있으며, 여기서, 다양한 통신 포트들(예를 들어, USB,

BLUETOOTH, ETHERNET, 무선 ETHERNET 등)을 포함할 수 있는 저속 확장 포트는 예를 들어 네트워크 어댑터를 통해, 하나 이상의 입/출력 디바이스들, 예컨대 키보드, 포인팅 디바이스, 스캐너, 또는 네트워킹 디바이스, 예컨대 스위치 또는 라우터에 커플링될 수 있다.

[0104] [0083] 컴퓨팅 디바이스(600)는 도 6에 도시된 방식으로, 다수의 상이한 형태들로 구현될 수 있다. 예를 들어, 컴퓨팅 디바이스(600)는 표준 서버(620)로서 또는 이러한 서버들의 그룹으로 여러 번 구현될 수 있다. 컴퓨팅 디바이스(600)는 랙 서버 시스템(624)의 일부로서 구현될 수 있다. 추가로, 컴퓨팅 디바이스(600)는, 랩톱 컴퓨터(622)와 같은 PC(personal computer)로 구현될 수 있다. 다른 구현에서, 컴퓨팅 디바이스(600)로부터의 컴포넌트들은 모바일 디바이스(미도시), 예컨대 디바이스(650)의 다른 컴포넌트들과 결합될 수 있다. 이러한 디바이스들 중 하나 이상의 것은 컴퓨팅 디바이스들(600, 650) 중 하나 이상의 것을 포함할 수 있고, 전체 시스템은 서로 통신하는 다수의 컴퓨팅 디바이스들(600, 650)을 구성할 수 있다.

[0105] [0084] 컴퓨팅 디바이스(650)는, 다양한 다른컴포넌트들 중에서도, 하나 이상의 프로세서들(652), 메모리(664), 입/출력 디바이스, 예컨대 디스플레이(654), 통신 인터페이스(666) 및 트랜시버(668)를 포함할 수 있다. 디바이스(650)에는, 추가의 저장을 제공하기 위해, 저장 디바이스, 예컨대 마이크로-드라이브 또는 일부 다른 관련된 디바이스가 제공될 수 있다. 컴포넌트들(650, 652, 664, 654, 666, 및 668) 중 하나 이상의 것은 다양한 버스들을 사용하여 상호연결될 수 있고, 컴포넌트들 중 몇 개는 공통 마더보드 상에 또는 적절하게 다른 방식으로 장착될 수 있다.

[0106] [0085] 프로세서(652)는, 메모리(664)에 저장된 명령들을 비롯하여, 컴퓨팅 디바이스(650) 내의 명령들을 실행하도록 구성될 수 있다. 프로세서(652)는 별개의 그리고 다수의 아날로그 및 디지털 프로세서들을 포함하는 칩들의 칩셋으로서 구현될 수 있다. 구현에서, 프로세서(652)는 디바이스(650)의 다른 컴포넌트들의 조정, 예컨대 사용자 인터페이스들의 제어, 디바이스(650)에 의한 애플리케이션 실행, 및 디바이스(650)에 의한 무선 통신을 위해 제공될 수 있다.

[0107] [0086] 프로세서(652)는 디스플레이(654)에 커플링된 디스플레이 인터페이스(656) 및 제어 인터페이스(658)를 통해 사용자와 통신하도록 구성될 수 있다. 디스플레이(654)는, 예를 들어, TFT LCD(Thin-Film-Transistor Liquid Crystal Display) 또는 OLED(Organic Light Emitting Diode) 디스플레이, 또는 다른 적절한 디스플레이 기술을 포함할 수 있다. 디스플레이 인터페이스(656)는 사용자에게 그래픽 및 다른 정보를 제시하기 위해 디스플레이(654)를 구동시키기 위한 적절한 회로를 포함할 수 있다. 제어 인터페이스(658)는 사용자로부터의 커맨드들을 수신하고 프로세서(652)에게의 제출을 위해 이들을 컨버팅할 수 있다. 구현에서, 외부 인터페이스(662)는 다양한 다른 디바이스들과 디바이스(650)의 근거리 통신(near area communication)을 가능하게 하기 위해 프로세서(652)와 통신하게 제공될 수 있다. 일 예에서, 외부 인터페이스(662)는 일부 구현들에서는 유선 통신을 위해, 또는 다른 구현들에서는 무선 통신을 위해 제공될 수 있으며, 다수의 인터페이스들이 활용될 수 있다.

[0108] [0087] 메모리(664)는 컴퓨팅 디바이스(650)내에 정보를 저장하도록 구성될 수 있다. 메모리(664)는 비-일시적 컴퓨터-판독가능 매체 또는 매체들, 하나 이상의 휘발성 메모리 유닛들, 또는 하나 이상의 비-휘발성 메모리 유닛들 중 하나 이상의 것으로 구현될 수 있다. 확장 메모리(674)가 제공될 수 있고 그리고 이 확장 메모리(674)는 예를 들어, SIMM(Single In Line Memory Module) 카드 인터페이스를 포함할 수 있는 확장 인터페이스(672)를 통해 디바이스(650)에 연결될 수 있다. 이러한 확장 메모리(674)는 디바이스(650)를 위한 여분의 저장 공간을 제공할 수 있거나, 또는 디바이스(650)의 애플리케이션들 또는 다른 정보를 또한 저장할 수 있다. 특정하게, 일 예에서, 확장 메모리(674)는 앞서 설명된 프로세스들을 실행하거나 보완하는 명령들을 포함할 수도 있고 보안 정보를 포함할 수도 있다. 따라서, 예를 들어, 확장 메모리(674)는 디바이스(650)에 대한 보안 모듈로서 제공될 수 있고, 디바이스(650)의 보안 사용을 허용하는 명령들로 프로그램될 수 있다. 게다가, 보안 방식으로 SIMM 카드상에 식별 정보를 배치하는 것과 같은 추가의 정보와 함께, SIMM 카드들을 통해 보안 애플리케이션들이 제공될 수 있다.

[0109] [0088] 메모리는, 아래에서 논의되는 것처럼, 예를 들어, 플래시 메모리 및/또는 NVRAM 메모리를 포함할 수 있다. 일 구현에서, 컴퓨터 프로그램 물건은 정보 캐리어에 유형적으로 구체화된다. 컴퓨터 프로그램 물건은, 실행될 때, 앞서 설명된 것들과 같은 하나 이상의 방법들을 수행하는 명령들을 포함한다. 정보 캐리어는, 컴퓨터-판독가능 매체 또는 머신-판독가능 매체, 예컨대, 메모리(664), 확장 메모리(674), 또는, 예를 들어, 트랜시버(668) 또는 외부 인터페이스(662)를 통해 수신될 수 있는 프로세서(652) 상의 메모리이다.

[0110] [0089] 디바이스(650)는 필요한 경우 디지털 신호 프로세싱 회로를 포함할 수 있는 통신 인터페이스(666)를 통해 무선으로 통신하도록 구성될 수 있다. 구현에서, 통신 인터페이스(666)는 다양한 모드들 또는 프로토콜들

하에서의 통신들, 예컨대, 특히, GSM 음성 통화들, SMS, EMS, 또는 MMS 메시징, CDMA, TDMA, PDC, WCDMA, CDMA2000, 또는 GPRS을 위해 제공될 수 있다. 일 예에서, 이러한 통신은, 예를 들어, 무선-주파수(radio-frequency) 트랜시버(668)를 통해 발생할 수 있다. 게다가, 단거리 통신이, 예컨대 블루투스, WiFi, 또는 다른 이러한 트랜시버(미도시)를 사용하여 발생할 수 있다. 더 추가로, GPS(Global Positioning System) 수신기 모듈(670)은 추가의 내비게이션-관련 및/또는 로케이션-관련 무선 데이터를 디바이스(650)에 제공할 수 있고, 디바이스(650)는 적절한 경우 디바이스(650) 상에서 실행되는 애플리케이션들에 의해 사용될 수 있다.

[0111] [0090] 디바이스(650)는, 사용자가 말한 정보를 수신하고 이를 사용가능한 디지털 정보로 컨버팅할 수 있는 오디오 코덱(660)을 사용하여 음향적으로(audibly) 통신하도록 구성될 수 있다. 일 예에서, 오디오 코덱(660)은 마찬가지로, 예를 들어 디바이스(650)의 핸드셋에서, 예컨대 스피커를 통해, 사용자를 위한 음향 사운드를 생성할 수 있다. 다양한 구현들에서, 이러한 사운드는 음성 전화 통화들로부터의 사운드를 포함할 수 있고, 레코딩된 사운드(예를 들어, 음성 메시지들, 음악 파일들 등)를 포함할 수 있고, 그리고 디바이스(650) 상에서 동작하는 애플리케이션들에 의해 생성된 사운드를 포함할 수 있다.

[0112] [0091] 컴퓨팅 디바이스(650)는 도 6에 도시된 방식으로 다수의 상이한 폼들로 구현될 수 있다. 예를 들어, 컴퓨팅 디바이스(650)는 셀룰러 전화 및/또는 일부 다른 저전력 모바일 통신 디바이스들을 포함하는 모바일 통신 디바이스(680)로서 구현될 수 있다. 다른 예에서, 컴퓨팅 디바이스(650)는 스마트 폰(682), 개인 정보 단말기, 또는 일부 다른 유사한 모바일 디바이스의 일부로서 구현될 수 있다.

[0113] [0092] 따라서, 본원에서 설명되는 시스템들, 방법들 및 기술들의 다양한 구현들이 디지털 전자 회로, 집적 회로, 특정하게 설계된 ASIC들(application specific integrated circuits), 컴퓨터 하드웨어, 펌웨어, 소프트웨어, 및/또는 이들의 조합들로 실현될 수 있다. 이들 다양한 구현들은, 특수용 또는 범용성이며 저장 시스템, 적어도 하나의 입력 디바이스 및 적어도 하나의 출력 디바이스로부터 데이터 및 명령들을 수신하도록 그리고 이들로 데이터 및 명령들을 송신하도록 커플링될 수 있는, 적어도 하나의 프로그램가능한 프로세서를 포함하는 프로그램가능 시스템상에서 실행가능한 그리고/또는 해석가능한 하나 이상의 컴퓨터 프로그램들에서의 구현을 포함할 수 있다.

[0114] [0093] 이들 컴퓨터 프로그램들(또한, 프로그램들, 소프트웨어, 소프트웨어 애플리케이션들 또는 코드로 공지됨)은 프로그램가능 프로세서를 위한 머신 명령들을 포함하고, 하이-레벨 절차 및/또는 객체-지향 프로그래밍 언어로, 그리고/또는 어셈블리/머신 언어로 구현될 수 있다. 본원에서 사용되는 것처럼, "머신-판독가능 매체", "컴퓨터-판독가능 매체"란 용어는, 머신-판독가능 신호로서 머신 명령들을 수신하도록 구성된 비-일시적 머신-판독가능 매체를 비롯하여, 프로그램가능 프로세서에 머신 명령들 및/또는 데이터를 제공하기 위해 사용되는 임의의 컴퓨터 프로그램 물건, 장치 및/또는 디바이스(예를 들어, 자기 디스크들(disks), 광학 디스크들(disks), 메모리, PLD((Programmable Logic Device)들)를 지칭한다. 다양한 예들에서, "머신-판독가능 신호"란 용어는 프로그램가능한 프로세서에 머신 명령들 및/또는 데이터를 제공하기 위해 사용되는 임의의 신호를 지칭할 수 있다.

[0115] [0094] 구현에서, 사용자와의 상호작용을 제공하기 위해, 본원에 설명된 시스템들, 방법들 및 기술들은, 사용자에게 정보를 디스플레이하기 위한 디스플레이 디바이스(예를 들어, CRT(cathode ray tube) 또는 LCD(liquid crystal display) 모니터) 및 사용자가 컴퓨터에 입력을 제공할 수 있게 하는 키보드 및 포인팅 디바이스(예를 들어, 마우스 또는 트랙볼))를 갖는 컴퓨터상에서 구현될 수 있다. 마찬가지로 사용자와의 상호작용을 제공하기 위해 다른 타입들의 디바이스들이 사용될 수 있다; 예를 들어, 사용자에게 제공되는 피드백은 임의의 형태의 감각 피드백(예를 들어, 시각적 피드백, 청각적 피드백, 또는 촉각적 피드백)일 수 있고; 사용자로부터의 입력은, 청각적, 스피치, 또는 촉각적 입력을 비롯한 임의의 형태로 수신될 수 있다.

[0116] [0095] 다양한 예들에서, 본원에 설명되는 시스템들, 방법들 및 기술들은 (예를 들어, 데이터 서버로서의) 백 엔드(back end) 컴포넌트를 포함하거나, 미들웨어(middleware) 컴포넌트(예를 들어, 애플리케이션 서버)를 포함하거나, 또는 프론트 엔드(front end) 컴포넌트(예를 들어, 본원에 설명된 시스템들, 방법들, 및 기술들의 구현과 사용자가 상호작용할 수 있게 하는 웹 브라우저 또는 그래픽 사용자 인터페이스를 갖는 클라이언트 컴퓨터)를 포함하거나, 또는 이러한 백 엔드, 미들웨어, 또는 프론트 엔드 컴포넌트들의 임의의 조합을 포함하는 컴퓨팅 시스템에서 구현될 수 있다. 시스템의 컴포넌트들은 임의의 형태 또는 매체의 디지털 데이터 통신(예를 들어, 통신 네트워크)에 의해 상호연결될 수 있다. 통신 네트워크들의 예들은 "LAN(local area network)", "WAN(wide area network)" 및 인터넷을 포함한다.

[0117] [0096] 컴퓨팅 시스템은 클라이언트와 서버를 포함할 수 있다. 클라이언트와 서버는 일반적으로 서로 원격적이

며, 통상적으로 통신 네트워크를 통해 상호 작용한다. 클라이언트와 서버의 관계는, 컴퓨터 프로그램들이 각각의 컴퓨터들 상에서 실행되며 서로 클라이언트-서버 관계를 가짐으로써 발생한다.

[0118] [0097] 특별히 상세하게 설명된 상기 실시예들은 단지 예이거나 가능한 실시예들이며, 포함될 수 있는 많은 다른 조합들, 추가들, 또는 대안들이 존재한다.

[0119] [0098] 컴포넌트들의 특정한 명명, 대문자 사용 용어들, 속성들, 데이터 구조들 또는 임의의 다른 프로그래밍 또는 구조적 양상은 필수적이거나 중요하지 않으며, 본 개시내용의 양상들 또는 이들의 특징들을 구현하기 위해 사용될 수 있는 메커니즘들은 상이한 이름들, 포맷들, 또는 프로토콜들을 가질 수 있다. 게다가, 시스템은, 설명된 것처럼, 하드웨어 및 소프트웨어의 결합을 통해 또는 전적으로 하드웨어 엘리먼트들에서 구현될 수 있다. 게다가 본원에 설명된 다양한 시스템 컴포넌트들 간의 기능의 임의의 특정한 구분은 단지 예시적이며, 필수적인 것은 아니다; 단일 시스템 컴포넌트에 의해 수행되는 기능들은 대신 다수의 컴포넌트들에 의해 수행될 수 있고, 다수의 컴포넌트들에 의해 수행되는 기능들은 대신 단일 컴포넌트에 의해 수행될 수 있다.

[0120] [0099] 게다가, 도면들에 도시된 논리적 흐름들이 원하는 결과들을 달성하기 위해 도시된 특정 순서, 또는 순차적 순서를 요구하는 것은 아니다. 또한, 다른 단계들이 제공될 수 있거나, 또는 설명된 흐름들로부터 단계들이 삭제될 수 있고, 다른 컴포넌트들이 설명된 시스템들로부터 추가 또는 제거될 수 있다. 따라서, 다른 실시예들이 하기 청구항들의 범주내에 속한다.

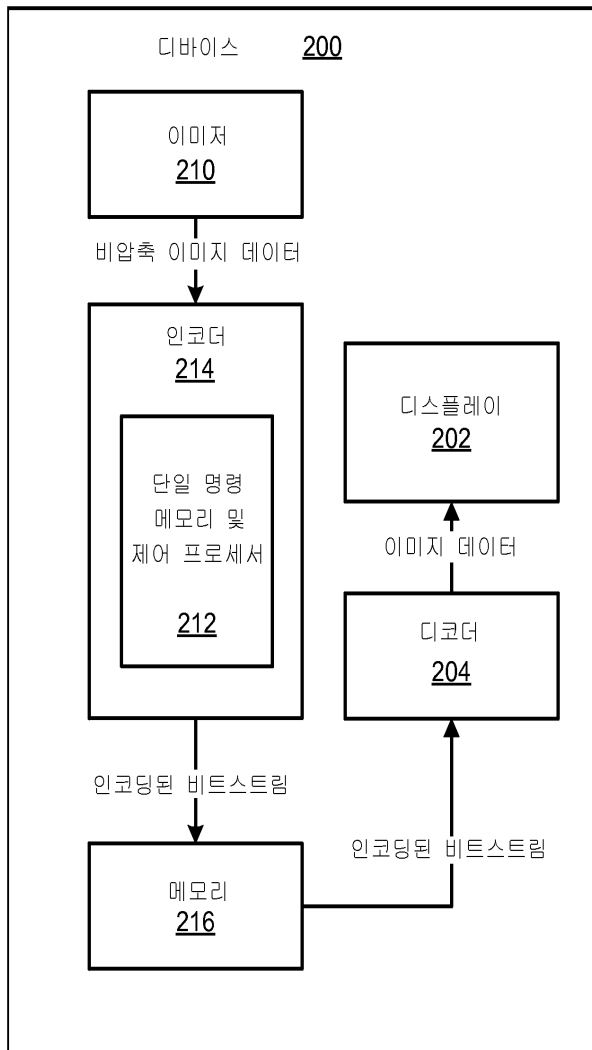
도면

도면1

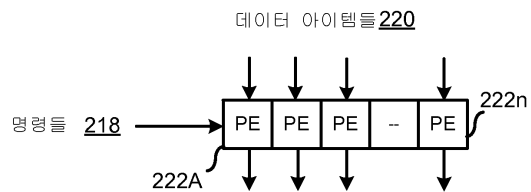
100



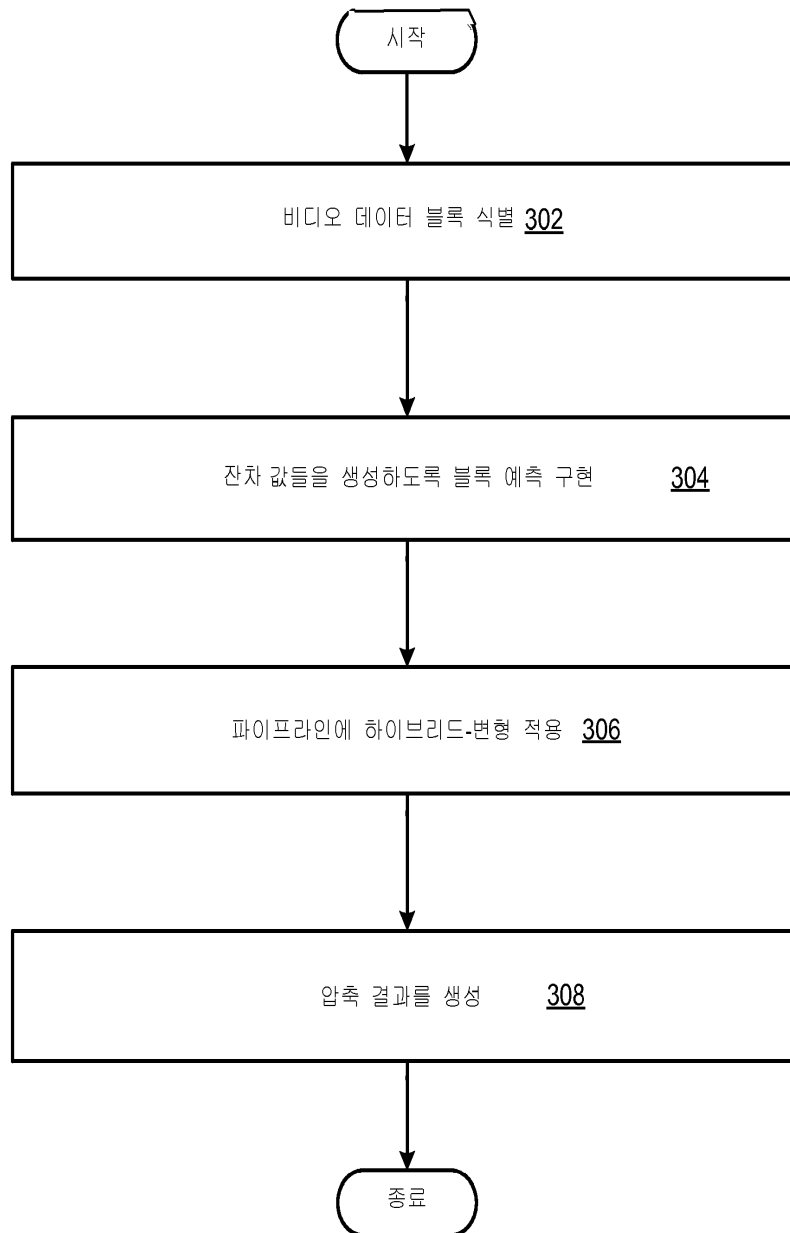
도면2a



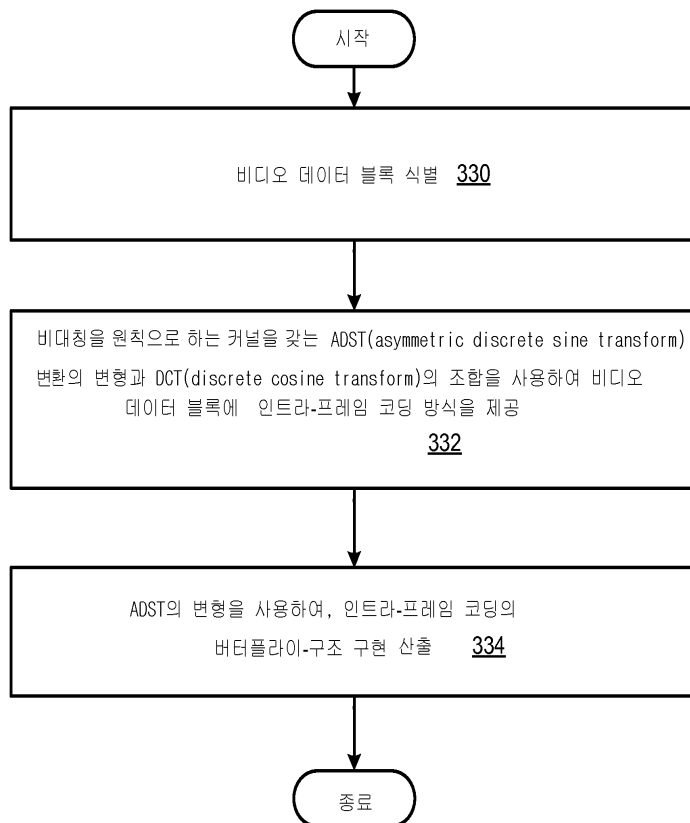
도면2b



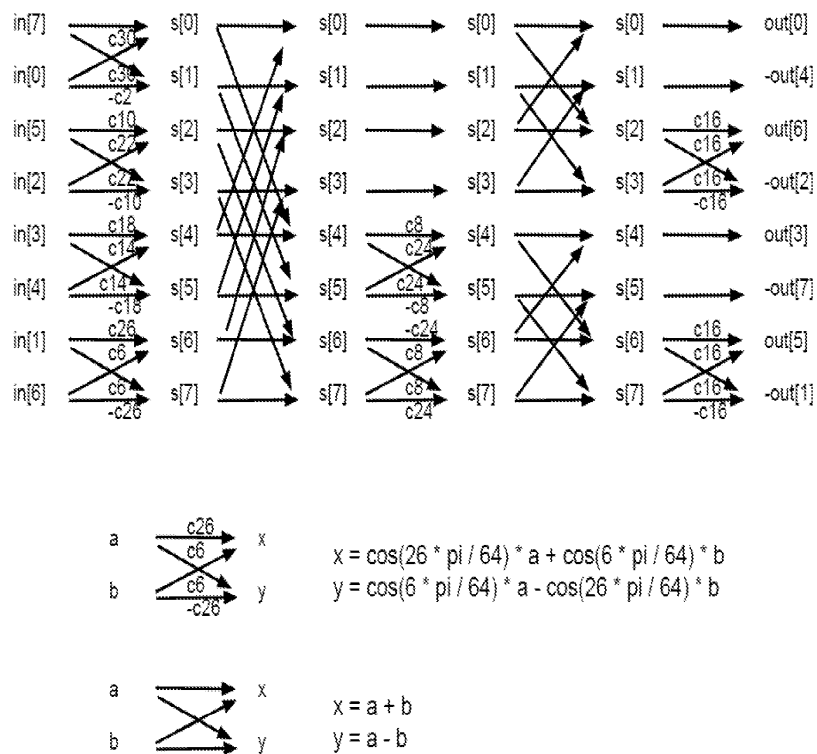
도면3a



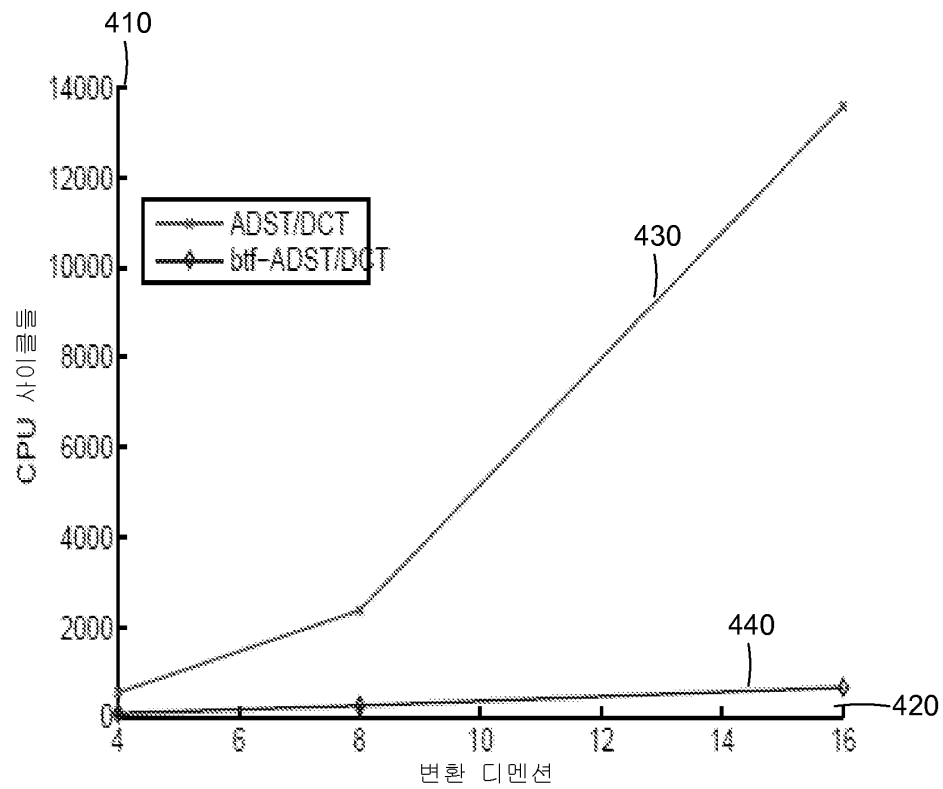
도면3b



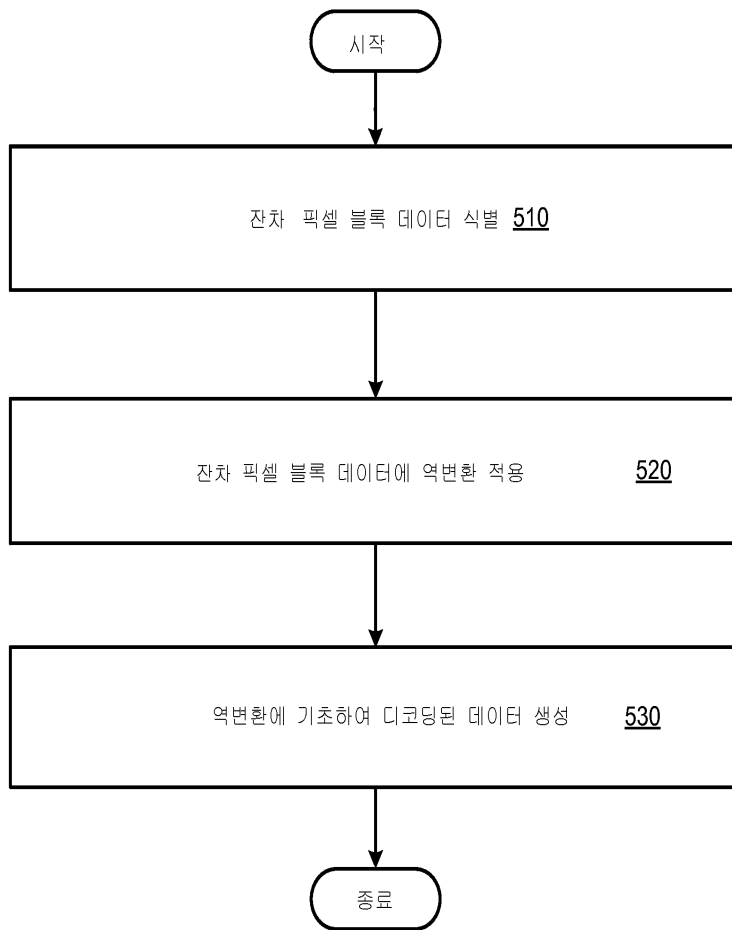
도면3c



도면4



도면5



도면6

