(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0126890
(43) 공개일자 2017년11월20일

(51) 국제특허분류(Int. Cl.)
H04N 19/523 (2014.01) *H04N 19/105* (2014.01)
H04N 19/117 (2014.01) *H04N 19/176* (2014.01)
H04N 19/182 (2014.01) *H04N 19/423* (2014.01)
H04N 19/59 (2014.01) *H04N 19/70* (2014.01)
H04N 19/80 (2014.01)
(52) CPC특허분류
H04N 19/523 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2017-7024674
(22) 출원일자(국제) 2016년02월03일
심사청구일자 없음
(85) 번역문제출일자 2017년09월01일
(86) 국제출원번호 PCT/US2016/016325
(87) 국제공개번호 WO 2016/144443
국제공개일자 2016년09월15일
(30) 우선권주장
62/129,299 2015년03월06일 미국(US)
14/710,967 2015년05월13일 미국(US)

(71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
치엔 웨이-정
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
왕 샹린
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
자오 신
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 30 항

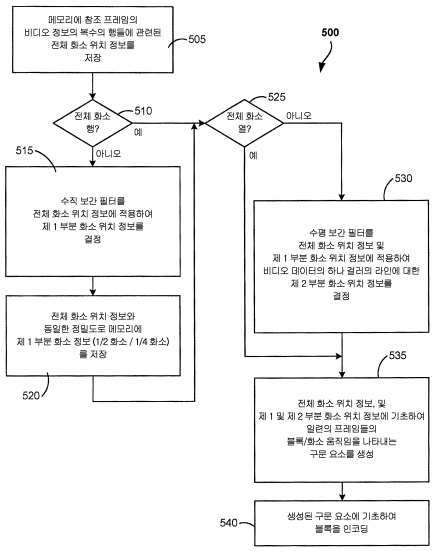
(54) 발명의 명칭 **움직임 검색에서 낮은 복잡도 1/4 화소 생성을 위한 방법 및 장치**

(57) 요 약

본 개시는 비디오 코딩을 위한 움직임 검색에서 낮은 복잡도의 1/4 화소 생성을 위한 시스템 및 방법을 제공한다. 그 방법은 메모리에 참조 프레임의 비디오 정보의 복수의 행에 관련된 전체 화소 (full-pixel) 위치 정보를 저장하는 단계를 포함할 수 있다. 그 방법은 또한, 상기 참조 프레임에 관련된 비디오 정보에 대

(뒷면에 계속)

대 표 도 - 도5



한 상기 전체 화소 위치 정보에 수직 보간 필터를 적용하여 제 1 부분 화소 위치 정보를 결정하는 단계를 포함할 수 있다. 이 방법은 또한, 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 데이터의 하나 걸러의 행에 대한 제 2 부분 화소 위치 정보를 결정하는 단계를 포함할 수 있다. 그 방법은 또한 제 1 부분 화소 위치 정보에 기초하여 현재 프레임의 화소 움직임 (pixel motion) 을 나타내는 구문 요소 (syntax element) 를 생성하는 단계를 포함할 수 있다. 그 방법은 또한 생성된 구문 요소에 기초하여 블록을 인코딩하는 단계를 포함할 수 있다.

(52) CPC특허분류
H04N 19/117 (2015.01)
H04N 19/176 (2015.01)
H04N 19/182 (2015.01)
H04N 19/423 (2015.01)
H04N 19/59 (2015.01)
H04N 19/70 (2015.01)
H04N 19/80 (2015.01)

명 세 서

청구범위

청구항 1

비디오 데이터를 인코딩하는 방법으로서,

메모리에 참조 프레임의 비디오 정보의 복수의 행에 관련된 전체 화소 위치 정보를 저장하는 단계;

검색된 위치가 상기 참조 프레임 내의 전체 화소 행 상에 위치하지 않을 때 참조 프레임에 관련된 비디오 정보에 대한 상기 전체 화소 위치 정보에 수직 보간 필터를 적용하여 제 1 부분 화소 위치 정보를 결정하는 단계;

상기 검색된 위치가 상기 참조 프레임 내의 전체 화소 열 상에 위치되지 않을 때 상기 전체 화소 위치 정보의 하나 걸러의 행 및 상기 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 하나 걸러의 라인에 대해 제 2 부분 화소 위치 정보를 결정하는 단계;

상기 전체 화소 위치 정보의 하나 걸러의 행, 상기 제 1 부분 화소 위치 정보, 및 상기 제 2 부분 화소 위치 정보에 기초하여 현재 프레임의 화소 움직임을 나타내는 구문 요소를 생성하는 단계; 및

생성된 상기 구문 요소에 기초하여 블록을 인코딩하는 단계

를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 수평 보간 필터를 적용하기 전에 상기 메모리에 상기 제 1 부분 화소 위치 정보를 저장하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 참조 프레임에 관련된 비디오 정보의 하나 걸러의 행에 상기 수직 보간 필터를 적용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 4

제 2 항에 있어서,

상기 메모리에 저장된 상기 전체 화소 위치 정보에 기초하여 상기 참조 프레임에 관련된 복수의 1/2 화소 위치를 미리 계산하는 단계; 및

상기 메모리에 상기 1/2 화소 위치에 관련된 1/2 화소 위치 정보를 저장하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 5

비디오 데이터를 인코딩하는 방법으로서,

메모리에 참조 프레임에 관련된 전체 화소 위치 정보를 저장하는 단계;

검색된 위치가 상기 참조 프레임 내의 전체 화소 행 상에 위치하지 않을 때:

상기 참조 프레임에 관련된 비디오 정보에 대한 상기 전체 화소 위치 정보의 하나 걸러의 행에 수직 보간 필터를 적용하여 제 1 부분 화소 위치 정보를 결정하는 단계, 및

상기 메모리에 상기 제 1 부분 화소 위치 정보를 저장하는 단계;

검색된 위치가 상기 참조 프레임 내의 전체 화소 열 상에 위치하지 않을 때:

상기 전체 화소 위치 정보 및 상기 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 상기 비디오 정보의 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정하는 단계;

상기 전체 화소 위치 정보의 하나 걸러의 행, 상기 제 1 부분 화소 위치 정보, 및 상기 제 2 부분 화소 위치 정보에 기초하여 현재 프레임의 화소 움직임을 나타내는 구문 요소를 생성하는 단계; 및

생성된 상기 구문 요소에 기초하여 블록을 인코딩하는 단계

를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 6

제 5 항에 있어서,

상기 메모리에 저장된 상기 전체 화소 위치 정보에 기초하여 상기 참조 프레임에 관련된 복수의 1/2 화소 위치를 미리 계산하는 단계; 및

상기 메모리에 상기 1/2 화소 위치에 관련된 1/2 화소 위치 정보를 저장하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 1/2 화소 위치 정보를 저장하는 단계는 상기 수직 보간 필터를 적용하기 전에 일어나는, 비디오 데이터를 인코딩하는 방법.

청구항 8

제 5 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는, 2개의 인접하는 전체 화소 위치 사이 중간에 위치하는 1/2 화소 위치를 나타내는 1/2 화소 위치 정보를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 9

제 5 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는, 1/2 화소 위치와 인접하는 전체 화소 위치 사이 중간에 위치된 1/4 화소 위치를 나타내는 1/4 화소 위치 정보를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 10

제 5 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는 전체 화소 위치 정보와 동일한 정밀도로 상기 메모리에 저장되는, 비디오 데이터를 인코딩하는 방법.

청구항 11

제 5 항에 있어서,

상기 참조 프레임과 상기 현재 프레임 사이의 화소 움직임은 벡터로 표시되고, 상기 벡터는 상기 제 1 부분 화소 위치 정보 및 상기 제 2 부분 화소 위치 정보 중 적어도 하나를 참조하는, 비디오 데이터를 인코딩하는 방법.

청구항 12

제 5 항에 있어서,

상기 검색된 위치가 상기 참조 프레임 내의 전체 화소 행 상에 위치될 때 상기 전체 화소 위치 정보에 기초하여 상기 현재 프레임의 화소 움직임을 나타내는 상기 구문 요소를 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

제 5 항에 있어서,

상기 검색된 위치가 상기 참조 프레임 내의 전체 화소 열 상에 위치될 때 상기 전체 화소 위치 정보에 기초하여 상기 현재 프레임의 화소 움직임을 나타내는 상기 구문 요소를 생성하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 14

비디오를 인코딩하기 위한 디바이스로서,

움직임 검색을 위해 메모리에 일련의 프레임에 관련된 전체 화소 위치 정보를 저장하도록 구성된 메모리;

상기 메모리에 동작가능하게 연결된 프로세서를 포함하고,

상기 프로세서는

검색된 위치가 참조 프레임의 전체 화소 행 상에 위치하지 않을 때, 상기 일련의 프레임에 관련된 비디오 정보에 대한 상기 전체 화소 위치 정보에 수직 보간 필터를 적용하여 상기 비디오 정보에 대한 제 1 부분 화소 위치 정보를 결정하고; 그리고

전체 화소 위치 정보와 동일한 정밀도로 상기 메모리에 상기 제 1 부분 화소 위치 정보를 저장하고; 그리고

검색된 위치가 상기 참조 프레임의 전체 화소 열 상에 위치하지 않을 때, 상기 전체 화소 위치 정보의 하나 걸러의 행 및 상기 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 상기 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정하고;

상기 전체 화소 위치 정보, 상기 제 1 부분 화소 위치 정보, 및 상기 제 2 부분 화소 위치 정보에 기초하여 상기 일련의 프레임들의 화소 움직임을 나타내는 구문 요소를 생성하고; 그리고

생성된 상기 구문 요소에 기초하여 블록을 인코딩하도록 구성된, 비디오를 인코딩하기 위한 디바이스.

청구항 15

제 14 항에 있어서,

상기 프로세서는 또한 상기 참조 프레임에 관련된 비디오 정보의 하나 걸러의 행에 상기 수직 보간 필터를 적용하여 상기 비디오 정보에 대한 상기 제 1 부분 화소 위치 정보를 결정하도록 구성된, 비디오를 인코딩하기 위한 디바이스.

청구항 16

제 14 항에 있어서,

상기 프로세서는 또한

상기 메모리에 저장된 상기 전체 화소 위치 정보에 기초하여 상기 참조 프레임에 관련된 1/2 화소 위치 정보를 미리 계산하고; 그리고

상기 메모리에 복수의 1/2 화소 위치에 관련되는 1/2 화소 위치 정보를 저장하도록 구성된, 비디오를 인코딩하기 위한 디바이스.

청구항 17

제 16 항에 있어서,

상기 프로세서는 상기 수직 보간 필터를 적용하기 전에 상기 1/2 화소 위치 정보를 저장하도록 구성된, 비디오를 인코딩하기 위한 디바이스.

청구항 18

제 14 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는, 2개의 인접하는 전체 화소 위치 사이 중간에 위치하는 1/2 화소 위치를 나타내는 1/2 화소 위치 정보를 포함하는, 비디오를 인코딩하기 위한 디바이스.

청구항 19

제 14 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는, 1/2 화소 위치와 인접하는 전체 화소 위치 사이 중간에 위치된 1/4 화소 위치를 나타내는 1/4 화소 위치 정보를 포함하는, 비디오를 인코딩하기 위한 디바이스.

청구항 20

제 14 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는 전체 화소 위치 정보와 동일한 정밀도로 상기 메모리에 저장된, 비디오를 인코딩하기 위한 디바이스.

청구항 21

제 14 항에 있어서,

상기 참조 프레임과 현재 프레임 사이의 화소 움직임은 벡터로 표시되고, 상기 벡터는 상기 제 1 부분 화소 위치 및 상기 제 2 부분 화소 위치 정보 중 적어도 하나를 참조하는, 비디오를 인코딩하기 위한 디바이스.

청구항 22

비디오 데이터를 인코딩하기 위한 장치로서,

움직임 검색을 위해 메모리에 일련의 프레임에 관련된 전체 화소 위치 정보를 저장하도록 구성된 저장 수단;

상기 저장 수단에 동작가능하게 연결된 프로세싱 수단으로서, 검색된 위치가 참조 프레임의 전체 화소 행 상에 위치하지 않을 때 상기 전체 화소 위치 정보에 수직 보간 필터를 적용하여 상기 일련의 프레임에 관련된 제 1 부분 화소 위치 정보를 결정하고; 그리고

전체 화소 위치 정보와 동일한 정밀도로 상기 메모리에 상기 제 1 부분 화소 위치 정보를 저장하고; 그리고

검색된 위치가 상기 참조 프레임의 전체 화소 열 상에 위치하지 않을 때 상기 전체 화소 위치 정보의 하나 걸러의 행 및 상기 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 상기 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정하도록 구성된, 상기 프로세싱 수단;

상기 전체 화소 위치 정보, 제 1 및 제 2 의 1/2 화소 위치 정보, 또는 제 1 및 제 2 의 1/4 화소 위치 정보에 기초하여 상기 일련의 프레임들의 화소 움직임을 나타내는 구문 요소를 생성하는 생성 수단; 및

생성된 상기 구문 요소에 기초하여 블록을 인코딩하도록 구성된 인코딩 수단

을 포함하는, 비디오 데이터를 인코딩하기 위한 장치.

청구항 23

제 22 항에 있어서,

저장 수단은 메모리를 포함하고 상기 프로세싱 수단은 하나 이상의 프로세서를 포함하는, 비디오 데이터를 인코딩하기 위한 장치.

청구항 24

제 22 항에 있어서,

상기 프로세싱 수단은 또한, 상기 메모리에 저장된 상기 전체 화소 위치 정보에 기초하여 상기 참조 프레임에 관련된 복수의 1/2 화소 위치를 미리 계산하도록 구성된, 비디오 데이터를 인코딩하기 위한 장치.

청구항 25

제 23 항에 있어서,

상기 저장 수단은 또한, 상기 1/2 화소 위치에 관련된 상기 1/2 화소 위치 정보를 저장하도록 구성된, 비디오 데이터를 인코딩하기 위한 장치.

청구항 26

제 22 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는 1/2 화소 위치 정보를 포함하고, 1/2 화소 위치는 2개의 인접하는 전체 화소 위치 사이 중간에 위치한, 비디오 데이터를 인코딩하기 위한 장치.

청구항 27

제 22 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는, 1/2 화소 위치와 인접하는 전체 화소 위치 사이 중간에 위치된 1/4 화소 위치를 나타내는 1/4 화소 위치 정보를 포함하는, 비디오 데이터를 인코딩하기 위한 장치.

청구항 28

제 22 항에 있어서,

상기 제 1 및 제 2 부분 화소 위치 정보는 전체 화소 위치 정보와 동일한 정밀도로 상기 메모리에 저장된, 비디오 데이터를 인코딩하기 위한 장치.

청구항 29

제 22 항에 있어서,

상기 참조 프레임과 현재 프레임 사이의 화소 움직임은 벡터로 표시되고, 상기 벡터는 상기 제 1 부분 화소 위치 및 상기 제 2 부분 화소 위치 정보 중 적어도 하나를 참조하는, 비디오 데이터를 인코딩하기 위한 장치.

청구항 30

제 22 항에 있어서,

상기 프로세싱 수단은 또한, 검색된 위치가 참조 프레임의 전체 화소 행 상에 위치하지 않을 때 상기 전체 화소 위치 정보의 하나 걸러의 행에 수직 보간 필터를 적용하여 상기 일련의 프레임에 관련된 비디오 정보의 하나 걸러의 라인에 대한 제 1 부분 화소 위치 정보를 결정하도록 구성된, 비디오 데이터를 인코딩하기 위한 장치.

발명의 설명

기 술 분 야

[0001] 본 개시는 비디오 코딩 및 압축의 분야에 관한 것이다. 특히, 본 개시는 고효율 비디오 코딩 (HEVC) 에 대한 움직임 추정 및 움직임 보상과 관련된다.

배 경 기 술

[0002] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기 (PDA) 들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 전자책 단말기, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은, MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding) 에 의해 정의되는 표준들, 현재 개발중인 HEVC 표준, 및 그러한 표준들의 확장들에 설명된 것들과 같은 비디오 코딩 프로세스들을 구현한다. 비디오 디바이스들은, 그러한 유형의 비디오 코딩을 구현함으로써 보다 효율적으로 디지털 비디오 정보를 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0003] 비디오 코딩 방법들은, 비디오 시퀀스들에 내재하는 중복성 (redundancy) 을 감소시키거나 또는 제거하기 위해

공간 (화상내) 예측 및/또는 시간 (화상간) 예측을 포함한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (예를 들어, 비디오 프레임, 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티션될 수도 있고, 이 비디오 블록들은 트리 블록들, 코딩 유닛 (CU) 들, 및/또는 코딩 노드들로도 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 화상에서 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오블록들은 동일한 화상에서의 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있고, 참조 화상들은 참조 프레임들로 지칭될 수도 있다.

[0004] 공간 또는 시간 예측은 코딩될 블록을 위한 예측 블록을 낳는다. 잔차 데이터는 코딩될 원래 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 지시하는 움직임 벡터와, 코딩된 블록 및 예측 블록 사이의 차이를 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 낳을 수도 있으며, 다음으로 이들은 양자화될 수도 있다. 초기에 2 차원 행렬로 배열된, 양자화된 변환 계수들은,변환 계수들의 1 차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩이 적용되어 훨씬 더 많은 압축을 달성할 수도 있다.

발명의 내용

[0005] 개요

[0006] 일반적으로, 본 개시는, 비디오 코딩에 관련된 시스템 및 방법들을 설명한다. 본 개시의 시스템, 방법 및 디바이스는 각각 여러 혁신적인 양태들을 가지며, 그들 중 단 하나만이 오로지 여기에 개시된 바람직한 속성들의 원인이 되지는 않는다.

[0007] 본 개시의 하나의 양태는 비디오를 인코딩하는 방법을 제공한다. 그 방법은 메모리에 참조 프레임의 비디오 정보의 복수의 행에 관련된 전체 화소 (full-pixel) 위치 정보를 저장하는 단계를 포함할 수 있다. 그 방법은 또한, 검색된 위치가 참조 프레임 내의 전체 화소 행 상에 위치하지 않을 때 참조 프레임에 관련된 비디오 정보에 대한 전체 화소 위치 정보에 수직 보간 필터를 적용하여 제 1 부분 화소 (sub-pel) 위치 정보를 결정하는 단계를 포함할 수 있다. 그 방법은 또한 메모리에 제 1 부분 화소 위치 정보를 저장하는 단계를 포함할 수 있다. 그 방법은 또한 제 1 부분 화소 위치 정보에 기초하여 현재 프레임의 화소 움직임 (pixel motion) 을 나타내는 구문 요소 (syntax element) 를 생성하는 단계를 포함할 수 있다. 그 방법은 또한 생성된 구문 요소에 기초하여 블록을 인코딩하는 단계를 포함할 수 있다.

[0008] 본 개시의 또 다른 양태는 비디오를 인코딩하는 또 다른 방법을 제공한다. 그 방법은 메모리에 참조 프레임에 관련된 전체 화소 위치 정보를 저장하는 단계를 포함할 수 있다. 검색된 위치가 참조 프레임 내의 전체 화소 행 (full-pixel row) 상에 위치하지 않을 때, 그 방법은 또한, 참조 프레임에 관련된 비디오 정보의 하나 걸러의 행에 대한 전체 화소 위치 정보에 수직 보간 필터를 적용하여 제 1 부분 화소 위치 정보를 결정하는 단계를 포함할 수 있다. 그 방법은 또한 메모리에 제 1 부분 화소 위치 정보를 저장하는 단계를 포함할 수 있다. 검색된 위치가 참조 프레임 내의 전체 화소 열 (full-pixel column) 상에 위치하지 않을 때, 그 방법은 또한, 전체 화소 위치 정보 및 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정하는 단계를 포함할 수 있다. 그 방법은 또한, 전체 화소 위치 정보, 제 1 부분 화소 위치 정보, 및 제 2 부분 화소 위치 정보에 기초하여 현재 프레임의 화소 움직임을 나타내는 구문 요소 (syntax element) 를 생성하는 단계를 포함할 수 있다. 그 방법은 또한 생성된 구문 요소에 기초하여 블록을 인코딩하는 단계를 포함할 수 있다.

[0009] 본 개시의 또 다른 양태는 비디오를 인코딩하기 위한 디바이스를 제공한다. 그 디바이스는 움직임 검색을 위해 메모리에 일련의 프레임에 관련된 전체 화소 위치 정보를 저장하도록 구성된 메모리를 가질 수 있다. 그 디바이스는 또한 메모리에 동작 가능하게 연결된 프로세서를 가질 수 있다. 검색된 위치가 참조 프레임의 전체 화소 행 상에 위치하지 않는 경우, 프로세서는 일련의 프레임에 관련된 비디오 정보에 대한 전체 화소 위치 정보에 수직 보간 필터를 적용하여 비디오 정보의 하나 걸러의 행에 대한 제 1 부분 화소 위치 정보를 결정할 수 있다. 그 프로세서는 또한 전체 화소 위치 정보와 동일한 정밀도로 메모리에 제 1 부분 화소 위치 정보를 저장할 수 있다. 검색된 위치가 참조 프레임의 전체 화소 열 상에 위치하지 않는 경우, 그 프로세서는, 전체 화소 위치 정보 및 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정할 수 있다. 그 프로세서는 또한, 전체 화소 위치 정보,

제 1 부분 화소 위치 정보, 및 제 2 부분 화소 위치 정보에 기초하여 일련의 프레임들의 화소 움직임을 나타내는 구문 요소를 생성할 수 있다. 그 프로세서는 또한 생성된 구문 요소에 기초하여 블록을 인코딩할 수 있다.

[0010] 본 개시의 또 다른 양태는 비디오 데이터를 인코딩하기 위한 장치를 제공한다. 그 장치는 움직임 검색을 위해 메모리에 일련의 프레임에 관련된 전체 화소 위치 정보를 저장하도록 구성된 저장 수단을 가질 수 있다. 그 장치는 또한 저장 수단에 동작가능하게 연결된 프로세싱 수단을 가질 수 있다. 그 프로세싱 수단은 검색된 위치가 참조 프레임의 전체 화소 행 상에 위치하지 않을 때 전체 화소 위치 정보에 수직 보간 필터를 적용하여 일련의 프레임에 관련된 비디오 정보의 하나 걸러의 라인에 대한 제 1 부분 화소 위치 정보를 결정할 수 있다. 그 프로세싱 수단은 또한 전체 화소 위치 정보와 동일한 정밀도로 메모리에 제 1 부분 화소 위치 정보를 저장할 수 있다. 그 프로세싱 수단은 또한, 검색된 위치가 참조 프레임의 전체 화소 열 상에 위치하지 않을 때 전체 화소 위치 정보 및 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 비디오 정보의 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정할 수 있다. 그 장치는 또한, 전체 화소 위치 정보, 제 1 및 제 2 의 1/2 화소 (half-pixel) 위치 정보, 또는 제 1 및 제 2 의 1/4 화소 (quarter-pixel) 위치 정보에 기초하여 일련의 프레임들의 화소 움직임을 나타내는 구문 요소를 생성하는 생성 수단을 가질 수 있다. 그 장치는 또한 생성된 구문 요소에 기초하여 블록을 인코딩하도록 구성된 인코딩 수단을 가질 수 있다.

[0011] 본 개시의 다른 특징 및 장점들은, 예로써, 본 개시의 양태들을 예시하는 다음의 설명으로부터 분명해질 것이다.

도면의 간단한 설명

[0012] 본 개시의 실시형태들의 구조 및 동작 양자 모두에 대한 상세들은 부분적으로 첨부 도면들의 연구에 의해 얻어질 수도 있고, 여기서 같은 참조 부호는 같은 부분들을 나타내고, 첨부 도면들 중에서:

도 1은 비디오 코딩 시스템의 기능 블록도이고;

도 2는 도 1의 비디오 인코더의 기능 블록도이고;

도 3은 도 1의 비디오 디코더의 기능 블록도이고;

도 4a 는 움직임 검색을 위한 참조 프레임의 그래픽 표현이고;

도 4b 는 움직임 검색 동안 코딩된 비디오 데이터 프레임 및 참조 프레임의 그래픽 표현이고; 그리고

도 5는 움직임 검색의 방법을 예시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0013] 첨부된 도면과 관련하여 아래에 제시되는 상세한 설명은 다양한 실시형태들의 설명으로서 의도된 것이며 본 개시가 실시될 수도 있는 실시형태들만을 나타내도록 의도된 것은 아니다. 상세한 설명은 실시형태들의 완전한 이해를 제공하는 목적을 위해 특정 상세들을 포함한다. 하지만, 본 개시는 이들 특정 상세들을 갖지 않을 수도 있음이 당업자에게 분명할 것이다. 일부 경우에, 잘 알려진 구조와 컴포넌트들은 설명의 간략화를 위해 단순화된 형태로 보여진다.

[0014] **비디오 코딩**

[0015] 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual 및 ITU-T H.264 (ISO/IEC MPEG-4 AVC 으로도 알려짐) 을 포함하며, 그의 SVC (Scalable Video Coding) 및 MVC (Multi-view Video Coding) 확장판을 포함한다. 이들 표준들 중 하나 이상이 본 명세서에 설명된 실시형태들과 함께 사용될 수 있다.

[0016] 또한, HEVC (High-Efficiency Video Coding) 이, ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발되었다. 최신 HEVC 초안 명세 (그리고, 이하 "HEVC WD") 는, http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-O1003-v1.zip 로부터 입수가능하다. HEVC 에 대한 멀티 뷰 확장, 즉 MV-HEVC 및 보다 진보된 3D 비디오 코딩 (3D-HEVC) 을 위한 또 다른 HEVC 확장이 JCT-3V 에 의해 개발되고 있으며, 한편 HEVC에 대한 확장 가능한 비디오 코딩 확장, 즉 SHVC 가 JCT-VC에 의해 개발되고 있다. MV-HEVC 의 최근 작업 초안 (WD) (이하 MV-HEVC WD5) 는 http://phenix.it-

sudparis.eu/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1004-v6.zip 로부터 입수가능하다. 3D-HEVC WD1의 최신 WD (이하 3D-HEVC WD1) 는 http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/6_Geneva /wg11/JCT3V-F1001-v3.zip 로부터 입수가능하다. SHVC 의 최근 작업 초안 (WD) (이하 SHVC WD3) 는 http://phenix.it-sudparis.eu/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-O1008-v3.zip 로부터 입수가능하다.

[0017] HEVC는 일반적으로 증가된 계산 복잡도의 대가로, 비교 가능한 이미지 품질로 비트레이트 요구사항을 절반 만큼 줄일 수 있다. 일부 실시형태들에서, HEVC 인코더들은 계산 복잡성, 압축률, 에러에 대한 견고성 및 인코딩 지연 시간을 절충 (trade off) 할 수 있다. 따라서, HEVC는 일반적으로 보다 높은 해상도 비디오 및 개선된 병렬 프로세싱 방법을 지원할 수 있다.

[0018] HEVC 비디오 코딩 계층은 화상간 (inter-picture) 및 화상내 (intra-picture) 예측 및 2차원 (2D) 변환 코딩을 구현하는 "하이브리드" 접근법을 사용한다. HEVC 인코더는 먼저, 화상내 예측을 이용하는 랜덤 액세스 포인트의 제 1 화상 또는 제 1 화상에 대한 블록 모양의 영역으로 화상을 분할할 수 있다. 본 명세서에서 사용된 화상내 예측은 일반적으로 그 화상 내의 정보에만 기초한 화상에서의 블록의 예측을 지칭할 수 있다. 모든 다른 화상에 대해서는, 화상간 예측이 사용된다. 화상간 예측은 일반적으로 예측 정보가 현재 화상을 생성하기 위해 다른 화상으로부터 사용되는 경우를 지칭할 수 있다. 예측 방법이 끝나고 화상이 루프 필터를 통과한 후에, 최종 화상 표현이 디코딩된 화상 버퍼에 저장된다. 디코딩된 화상 버퍼에 저장된 화상은 다른 화상의 예측을 위해 사용될 수 있다.

[0019] HEVC 표준은 파라미터 세트 개념을 채용한다. HEVC 에서의 시퀀스 파라미터 세트 (SPS), 화상 파라미터 세트 (PPS) 및 비디오 파라미터 세트 (VPS) 메커니즘은 코딩된 비디오 블록 데이터의 송신으로부터 드물게 변화하는 정보의 송신을 분리시킬 수도 있다.

[0020] HEVC에서, SPS 화상, PPS 또는 VPS의 식별자는 'ue(v)' 라는 이름의 가변 길이 인코딩 코딩 방법을 사용하여 코딩된다. HEVC 작업 초안은 왼쪽 비트 먼저 무부호 (unsigned) 정수 Exp-Golomb-코딩된 구문 요소로서 'ue(v)' 코딩된 구문 요소를 정의한다. 각 SPS 는 SPS 식별자 (ID) 를 포함하고, 각 PPS는 PPS ID 및 SPS ID 를 포함하고, 각 슬라이스 헤더는 PPS ID를 포함한다. 각각의 버퍼링 기간 보충 강화 정보 (SEI) 메시지는 또한 SPS ID를 포함한다. 일반적으로, 비디오 코딩 디바이스 또는 비디오 프로세싱 디바이스는 네트워크 추상화 계층 (NAL) 유닛 내의 비트스트림의 데이터를 캡슐화할 수도 있다. NAL 유닛들은, VCL (Video Coding Layer) NAL 유닛들 및 비-VCL NAL 유닛들을 포함한다. VCL NAL 유닛들은 일반적으로, 코딩된 비디오 데이터 및 파라미터 세트, 예컨대 SPS, PPS 및 VPS와 같은 비디오 인코더에 의해 사용되는 정보를 포함한다. 비-VCL NAL 유닛들 SEI 메시지를 정의하는 정보를 포함하고, 이들은 VCL NAL 유닛들로부터 코딩된 화상 샘플들을 디코딩하는데 필요하지 않는 정보를 포함할 수도 있지만, 디코딩, 디스플레이, 에러 레질런스 (error resilience) 및 다른 목적들에 관한 프로세스들을 도울 수도 있다.

[0021] **비디오 코딩 시스템**

[0022] 도 1은 비디오 코딩 시스템의 기능 블록도이다. 비디오 코딩 시스템 ("시스템") (10) 은 본 개시에서 설명되는 양태에 따른 방법을 이용할 수도 있다. 여기에 기재 사용된, 용어 "비디오 코더" 는 비디오 인코더 및 비디오 디코더 양자 모두를 총칭적으로 지칭한다. 본 개시에서, 용어 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 및 비디오 디코딩을 총칭적으로 지칭할 수도 있다.

[0023] 도 1에 도시된 바처럼, 시스템 (10) 은, 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 생성할 수 있는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋 이를테면 소위 "스마트" 폰들, 소위 "스마트" 패드, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 재생기들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 어느 것을 포함할 수 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 갖추어질 수도 있다.

[0024] 목적지 디바이스 (14) 는, 링크 (16) 를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 링크 (16) 는, 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16) 는, 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스 (14) 로 직접 송신할 수 있게 하기 위한 통신 매체

를 포함할 수도 있다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 무선 통신 프로토콜 등의 통신 표준에 따라 변조하고, 목적지 디바이스 (14) 로 송신할 수도 있다. 통신 매체는 라디오 주파수 (radio frequency; RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는, 로컬 영역 네트워크, 와이드 영역 네트워크, 또는 인터넷과 같은 글로벌 네트워크 등의 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터, 스위치, 기지국, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 가능하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0025] 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (34) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (34) 로부터 액세스될 수도 있다. 저장 디바이스 (34) 는, 하드 드라이브, 블루레이 디스크, DVD, CD-ROM, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체 등의 다양한 분산형 또는 로컬 적으로 액세스되는 데이터 저장 매체 중 어느 것을 포함할 수도 있다. 다른 예에서, 저장 디바이스 (34) 는, 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오를 유지할 수도 있는, 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는, 스트리밍 또는 다운로드를 통해 저장 디바이스 (34) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은, (예를 들어, 웹사이트용) 웹 서버, FTP 서버, NAS (network attached storage) 디바이스, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는, 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터를 액세스하기 위해 적당한 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스 (34) 로부터 인코딩된 비디오 데이터의 송신은, 스트리밍 송신, 다운로드 송신 또는 양자 모두의 조합일 수도 있다.

[0026] 본 개시의 방법들은 무선 애플리케이션들 또는 세팅들에 반드시 한정되는 것은 아니다. 그 방법들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트, 케이블 텔레비전 송신, 위성 텔레비전 송신, 예를 들어 인터넷을 통한 스트리밍 비디오 송신, 데이터 저장 매체에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체에 저장된 디지털 비디오의 디코딩, 또는 다른 어플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중 어느 것을 지원하는 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 및/또는 화상 통화등의 애플리케이션들을 지원하기 위하여 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0027] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 포함한다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 소스, 이를테면 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스, 및/또는 소스 비디오로서 컴퓨터 그래픽스 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 그러한 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 하지만, 본 개시에 설명된 방법들은, 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 응용들에 적용될 수도 있다.

[0028] 비디오 인코더 (20) 는 캡처된, 이전에 캡처된 또는 컴퓨터 생성된 비디오를 인코딩할 수도 있다. 또한, 일부 실시형태들에서, 비디오 디코더 (30) 는 제 1 비트스트림의 파라미터 세트 ID 및 고유 파라미터 세트 ID에 기초하여 제 1 비트스트림과 제 2 비트스트림을 구별할 수도 있다. 다른 예에서, 소스 디바이스 (12) 의 출력 인터페이스 (22) 는 인코딩된 비디오 데이터를 직접적으로 목적지 디바이스 (14) 로 송신할 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로), 디코딩 및/또는 플레이백을 위해, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 나중 액세스를 위해 저장 디바이스 (34) 에 저장될 수도 있다.

[0029] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신되거나, 또는 저장 디바이스 (34) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩함에 있어서, 비디오 디코더 (30) 와 같은 비디오 디코더에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성

된 다양한 구문 요소들을 포함할 수도 있다. 그러한 구문 요소들은, 통신 매체 상에서 송신되거나, 저장 매체 상에 저장되거나, 또는 파일 서버에 저장되는 인코딩된 비디오 데이터와 함께 포함될 수도 있다.

[0030] 디스플레이 디바이스 (32) 는, 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는, 통합된 디스플레이 디바이스를 포함할 수도 있고 또한 외부 디스플레이 디바이스와 인터페이스 접속하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 표시하고, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 어느 것을 포함할 수도 있다.

[0031] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 반도체 (ASIC), 필드 프로그램가능 게이트 어레이 (FPGA), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 회로 중 어느 것으로서 구현될 수도 있다. 그 방법들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 적합한 비일시적 컴퓨터 판독가능 매체에 그 소프트웨어를 위한 명령들을 저장하고 본 개시의 방법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 쪽이 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다.

[0032] HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초한다. HM 은 예컨대, ITU-TH.264/AVC 에 따른 기존 디바이스들에 비하여 비디오 코딩 디바이스들의 여러 추가적인 능력들을 상정한다. 예를 들어, H.264 는 9개의 인트라 예측 인코딩 모드들을 제공하지만, HM 는 무려 33개의 인트라 예측 인코딩 모드들을 제공할 수도 있다.

[0033] 일반적으로, HM 의 작업 모델은, 비디오 프레임 또는 화상이, 루마 및 크로마 샘플들 양자 모두를 포함하는 트리블록들 또는 최대 코딩 유닛 (LCU) 들의 시퀀스로 분할될 수도 있다는 것을 설명한다. 트리블록은 H.264 표준의 매크로블록과 유사한 목적을 갖는다. 슬라이스는, 코딩 순서에서 다수의 연속적인 트리블록들을 포함한다. 비디오 프레임 또는 화상은, 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각 트리블록은 쿼드트리에 따라 코딩 유닛 (CU) 들로 스플리팅될 수도 있다. 예를 들어, 쿼드트리의 루트 노드로서, 트리블록은 4개의 자식 노드 (child node) 들로 스플리팅될 수도 있고, 각 자식 노드는 차례로 부모 노드 (parent node) 가 될 수도 있고 또 다른 4개의 자식 노드들로 스플리팅될 수도 있다. 쿼드트리의 리프 노드로서, 최종, 스플리팅되지 않은 자식 노드는, 코딩 노드, 즉 코딩된 비디오 블록을 포함한다. 코딩된 비트스트림과 연관된 신택스 데이터는, 트리블록이 스플릿될 수도 있는 최대 횟수를 정의할 수도 있고, 또한 코딩 노드들의 최소 크기를 정의할 수도 있다.

[0034] CU 는 코딩 노드 그리고 그 코딩 노드와 연관된 예측 유닛 (PU) 들 및 변환 유닛 (TU) 들을 포함한다. CU 의 크기는 코딩 노드의 크기에 대응하고 형상이 정사각형이어야 한다. CU 의 크기는 8x8 화소들로부터, 최대 64x64 화소들 이상인 트리블록의 크기에 이르기까지의 범위일 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 신택스 데이터는, 예를 들어, CU 를 하나 이상의 PU 들로 파티셔닝하는 것을 기술할 수도 있다. 파티셔닝 모드들은, CU 가 스킵 또는 직접 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지 간에 달라질 수도 있다. PU 들은 형상이 비정사각형으로 파티셔닝될 수도 있다. CU 와 연관된 신택스 데이터는 또한, 예를 들어, 쿼드트리에 따라 CU 를 하나 이상의 TU 들로 파티셔닝하는 것을 기술할 수도 있다. TU 는 형상이 정사각형 또는 비정사각형일 수 있다.

[0035] HEVC 표준은, TU들에 따른 변환을 허용하고, 이는 상이한 CU들에 대해서 상이할 수도 있다. TU 들은 파티셔닝된 LCU에 대해 정의된 정해진 CU 내에 PU 들의 크기에 기초하여 사이징될 수 있지만, 이는 항상 그렇지는 않을 수도 있다. TU 들은 동일한 크기이거나 또는 PU 들보다 더 작을 수 있다. 일부 예들에서, CU 에 대응하는 잔차 샘플들은 "잔차 쿼드트리 (residual quad tree)" (RQT) 로 알려진, 쿼드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 변환 유닛 (TU) 들로 지칭될 수도 있다. TU 들과 연관된 화소 차이 값들이 변환되어 변환 계수들을 생성하고, 이들은 양자화될 수도 있다.

[0036] 일반적으로, PU 는 예측 프로세스에 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라 모드 인코딩될 때, PU 는 PU 를 위한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있다. 다른 실시예로서, PU 가 인터 모드 인코딩될 때, PU 는 PU 를 위한 움직임 벡터를 정의하는 데이터를 포함할 수도 있다. PU 를 위한

움직임 벡터를 정의하는 데이터는, 예를 들어, 움직임 벡터의 수평 성분, 움직임 벡터의 수직 성분, 움직임 벡터를 위한 해상도 (예를 들어, 1/4 화소 정밀도 또는 1/8 화소 정밀도), 움직임 벡터가 가리키는 참조 화상, 및/또는 움직임 벡터를 위한 참조 화상 리스트 (예를 들어, 리스트 0, 리스트 1, 또는 리스트 C) 를 기술할 수도 있다.

[0037] 일반적으로, TU 는 변환 및 양자화 프로세스들에 사용된다. 하나 이상의 PU 들을 갖는 정해진 CU 는 하나 이상의 변환 유닛 (TU) 들을 또한 포함할 수도 있다. 예측 다음에, 비디오 인코더 (20) 는 PU 에 대응하는 잔차 값들을 계산할 수도 있다. 잔차 값들은, 엔트로피 코딩을 위한 직렬화된 변환 계수들을 생성하기 위하여 TU들을 이용하여, 변환 계수들로 변환, 양자화 및 스캔될 수도 있는 화소 차이 값들을 포함한다. 본 명세서에서 사용된 "비디오 블록" 이라는 용어는 일반적으로 CU의 코딩 노드를 지칭할 수도 있다. 일부의 특정 경우들에서, 본 개시는 또한, 코딩 노드 및 PU 및 TU 들을 포함하는, 이를테면 CU 또는 LCU 인, 트리블록을 지칭하기 위해 용어 "비디오 블록" 을 사용할 수도 있다.

[0038] 비디오 시퀀스는 일련의 비디오 프레임들 또는 화상들을 포함할 수 있다. 화상들의 그룹 (GOP) 은 일반적으로, 일련의 하나 이상의 비디오 화상들을 포함한다. GOP 는 GOP 의 헤더, 하나 이상의 화상들의 헤더, 또는 다른 곳에 신택스 데이터를 포함할 수도 있고, 이 신택스 데이터는 GOP 에 포함된 화상들의 수를 기술한다. 화상의 각 슬라이스는, 각각의 슬라이스를 위한 인코딩 모드를 기술하는 슬라이스 신택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터를 인코딩하기 위하여 개개의 비디오 슬라이스들 내의 비디오 블록들에 대해 동작할 수 있다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정되거나 또는 변화하는 크기를 가질 수도 있고, 명시된 코딩 표준에 따라 크기가 다를 수도 있다.

[0039] 예로서, HM 은 다양한 PU 크기들에서의 예측을 지원한다. 특정 CU 의 크기가 2N×2N 이라고 가정하면, HM 는 2N×2N 또는 N×N 의 PU 크기들에서의 인트라 예측, 그리고 2N×2N, 2N×N, N×2N, 또는 N×N 의 대칭적 PU 크기들에서의 인터 예측을 지원한다. HM 은 또한, 2N×nU, 2N×nD, nL×2N, 및 nR×2N 의 PU 크기들에서의 인터 예측을 위한 비대칭적 파티셔닝을 지원한다. 비대칭적 파티셔닝에서, CU 의 일 방향은 파티셔닝되지 않는 반면, 타 방향은 25% 및 75% 으로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 부분은 “n” 다음에 “상 (Up)”, “하 (Down)”, “좌 (Left)”, 또는 “우 (Right)” 의 표시에 의해 표시된다. 따라서, 예를 들어, "2N×nU" 는, 상단의 2N×0.5N PU 및 하단의 2N×1.5N PU 로 수평적으로 파티셔닝되는 2N×2N CU 를 지칭한다.

[0040] 본 개시에서, “N×N” 그리고 “N 바이 N” 은, 수직 및 수평 크기 (dimension) 들의 면에서 비디오 블록의 화소 크기들, 예를 들면, 16×16 화소들 또는 16 바이 16 화소들을 지칭하는데 상호교환가능하게 사용될 수도 있다. 일반적으로,16×16 블록은, 수직 방향에서 16 개 화소들 (y = 16) 그리고 수평 방향에서 16 개 화소들 (x = 16) 을 가질 것이다. 마찬가지로, N×N 블록은 일반적으로 수직 방향에서 N 개 화소들 그리고 수평 방향에서 N 개 화소들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서 화소들은 행과 열들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향과 동일한 수의 수평 방향 화소들을 반드시 가질 필요는 없다. 예를 들면, 블록들은 N×M 화소들을 포함할 수도 있고, 여기서 M은 N과 반드시 동일한 것은 아니다.

[0041] CU 의 PU 들을 이용한 인트라 예측 또는 인터 예측 코딩 다음에, 비디오 인코더 (20) 는 CU 의 TU 들을 위한 잔차 데이터를 계산할 수도 있다. PU 들은, (화소 도메인으로도 지칭되는) 공간 도메인에서 화소 데이터를 포함할 수도 있고, TU들은, 변환, 예를 들어, 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환을, 잔차 비디오 데이터에 적용한 다음에 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는, PU 들에 대응하는 예측 값들 및 인코딩되지 않은 화상의 화소들간의 화소 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는, CU 를 위한 잔차 데이터를 포함한 TU 들을 형성할 수도 있고, 다음으로 그 TU 들을 변환하여 CU 를 위한 변환 계수들을 생성할 수도 있다.

[0042] 변환 계수들을 생성하기 위한 임의의 변환 다음에, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 일반적으로 양자화는, 변환 계수들이 양자화되어 그 계수들을 나타내는데 사용된 데이터의 양을 감소시킬 수 있으며, 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도 (bit depth) 를 감소시킬 수도 있다. 예를 들면, n-비트 값은 양자화 동안 m-비트 값으로 절사 (round down) 될 수도 있고, 여기서 n은 m보다 더 크다.

[0043] 일부 예들에서, 비디오 인코더 (20) 는, 미리정의된 스캔 순서를 이용하여 양자화된 변환 계수들을 스캔함으로써 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔을 수행할 수도 있다. 양자화된 변환 계수들을 스캔하여 1차원 벡터를 형성한 후에, 비디오 인

코더 (20) 는 1차원 벡터를, 예를 들어, CAVLC (context adaptive variable length coding), CABAC (context adaptive binary arithmetic coding), SBAC (syntax-based context-adaptive binary arithmetic coding), PIPE (Probability Interval Partitioning Entropy) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라, 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 인코딩된 비디오 데이터와 연관된 구문 요소들을 엔트로피 인코딩할 수도 있다.

[0044] CABAC 을 수행하기 위하여, 비디오 인코더 (20) 는, 송신될 심볼에 콘텍스트 모델 내의 콘텍스트를 배정할 수도 있다. 콘텍스트는, 예를 들어, 심볼의 이웃하는 값들이 비영 (non-zero) 인지 여부에 관한 것일 수도 있다. CAVLC 을 수행하기 위하여, 비디오 인코더 (20) 는, 송신될 심볼을 위해 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은, 상대적으로 더 짧은 코드들이 더 높은 확률 심볼들에 대응하는 한편, 더 긴 코드들이 더 적은 확률 심볼들에 대응하도록 구성될 수도 있다. 이런 식으로, VLC 의 사용은, 예를 들어, 송신될 각 심볼에 동일 길이 코드워드들을 이용하는 것에 비해, 비트 절약 (bit savings) 을 달성할 수도 있다. 확률 결정은, 심볼에 할당된 콘텍스트에 기초할 수 있다.

[0045] 도 2는 도 1의 비디오 인코더의 기능 블록도이다. 비디오 인코더 (20) 는 캡쳐된 비디오 데이터를 수신하고 비디오 데이터를 비디오 블록들로 파티셔닝하도록 구성된 파티셔닝 모듈 (35) 을 포함할 수 있다. 파티셔닝 모듈 (35) 은 비디오 데이터를 슬라이스, 타일 또는 다른 더 큰 유닛으로 파티셔닝할 수 있다. 파티셔닝 모듈 (35) 은 또한 비디오 블록들을, 예를 들어 LCU 및 CU의 쿼드트리 구조에 따라, 파티셔닝할 수 있다. 비디오 인코더 (20) 는 슬라이스를, 다수의 비디오 블록들로 (그리고 가능하게는 타일들로 지칭되는 비디오 블록들의 세트들로) 분할할 수도 있다.

[0046] 비디오 인코더 (20) 는 또한 파티셔닝 모듈 (35) 에 동작 가능하게 연결되고 파티셔닝된 비디오 데이터를 수신하도록 구성된 예측 모듈 (41) 을 가질 수 있다. 예측 모듈 (41) 은, 복수의 가능한 코딩 모드들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나를, 에러 결과 (예를 들어, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 현재 비디오 블록을 위해, 선택할 수 있다. 예측 모듈 (41) 은 비디오 인코더 (20) 로 하여금 비디오 슬라이스 내의 비디오 블록의 인트라 코딩 및 인터 코딩을 수행하게 할 수 있다. 인트라 코딩은, 정해진 비디오 프레임 또는 화상 내의 비디오에서 공간적 중복성을 감소 또는 제거하기 위하여 공간적 예측에 의거한다. 인터 코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내의 비디오에서 시간적 중복성을 감소 또는 제거하기 위하여 시간적 예측에 의거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다. 인터 모드들, 이를테면 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 은, 여러 시간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다.

[0047] 예측 모듈 (41) 은 움직임 추정 모듈 (42), 하나 이상의 보간 필터들 (43), 움직임 보상 모듈 (44) 및 인트라 예측 모듈 (46) 을 포함할 수 있다. 이들 컴포넌트들의 각각은 비디오 인코더 (20) 내의 하나 이상의 프로세서에 동작 가능하게 접속되거나 또는 통합될 수 있다. 개념적인 목적을 위하여 따로 논의되었지만, 움직임 추정 모듈 (42), 보간 필터 (43), 움직임 보상 모듈 (44) 및 인트라 예측 모듈 (46) 은 고도로 통합될 수도 있다.

[0048] 움직임 추정 모듈 (42) 은, 비디오 시퀀스에 대해 미리결정된 패턴에 따라 비디오 슬라이스를 위한 인터 예측 모드를 결정하도록 구성될 수 있다. 하나 이상의 참조 화상에서 하나 이상의 예측 블록에 대한 현재 비디오 블록의 그러한 인터 예측 또는 인터 예측 코딩은 시간적 압축을 제공할 수 있다. 미리결정된 패턴은, P 슬라이스들, B 슬라이스들 또는 GPB 슬라이스들로서 시퀀스에서 비디오 슬라이스들을 표기할 수도 있다. 본 명세서에서 사용된 움직임 추정은 일반적으로, 비디오 블록들을 위한 움직임을 추정하는 움직임 벡터들을 생성하는 프로세스를 지칭할 수도 있다. 예를 들어, 움직임 벡터는, 참조 프레임 또는 참조 화상 내의 예측 블록에 대해 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. 참조 프레임들은 예측 모듈 (41) 에 동작 가능하게 연결된 참조 화상 메모리 (64) 에 저장될 수 있다.

[0049] 일부 실시형태들에서, 움직임 벡터는 정수 미만 화소 ("부분 화소") 정밀도를 가질 수 있다. 예를 들어, 움직임 벡터의 수평 성분 및 수직 성분 양자 모두는 각각의 전체 정수, 또는 전체 화소, 성분 및 부분 화소 성분을 가질 수도 있다. 그러한 부분 화소 정밀도는 참조 프레임 내의 전체 화소 위치로부터 보간될 수 있다. 이와 같이, 분수 화소 (pixel), 분수 화소 (pel), 정수 미만 및 부분 화소 움직임 추정이라는 용어는 상호교환가능하게 사용될 수도 있다. 보간 필터 (43) 는 보간된 위치를 제공할 수 있다. 보간 필터 (43) 는 움직임 추정 및/또는 움직임 보상의 일부로서 보간을 수행하기 위해 움직임 추정 모듈 (42) 및 움직임 보상 모듈 (44) 중 하나 또는 양자 모두에 의해 호출될 수 있다. 이것은 아래에서 도 4a 및 도 4b와 관련하여 보다

상세히 설명된다.

[0050] 예측 블록은, 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이에 관하여, 코딩될 비디오 블록의 PU에 밀접하게 매칭하는 것으로 구해진 블록이다. 일부 예들에서, 움직임 예측 모듈 ("예측 모듈") (41) 은 참조 화상 메모리 (64) 에 저장된 참조 화상들의 정수 미만 화소 위치 (sub-integer pixel position) 들을 위한 값들을 계산할 수 있다. 예를 들어, 예측 모듈 (41) 은 참조 화상의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 화소 위치들의 값들을 보간할 수 있다. 그러므로, 움직임 추정 모듈 (42) 은, 전체 (full) 화소 위치들 그리고 분수 화소 위치들에 대해 움직임 탐색을 수행하고 분수 화소 정밀도로 움직임 벡터를 출력할 수도 있다.

[0051] 움직임 추정 모듈 (42) 은, PU 의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서 비디오 블록의 PU를 위한 움직임 벡터를 계산할 수 있다. 참조 화상은, 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1) 로부터 선택될 수 있고, 이들의 각각은 참조 화상 메모리 (64) 에 저장된 하나 이상의 참조 화상들을 식별한다. 움직임 추정 모듈 (42) 은 계산된 움직임 벡터를 움직임 보상 모듈 (44) 및 엔트로피 인코딩 모듈 (56) 에 전송할 수 있다.

[0052] 움직임 보상은 움직임 추정에 의해 결정된 움직임 벡터에 기초한 예측 블록의 페칭 (fetching) 또는 생성을 수반할 수 있다. 이것은 부분 화소 정밀도에 대한 보간 (예를 들어, 보간 필터들 (43)) 을 포함할 수 있다. 현재 비디오 블록의 PU를 위한 움직임 벡터의 수신시에, 움직임 보상 모듈 (44) 은, 움직임 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 로케이팅할 수 있다. 움직임 보상 모듈 (44) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 구문 요소들을 생성할 수 있다.

[0053] 예측 모듈 (41) 내의 인트라 예측 모듈 (46) 은 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대해 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수 있다. 인트라 예측 모듈 (46) 은, 상술된 바와 같이, 움직임 추정 모듈 (42) 및 움직임 보상 모듈 (44) 에 의해 수행되는 인터 예측에 대한 대안으로서, 현재 블록을 인트라 예측할 수 있다. 특히, 인트라 예측 모듈 (46) 은 현재 블록을 인코딩하는데 이용할 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 모듈 (46) 은, 예를 들어, 별개의 인코딩 패스들 동안, 다양한 인트라 예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있다. 예를 들어, 인트라-예측 모듈 (46) 은 다양한 테스트된 인트라-예측 모드들에 대한 레이트 왜곡 분석을 이용하여 레이트 왜곡 값들을 산출하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은, 인코딩된 블록과, 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양 뿐만 아니라 인코딩된 블록을 생성하는데 이용된 비트 레이트 (즉, 비트들의 수) 를 결정할 수 있다. 인트라 예측 모듈 (46) 은 여러 인코딩된 블록들에 대한 레이트 및 왜곡들로부터 비 (ratio) 를 산출하여 어느 인트라 예측 모드가 블록에 대한 최상의 레이트 왜곡 값을 나타내는지를 결정할 수도 있다.

[0054] 인트라 예측 모듈 (46) 은 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 모듈 (56) 에 제공할 수 있다. 엔트로피 인코딩 모듈 (56) 은 선택된 인트라 예측 모드를 표시하는 정보를 인코딩할 수 있다. 비디오 인코더 (20) 는 송신된 비트스트림에서 구성 데이터를 포함할 수 있고, 이는 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (코드워드 맵핑 테이블이라고도 한다), 다양한 블록들을 위한 인코딩 콘텍스트들의 정의들, 그리고 콘텍스트들의 각각을 위해 사용할 최고 확률 인트라 예측 모드, 인트라 예측 모드 인덱스 테이블, 및 수정된 인트라 예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0055] 예측 모듈 (41) 이 인터 예측 또는 인트라 예측 중 어느 하나를 통해 현재 비디오 블록에 대한 예측 블록을 생성한 후에, 비디오 인코더 (20) 는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 이 프로세스는 화소 차이 값을 초래할 수 있다. 화소 차이 값들은, 블록을 위한 잔차 데이터를 형성하고, 루마 및 크로마 차이 성분들 양자 모두를 포함할 수 있다. 예측 모듈 (41) 및 분할 모듈 (35) 에 동작 가능하게 연결된 합산기 (50) 는 이 감산 연산을 수행하도록 구성될 수 있는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0056] 잔차 블록에서 잔차 비디오 데이터는 하나 이상의 TU 들에 포함될 수도 있고 합산기 (50) 에 동작가능하게 연결된 변환 모듈 (50) 에 적용될 수도 있다. 변환 모듈 (52) 은 DCT (discrete cosine transform) 또는 개념

적으로 유사한 변환과 같은 변환을 이용하여 잔차 변환 계수들로 잔차 비디오 데이터를 변환할 수 있다. 변환 모듈 (52) 은 잔차 비디오 데이터를 화소 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 변환할 수 있다. 예측 모듈 (41) 은 결과적인 인트라 또는 인터 코딩된 블록을 합산기 (50) 에 제공하여 잔차 블록 데이터를 생성할 수 있다. 결과적인 인트라 또는 인터 코딩된 블록은 또한, 참조 화상으로서 사용을 위한 인코딩된 블록을 재구성하기 위해 합산기 (62) 에 제공될 수 있다.

[0057] 비디오 인코더 (20) 는 또한 합산기 (62) 에 동작 가능하게 연결된 필터 모듈 (63) 을 포함할 수 있다. 필터 모듈 (63) 은 디블록킹 필터, ALF (adaptive loop filter) 및 SAO (sample adaptive offset) 필터 등의 하나 이상의 루프 필터들을 나타낼 수 있다. 필터 모듈 (63) 이 인루프 필터인 것으로서 도 2에 도시되어 있지만, 다른 구성들에서, 필터 모듈 (63) 은 포스트 루프 필터로서 구현될 수 있다. 필터 모듈 (63) 은 참조 화상을 참조 화상 메모리 (64) 에 제공할 수 있다.

[0058] 변환 모듈 (52) 은 양자화 유닛 (54) 에 결과적인 변환 계수들을 전송할 수 있다. 양자화 모듈 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킬 수 있다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도 (bit depth) 를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 다음으로, 일부 예들에서, 양자화 모듈 (54) 은 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수 있다. 일부 실시 형태들에서, 엔트로피 인코딩 모듈 (56) 은 스캔을 수행할 수도 있다.

[0059] 양자화 다음에, 엔트로피 인코딩 모듈 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩할 수 있다. 예를 들어, 엔트로피 인코딩 모듈 (56) 은 CAVLC, CABAC, SBAC, 및/또는 PIPE, 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 방법을 수행할 수도 있다. 엔트로피 인코딩 모듈 (56) 에 의한 엔트로피 인코딩 후에, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신될 수 있다. 비트스트림은 또한 비디오 디코더 (30) 에 의해 나중에 송신 또는 취출을 위해 보관될 수 있다. 엔트로피 인코딩된 모듈 (56) 은 또한, 움직임 벡터들 그리고 코딩되고 있는 현재 비디오 슬라이스를 위한 다른 구문 요소들을 엔트로피 인코딩할 수 있다.

[0060] 비디오 인코더 (20) 는 또한 엔트로피 인코딩 모듈 (56) 에 동작 가능하게 연결된 역 양자화 모듈 (58) 을 포함할 수 있다. 역 변환 모듈 (60) 은 또한 역 양자화 모듈 (58) 및 합산기 (62) 에 동작 가능하게 연결될 수 있다. 역 양자화 모듈 (58) 및 역 변환 모듈 (60) 은 역 양자화 및 역 변환을 각각 적용하여, 참조 화상의 참조 블록으로서 나중에 사용하기 위해 화소 도메인에서 잔차 블록을 재구성할 수 있다. 움직임 보상 모듈 (44) 은, 참조 화상 리스트들 중의 하나 내의 참조 화상들 중의 하나의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수 있다. 움직임 보상 모듈 (44) 은 또한 하나 이상의 보간 필터들 (43) 중 하나 이상을 그 재구성된 잔차 블록에 적용하여, 움직임 추정에 사용하기 위한 정수 미만 화소 값들을 산출할 수 있다.
합산기 (62) 는 재구성된 잔차 블록을 움직임 보상 모듈 (44) 에 의해 생성되는 움직임 보상된 예측 블록에 가산하여 참조 블록을 생성해 참조 화상 메모리 (64) 에 저장할 수 있다. 참조 블록은 움직임 추정 모듈 (42) 및 움직임 보상 모듈 (44) 에 의해 참조 블록으로서 사용되어 후속 비디오 프레임 또는 화상에서 블록을 인터 예측할 수 있다.

[0061] 도 3은 도 1의 비디오 디코더의 기능 블록도이다. 비디오 디코더 (30) 는 엔트로피 디코딩 모듈 (80), 예측 모듈 (81), 역 양자화 모듈 (86), 역 변환 모듈 (88), 합산기 (90), 필터 모듈 (91), 및 참조 화상 메모리 (92) 를 포함할 수 있다. 예측 모듈 (81) 은, 움직임 보상 모듈 (82) 및 인트라 예측 모듈 (84) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 2 의 비디오 인코더 (20) 에 대해 설명된 인코딩에 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다.

[0062] 디코딩 프로세스 동안, 비디오 디코더 (30) 는, 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 구문 요소들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 는 네트워크 엔티티 (29) 로부터 인코딩된 비디오 비트스트림을 수신할 수도 있다. 네트워크 엔티티 (29) 는, 예를 들어, 서버, 비디오 에디터/스플라이서, 또는 전술된 방법들의 하나 이상을 구현하도록 구성된 다른 그러한 디바이스일 수도 있다. 전술된 바처럼, 본 개시에 설명된 방법들의 일부는, 네트워크 엔티티 (29) 가 인코딩된 비디오 비트스트림을 비디오 디코더 (30) 로 송신하기 전에 네트워크 엔티티 (29) 에 의해 구현될 수도 있다. 일부 비디오 디코딩 시스템들에서, 네트워크 엔티티 (29) 및 비디오 디코더 (30) 는 별개의 디바이스들의 부분들일 수도 있는 한편, 다른 사례들에서는, 네트워크 엔티티 (29) 에 대해 설명된 기능은, 비디오 디코더 (30) 를 포함하는 동일한 디바이스에 의해 수행될 수도 있다.

[0063] 비디오 디코더 (30) 의 엔트로피 디코딩 모듈 (80) 은 양자화된 계수들, 움직임 벡터들, 및 다른 구문 요소들을

생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 모듈 (80) 은 움직임 벡터들 및 다른 구문 요소들을 예측 모듈 (81) 에 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 구문 요소들을 수신할 수도 있다.

[0064] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩될 때, 예측 모듈 (81) 의 인트라 예측 모듈 (84) 은, 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터 시그널링된 인트라 예측 모드 및 데이터에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩될 때, 예측 모듈 (81) 의 움직임 보상 모듈 (82) 은 엔트로피 디코딩 모듈 (80) 로부터 수신된 움직임 벡터들 및 다른 구문 요소들에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 블록들을 생성한다. 예측 블록들은 참조 화상 리스트들 중의 하나의 리스트 내의 참조 화상들 중의 하나의 화상로부터 생성될 수도 있다. 비디오 디코더 (30) 는 참조 화상 메모리 (92) 내에 저장된 참조 화상들에 기초하여, 디폴트 구성 (default construction) 방법들을 이용하여 참조 프레임 리스트들, 리스트 0 및 리스트 1 을 구성할 수도 있다.

[0065] 움직임 보상 모듈 (82) 은, 움직임 벡터들 및 다른 구문 요소들을 파싱 (parsing) 하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록을 위한 예측 블록들을 생성한다. 예를 들어, 움직임 보상 모듈 (82) 은 수신된 구문 요소들의 일부를 사용하여 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예를 들어, 인트라 또는 인터 예측), 인터 예측 슬라이스 타입 (예를 들어, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스를 위한 참조 화상 리스트들의 하나 이상을 위한 구성 정보, 슬라이스의 각 인터 인코딩된 비디오 블록을 위한 움직임 벡터들, 슬라이스의 각 인터 코딩된 비디오 블록을 위한 인터 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0066] 움직임 보상 모듈 (82) 은 또한, 보간 필터들 (이 도면에 도시되지 않음) 에 기초하여 보간을 수행할 수도 있다. 일부 실시형태들에서, 비디오 디코더 (30) 내의 보간 필터는 보간 필터 (43) (도 2) 와 유사할 수 있다. 움직임 보상 모듈 (82) 은 보간 필터를 사용하여 정수 미만 화소 또는 특정 참조 블록 내의 부분 화소 위치에 대한 보간된 값을 계산할 수도 있다. 일부 실시형태들에서, 움직임 보상 모듈 (82) 은 수신된 구문 요소들을 통해 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정하고, 그 동일 또는 유사한 보간 필터들 (43) 을 이용하여 예측 블록들을 생성할 수 있다.

[0067] 역 양자화 모듈 (86) 은 비트스트림에서 제공되고 엔트로피 디코딩 모듈 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉 탈양자화한다. 역 양자화 프로세스는, 양자화의 정도, 그리고, 마찬가지로, 적용되어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서 각 비디오 블록을 위해 비디오 인코더 (20) 에 의해 산출된 양자화 파라미터의 이용을 포함할 수도 있다. 역 변환 모듈 (88) 은, 화소 도메인에서 잔차 블록들을 생성하기 위해 변환 계수들에, 역 변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용한다.

[0068] 움직임 보상 모듈 (82) 이 움직임 벡터들 및 다른 구문 요소들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역변환 모듈 (88) 로부터의 잔차 블록들을 움직임 보상 모듈 (82) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성할 수 있다. 합산기 (90) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낼 수 있다. 원하는 경우, (코딩 루프내 또는 코딩 루프 후의) 루프 필터들이 또한 화소 천이들을 매끄럽게 하거나 또는 다른 방법으로 비디오 품질을 향상시키는데 사용될 수도 있다. 필터 모듈 (91) 은 디블록킹 필터, ALF (adaptive loop filter) 및 SAO (sample adaptive offset) 필터 등의 하나 이상의 루프 필터들을 나타내도록 의도된다. 필터 모듈 (91) 이 인루프 필터인 것으로서 도시되어 있지만, 일부 실시형태들에서, 필터 모듈 (91) 은 포스트 루프 필터로서 구현될 수도 있다. 다음으로, 정해진 프레임 또는 화상에서 디코딩된 비디오 블록들은 참조 화상 메모리 (92) 에 저장되고, 이는 후속 움직임 보상을 위해 사용된 참조 화상들을 저장한다. 참조 화상 메모리 (92) 는 또한, 도 3 의 디스플레이 디바이스 (32) 등의 디스플레이 디바이스 상에 나중에 제시하기 위해 디코딩된 비디오를 저장한다.

[0069] 도 4a 는 움직임 검색을 위한 참조 프레임의 그래픽 표현이다. 참조 프레임 (400) 은 비디오 데이터의 8 × 8 화소 블록으로서 나타나 있다. 특정 실시형태들에서, 참조 프레임 (400) 의 크기는 8 × 8 보다 클 수 있다. 따라서, 그 크기는 설명의 목적을 위한 것이며 제한적인 것으로 고려되어서는 안된다. 따라서, 참조 프레임 (400) 은 64개 화소 (8 바이 8) 을 가질 수 있다. 개개의 화소들 각각은 도시된 바와 같이 4 ×

4 부분 화소로 더 분할될 수 있다. 참조 프레임 (400) 내의 정사각형들 각각은 전체 화소 또는 부분 화소 위치를 나타낸다. 참조 프레임 (400) 은 회색 색상의 대문자 "A-H" 및 "A1-H1" 를 갖는 전체 화소 위치를 나타낸다. 부분 화소 위치는 소문자 "a-o" 로 표시된다. 옅은 회색 색상의 부분 화소 위치 b, h 및 j은 1/2 화소 위치를 나타낸다. 모두 흰색 색상의 부분 화소 위치 a, c, d, e, f, g, i, k, l, m, n 및 o 는 1/4 화소 (Q 화소) 위치를 나타낸다. 참조 프레임 (400) 내의 위치들 중 일부만이 라벨링되어 있지만, 전체 화소 위치에서 부분 화소 위치들까지의 라벨링 관례 및 관계는 참조 프레임 (400) 전체에 걸쳐 반복될 수 있음을 이해해야 한다.

[0070] 이하의 설명은 비디오 인코더 (20) 에서의 예측 모듈 (41) 및 움직임 추정 모듈 (42) (도 2) 의 측면에서 제공된다. 그러나, 일부 실시 형태들에서, 이하의 설명은 비디오 디코더 (30) (도 3) 에서의 예측 모듈 (81) 및 움직임 보상 모듈 (82) 에도 적용될 수 있다.

[0071] 움직임 추정 모듈 (42) 은 참조 프레임 (400) 의 전체 화소 및 부분 화소 위치에 관하여 현재 프레임 (도 4b와 관련하여 후술됨) 에 대한 움직임 검색을 수행할 수 있다. 움직임 추정 모듈 (42) 은 움직임 검색의 해상도 및 정밀도에 따라, 전체 화소 (예를 들어, 블록 A-H 및 A1-H1) 및 부분 화소 위치 (a-0) 를 참조로서 사용하여 분수 화소 정밀도로 움직임 벡터를 출력할 수 있다. 분수 화소 움직임 추정에서, 움직임 추정 모듈 (42) 은 전체 화소 위치외의 위치로의 변위를 나타내는 움직임 벡터를 계산할 수 있다. 따라서, 움직임 벡터는 분수 화소 정밀도, 예를 들어, 1/2 화소 정밀도, 1/4 화소 정밀도, 1/8 화소 정밀도 또는 다른 분수 화소 정밀도를 가질 수도 있다. 따라서, 분수 화소 움직임 추정은 예측 모듈 (41) 이 정수 화소 (또는 전체 화소) 위치보다 더 높은 정밀도로 움직임을 추정할 수 있도록 허용하고, 따라서, 예측 모듈 (41) 은 보다 정확한 예측 블록을 생성할 수 있다.

[0072] 부분 화소 움직임 검색은 움직임 추정을 위한 증가된 해상도 및 바람직한 압축 비를 달성하는데 중요할 수 있다. 그러나, 움직임 추정 모듈 (42) 은 부분 화소 (예를 들어, 1/2 화소, 1/4 화소) 예측을 위해, 보간 필터 (43) 에 의한 보간 필터링을 위해 상당한 리소스를 요구할 수도 있으므로 계산 복잡도가 비교적 높을 수 있다. 계산 복잡성을 줄이기 위해, 정제 검색 (refinement search) 을 적용할 수 있다. 정제 검색 동안, 비디오 인코더 (20) 는 부분 화소 움직임 검색을 적용하여 전체 화소 움직임을 정제할 수 있다. 전체 화소 움직임은, 부분 화소 위치를 보간하지 않고서 참조 프레임 (400) 상의 움직임 검색 알고리즘 (예를 들어, 참조 화상 메모리 (64) 에 저장됨) 을 통해 획득될 수 있다. 일단 전체 화소 위치가 획득되면, 신호 및 움직임 예측을 정제하기 위해 이웃하는 부분 화소 위치들이 해결 (resolve) 될 수 있다. 일부 실시 형태들에서, 보간 필터 (43) 는 적어도 2개의 정제 단계 (stage) 들을 적용하도록 구성될 수 있다. 제 1 단계는 전체 화소 위치를 중심으로 한 1/2 화소 보간일 수 있고, 제 2 단계는 1/2 화소 위치를 중심으로 한 Q 화소 보간일 수 있다.

[0073] HEVC에서, 예를 들어, Q 화소 보간 프로세스는 2개의 1-D 보간 프로세스 (예를 들어, 수평 및 수직) 으로 구현될 수 있다. 이것은, 예를 들어 프레임 (400) 의 Q 화소 위치들 a, c, d, e, f, g, i, k, l, m, n 및 o를 생성한다. 입력이 전체 화소 위치의 8 비트 표현인 경우, 제 1 의 1-D 보간의 출력 (및 제 2 의 1-D 보간의 입력) 은 8 비트보다 클 수 있다 (예 : 16 비트). 따라서, 일부 실시 형태들에서, 메모리는 16 비트 입력의 저장을 수용할 수 있다. 일부 다른 실시 형태들에서, 저장은 더 클 수 있다. 정밀도가 2배가 될 수 있기 때문에, 제 1 보간이 프로세스에 대해 더 많은 샘플들을 필요로 할 수 있을지라도, 제 2 보간은 제 1 보간의 복잡도의 2배 (예를 들어, 8 비트 내지 16 비트의 입력 비트 정밀도) 일 수 있다. 예를 들어, 제 1 보간은 8 개의 더 많은 행 또는 열을 가질 수도 있다. 일부 실시형태에서, 보간 필터는 16 비트보다 작은 정밀도를 출력 할 수 있으며, 이 경우 저장은 또한 16 비트보다 작다. 일부 다른 실시형태들에서, 정밀도는 인위적으로 16 비트보다 작게 트렁케이트 (truncate) 될 수 있고 그 다음에 데이터를 패킹 (packing) 할 수 있다.

[0074] 일부 실시 형태에서, 부분 화소 위치 보간은 비디오 인코더 (20) 가 연산 요구사항을 감소시키기 위해 전체 참조 프레임 (400) 을 미리 계산하고 메모리 (예를 들어, 메모리 (34) 또는 참조 화상 메모리 (64)) 에 저장할 수 있게 한다. 이것은 현재 프레임에 있는 각 블록에 대한 모드 결정 및 움직임 검색 중에 각 부분 화소 위치가 여러 번 액세스될 수도 있으므로 유익할 수 있다. 그러나, 비디오 인코더 (20) 의 속도는 보간된 위치를 저장함으로써 메모리 사용 증가를 겪을 수 있다. 따라서, 일부 실시 형태들에서, 비디오 인코더 (20) 는 메모리 (34) 에 1/2 화소 (H-화소) 위치만을 저장할 수도 있다. 이는 더 높은 메모리 사용량 및 보다 낮은 계산 요구사항간에 절충을 제공할 수 있다. 저장된 1/2 화소 위치로부터 필요할 때 Q 화소 위치가 계산 (예를 들어, 보간) 될 수 있다.

[0075] 일부 실시형태에서, 움직임 추정 모듈 (42) 은 8 비트 입력을 갖는 움직임 솔루션에 대한 보간을 수행할 수 있다. 그 프로세싱에는 16 비트 또는 32 비트 솔루션의 사용이 포함될 수 있지만, 그 출력은 예를 들어 HEVC 에서와 같이 8 비트로 트렁케이트될 수 있다.

[0076] 일부 실시 형태에서, Q 화소 위치 (a, c, d 및 l) 를 얻는 가장 직접적인 또는 가장 덜 복잡한 방법은 (8개의 전체 화소의 화소 A-H 의 필터 지원을 갖는) 전체 화소 위치 D 에 수평 또는 수직 보간 필터 (43) 를 적용하는 것일 수 있다. 이것은 참조 프레임 (400) 내에서 1/2 화소 위치(들) b 를 도출할 수 있다. 움직임 추정 모듈 (42) 이 수평 보간 필터 (43) 로 보간 프로세스를 시작하면, 수직 보간 필터 (43) 가 나머지 1/2 화소 위치 h 및 j 그리고 (예를 들어, 1/2 화소 위치로부터의) Q 화소 위치를 도출하기 위해 뒤따를 수 있다. 예를 들어, 위치 e, i 및 m 은 Q 화소 위치 a 를 사용하여 수직 보간 필터로부터 도출될 수 있다. 또한, 위치 f 및 n 은 위치 b를 사용하여 수직 보간 필터 (43) 로부터 도출될 수 있고, 위치 g, k 및 o 는 위치 c를 사용하여 수직 보간 필터 (43) 로부터 도출될 수 있다. 위치 a 및 위치 c에서 도출된 임시 데이터는 수직 필터링을 적용하기 전에 저장 디바이스 (34) 내에 저장될 수 있다. 일부 실시 형태에서, 임시 데이터는 보간 필터 (43) 의 설계에 따라 8 비트 또는 16 비트로 저장될 수 있다. 일부 경우에, 위에서 언급한 것처럼, 정밀도가 저장에 의해 허용되는 것보다 크다.

[0077] 그러나, 보간 프로세스가 수직 보간 필터 (43) 로 시작되고 수평 보간 필터 (43) 가 뒤 따르는 경우, 위치 e, f 및 g 는 위치 d 로부터 도출될 수 있다; 위치 i 및 k는 위치 h 로부터 도출될 수 있다; 그리고 위치 m, n, o 는 위치 l 로부터 도출될 수 있다. 위치 d 및 위치 l 에서의 임시 데이터는 수평 필터링을 적용하기 전에 8 비트 또는 16 비트로 저장될 수 있다.

[0078] 일부 실시 형태에서, 서브샘플링된 에러 측정과 조합한 이러한 제 2 접근법 (예를 들어, 수직 및 그 다음에 수평 보간) 은 움직임 검색 및 결과적인 계산의 복잡성을 더 감소시킬 수 있다. 서브샘플링된 에러 측정에서, 참조 프레임 (400) 내의 데이터의 하나 걸러의 행만이 예측 에러를 측정하는데 사용된다. 따라서, 비디오 인코더 (20) 는 에러 측정에 대한 계산을 더 감소시킬 수 있고, 부분 화소 위치 정보를 취출 (retrieve) 하기 위해 메모리 (64) 에 반복적으로 액세스하기 위한 요구사항을 감소시킬 수 있다.

[0079] 도 4b 는 서브샘플 에러 측정을 이용한 움직임 추정의 그래픽 표현이다. 현재 프레임 (450) 은 움직임 추정을 위해 참조 프레임 (400) (도 4a) 을 이용하여 비디오 인코더 (20) 에 의해 인코딩될 수 있다. 참조 프레임 (400) 은 상술한 것과 유사하지만 부분 화소 위치 정보는 도시되지 않는다. 현재 프레임 (450) 의 인코딩이 진행됨에 따라, 움직임 추정 모듈 (42) 은 움직임 추정을 위해 기준 프레임 (400) 으로부터의 전체 화소 또는 부분 화소 위치를 사용할 수 있다.

[0080] 현재 프레임 (450) 은 참조 프레임 (400) 과 유사하게 비디오 데이터의 8 x 8 화소 블록으로서 도시된다. 현재 프레임 (450) 은 더 큰 비디오 프레임의 일부일 수 있으므로, 8 x 8 화소 묘사는 제한적인 것으로 고려되어서는 안된다. 현재 프레임 (450) 의 크기는 또한 예를 들어 HEVC에서 현재 프레임 (450) 을 인코딩하기 위해 필요에 따라 16x16, 32x32, 64x64 화소 또는 더 큰 것과 같은 더 큰 프레임일 수 있다. 현재 프레임 (450) 은 또한 필요에 따라 비대칭적일 수 있다. 일부 실시형태들에서, 현재 프레임 (450) 내의 블록들 각각은 전체 화소 위치 또는 부분 화소 위치를 나타낼 수 있다.

[0081] 일부 실시형태에서, 비디오 인코더 (20) 는 서브샘플링된 에러 측정을 사용하여 계산 복잡성을 감소시킬 수 있다. 서브샘플링된 에러 측정은 예측 에러를 측정하거나 또는 그렇지 않으면 결정하기 위해 참조 프레임 (400) 및 현재 프레임 (450) 내의 비디오 데이터의 하나 걸러의 행을 이용할 수 있다. 이러한 실시형태에서, 움직임 추정 모듈 (42) 은 우선 비디오 데이터에 수직 보간 필터 (43) 를 적용할 수 있다. 수평 보간 필터 (43) 가 하나 걸러의 수직 보간된 행 (예를 들어, 행 0, 행 2, 행 4 및 행 6) 에 대한 서브샘플링된 에러 측정을 사용하여 뒤따를 수 있다. 서브 샘플링된 에러 측정의 구현으로 인해, 참조 프레임 (400) 내의 비디오 데이터의 하나 걸러의 행만이 수평 보간 동안 움직임 검색을 위해 필요하며, 이는 계산 복잡성을 감소시킨다. 유사하게, 현재 프레임 (450) 의 하나 걸러의 행만 움직임 추정을 위해 사용될 수도 있다.

[0082] 일부 다른 실시형태에서, 움직임 추정 모듈 (42) 은 수직 보간 동안 하나 걸러의 행만을 고려하기 위한 (비디오 인코더 (20) 에서의) 추가 명령들을 구현할 수 있다. 수평 보간이 뒤따를 수 있다. 따라서, 수직 보간 필터 (43) 는 데이터의 하나 걸러의 행에 적용될 수 있고, 다음으로 수직 보간으로부터 도출된 위치들에 대해 수평 보간 필터 (43) 가 적용된다. 따라서, 수직 보간 필터링이 먼저 적용될 경우, 수평 보간 필터링 동안

연산이 절반 만큼 감소될 수 있다.

[0083] 일 예에서, 비디오 인코더 (20) 는, 여기서 화소 (452) 로도 지칭되는, 현재 프레임 (450) 의 상부 좌측 화소 "k" 에서 시작하여 화소 단위로 현재 프레임 (450) 을 인코딩할 수 있다. 움직임 추정 모듈 (42) 은 화소 (452) 에 가장 밀접하게 매칭되는 참조 프레임 (400) 에서 콜로케이트된 화소 (collocated pixel) 을 사용하여 화소 (452) 에 대한 움직임 벡터를 생성하기 위해 참조 프레임 (400) 을 사용할 수 있다. 도시된 바와 같이, 콜로케이트된 화소 (454) 은 행 2, 열 1에서 발견된다. 그 다음, 움직임 추정 모듈 (42) 은 가장 가깝게 매칭되는 블록 (454) 과 블록 (452) 의 위치 사이의 차이에 기초하여 움직임 벡터를 결정할 수 있다. 예시된 예에서, 움직임 벡터는 (-2, -1) 일 수 있다. 일부 실시형태에서, 콜로케이트된 화소 (454) 는 전체 화소 또는 부분 화소 위치일 수 있다.

[0084] 일부 실시형태에서, 움직임 추정 모듈 (42) 이 움직임 검색을 수행할 때, 비디오 인코더 (20) 는 서브샘플링된 에러 측정 때문에 참조 프레임 (400) 의 하나 걸러의 행을 보간하기만 하면된다. 보간된 화소 위치는 흰색 정사각형으로 표시되어 있다. 회색 정사각형은 보간되지 않은 위치를 나타내거나 또는 그렇지 않으면 움직임 검색의 목적을 위해 무시될 수도 있다. 다시 간략히 도 4a를 참조하면, 일부 실시형태에서, 이것은 행 (및 대응하는 부분 화소 위치) 1, 3, 5 및 7의 보간을 생략하는 것에 대응할 수 있다. 따라서, 비디오 인코더 (20) 는 도 4b 에 도시된 바와 같이 움직임 추정을 위해 행 0, 2, 4 및 6을 사용할 수 있다.

[0085] 일부 실시 형태들에서, 움직임 추정 모듈 (42) 은 움직임 검색을 위해 현재 프레임 (450) 내에서 하나 걸러의 행만을 고려할 수도 있다. 인코더가 움직임 검색을 위해 고려하는 행은 참조 프레임 (400) 과 유사하게 흰색 블록으로 도시되어 있다. 참조 프레임 (400) 에서 비디오 데이터의 하나 걸러의 행을 사용하기 때문에, 보간 및 관련 계산의 수는 감소된다. 이것은 코딩 효율과 계산 복잡도의 감소 사이에 소정의 절충을 낳을 수 있다.

[0086] 이하의 표 1은 X 바이 Y 의 크기를 갖는 PU 를 얻기 위해 얼마나 많은 8 비트 보간이 필요할 수도 있는지를 나타낸다. 단순화되지 않으면, 비디오 디코더 (30) 는 인코딩된 데이터에 직접 작용할 수 있다. 복잡성은 16 비트 보간 또는 2개의 8 비트 보간에 의해 지배될 수 있다. 예를 들어, 가장 작은 PU (예 : X = 8 및 Y = 4) 의 경우 인코더 또는 디코더에 대한 행 (1) 및 (2) 에서 e, g, m 및 o 위치를 해결하기 위한 계산은 (8+7)*4 +8*4*2 = 124 이다. 예를 들어, 행 (3) 에 도시된 바와 같이, 수평 보간 필터 전에 수직 보간 필터가 적용되고 수평 보간 동안 서브샘플링되면, 계산은 (8+7)*4+8*4/2 = 76 로 감소될 수 있다. 행 (3) 은 예측 모듈 (41) 에 대한 수직 서브샘플링 명령 없는 계산 복잡성을 나타낸다는 것에 유의해야 한다. 예를 들어, 모든 행에 대한 부분 화소 위치는 수직 보간 동안 계산될 수 있지만, 계산된 위치의 절반 (예를 들어, 하나 걸러의 행) 은 하나 걸러의 행에 수평 보간을 수행할 때 무시될 수도 있다.

		a, c	d, l	f, n	i, k	e, g	m, o
(1)	디코더	X*Y	X*Y	(X+7)*Y+X*Y*2	(X+7)*Y+X*Y*2	(X+7)*Y+X*Y*2	(X+7)*Y+X*Y*2
(2)	서브샘플 없는 인코더	X*Y	X*Y	X*Y	X*Y	(X+7)*Y+X*Y	(X+7)*Y+X*Y
(3)	서브샘플 있는 (수직 서브샘플 명령 없는) 인코더	X*Y/2	X*Y	X*Y	X*Y/2	(X+7)*Y+X*Y/2	(X+7)*Y+X*Y/2
(4)	서브샘플 있는 (수직 서브샘플 명령 있는) 인코더	X*Y/2	X*Y/2	X*Y/2	X*Y/2	((X+7)*Y+X*Y)/2	((X+7)*Y+X*Y)/2

[0087] 표 1 - 부분 화소 위치의 보간을 위한 계산 (1/2 화소 위치가 미리 계산되고 저장됨)

[0088] 또한, 예측 모듈 (41) 은 8개 Q 화소 이웃들에 대한 예측을 보다 잘 해결하기 위해 전체 화소 또는 부분 화소 위치들 중 하나에 중심을 둔 정제 검색을 적용할 수 있다. 따라서, 예측 모듈 (41) 은 제 1 의 1-D 보간을 저장 디바이스 (34) 에 저장할 수 있고 최대 3개의 추가 Q 화소 위치에 대해 저장 디바이스 (34) 를 참조할 수 있다. 예를 들어, 위치 e, f 및 g는 모두 위치 d 로부터 도출될 수 있다. 따라서, Q 화소 움직임 검색은 위치 j 에서 시작될 수 있고 계산은 (X + 7)*Y*3 + (X*Y*2*8) 로부터 (X + 7)*Y *2 + (X*Y/2)*8 로 감소된다.

[0089] 서브샘플 경우의 복잡성을 더 줄이기 위해, 검색이 위치 b 및 j에서 시작되면 Q 화소 위치 m, o, e 및 g 는 다르게 계산될 수 있다. 일부 실시형태에서, 위치 l 로부터 Q 화소 위치 m 및 o를 도출하고 위치 d 로부터 Q-화소 위치 e 및 g 을 도출하는 대신에, 위치 m 및 o 이 위치 n으로부터 도출될 수 있고, 위치 e 및 g 가 위치 f 로부터 도출될 수 있다. 결과적으로, 행 (3) 에 나타낸 바와 같이 (X+7)*Y*2 만큼 계산을 줄일 수 있다.

[0090] 일부 실시형태에서, 추가적인 수직 서브샘플 명령들이 포함될 수 있다. 이러한 수직 서브 샘플 명령들로, 예측 모듈 (41) 은 (행 (3) 에서와 같이 모든 행과는 반대로) 하나 걸러의 행에 수직 보간을 수행하고 행 (4) 에 보여진 바와 같이 계산을 더 감소시킬 수 있다. 이것은 예를 들어 위치 a, c, d, l, f, n (X*Y/2) 에 대해 보여진다. 따라서, 하나 걸러의 행에 대한 부분 화소 위치가 보간 (수직 보간) 되고, 다음으로 하나 걸러의 행에 대한 계산된 위치가 수평 보간 동안 사용될 수 있다.

[0091] 도 5는 움직임 검색 방법의 실시형태를 예시하는 플로우차트이다. 방법 (500) 은 인코딩될 참조 프레임 (400) 에 대한 1/2 화소 및 Q 화소 위치 정보를 도출하기 위해 부분 화소 움직임 검색을 위한 움직임 추정 모듈 (42) 에서 사용될 수 있다. 방법 (500) 은 예측 모듈 (41) 이 일련의 비디오 프레임에 관련된 전체 화소 위치 정보를 저장 디바이스 (34) 에 저장할 수 있을 때 단계 (505) 에서 시작할 수 있다. 전체 화소 위치 정보는 비디오 인코더 (20) 에 의해 수신된 비디오 데이터에 관련될 수 있고 참조 프레임 (400) 을 더 포함할 수 있다 (도 4a 및 도 4b).

[0092] 결정 단계 (510) 에서, 움직임 추정 모듈 (42) 은 검색된 위치 (1/2 화소, Q 화소) 가 단계 (505) 에서 저장된 전체 화소 위치를 갖는 참조 프레임 (400) 의 행 ("전체 화소 행") 상에 위치되는지 여부를 결정할 수 있다. 예를 들어, 도 4a의 참조 프레임 (400) 을 참조하면, 1/2 화소 위치 b 는 특히, 전체 화소 위치 D 및 E 를 갖는 전체 화소 행 상에 위치된다. 움직임 추정 모듈 (42) 이 위치 h, i, j 및 k 를 검색하는 경우, 예를 들어, 이들 1/2 화소 및 Q 화소 위치는 전체 화소 행 상에 있지 않다. 비디오 인코더 (20) 가 검색된 위치는

전체 화소 행 상에 없다고 결정하면, 방법 (500) 은 단계 (515) 로 진행된다. 단계 (515) 에서, 움직임 추정 모듈 (42) 은 전체 화소 위치 정보 (예를 들어, 위치 D, E, E1 및 F1) 에 수직 보간 필터 (43) 를 적용하여 제 1 부분 화소 위치 정보를 결정할 수 있다. 일부 실시형태에서, 움직임 추정 모듈 (42) 은 수직 보간 필터 (43) 를 전체 화소 위치 정보의 하나 걸러의 행에 추가로 적용할 수 있다 (예를 들어, 서브샘플 에러 측정). 이것은 상기 표 1에서 언급된 바와 같이 계산 복잡성을 감소시킬 수 있지만, 추가 명령 (예를 들어, 표 1의 행 (4)) 을 필요로 할 수도 있다. 부분 화소 위치 정보는 참조 프레임 (400) 또는 일련의 프레임에 관련된 비디오 정보의 하나 걸러의 라인 또는 행에 대한 제 1 의 1/4 화소 위치 정보 또는 제 1 의 1/2 화소 위치 정보일 수 있다.

[0093] 단계 (520) 에서, 움직임 추정 모듈 (42) 은 단계 (505) 에서 전체 화소 위치 정보와 동일한 정밀도로 메모리 (예를 들어, 저장 디바이스 (34)) 에 제 1 부분 화소 위치 정보를 저장할 수 있다.

[0094] 단계 (510) 로 돌아가서, 전체 화소 행에 대해 검색이 수행되는 경우 또는 단계 (520) 다음에, 방법 (500) 은 결정 단계 (525) 로 이동할 수 있다.

[0095] 결정 단계 (525) 에서, 움직임 추정 모듈 (42) 은 검색된 위치가 전체 화소 열에 있는지 여부를 결정할 수 있다. 예를 들어, 단계 (515) 에서 상기와 유사하게, 부분 화소 위치들 a, e, i, 및 m 은 부분 화소 위치들 d, h 및 l 과 동일한 방식으로 전체 화소 열에 있지 않다. 따라서, 단계 (530) 에서, 검색된 위치가 전체 화소 열에 있지 않으면, 움직임 추정 모듈 (42) 은 전체 화소 위치 정보, 제 1 부분 화소 위치 정보에 수평 보간 필터를 적용하여 제 2 부분 화소 위치 정보를 결정할 수 있다. 제 2 부분 화소 위치 정보는 비디오 정보의 하나 걸러의 라인에 대한 제 2 의 1/4 화소 위치 정보 또는 제 2 의 1/2 화소 위치 정보일 수 있다. 따라서, 움직임 추정 모듈 (42) 은 수평 보간이 비디오 데이터의 하나 걸러의 행에 대해 수행되기 때문에 서브샘플 에러 측정을 구현함으로써 계산을 1/2 만큼 감소시킬 수 있다. 수평 보간 중에 고려되지 않은 행들은 무시될 수 있다. 전술한 바와 같이, 수직 보간 필터 (43) 가 단계 (515) 에서 전체 화소 위치 정보의 하나 걸러의 행에 적용되면, 계산은 표 1의 행 (4) 에 언급된 바와 같이 더 감소된다.

[0096] 결정 단계 (525) 에서, 검색된 위치가 전체 화소 열 상에 있는 경우, 또는 단계 (530) 후에, 방법 (500) 은 단계 (535) 로 이동한다.

[0097] 단계 (535) 에서, 예측 모듈은 전체 화소 위치 정보 및/또는 제 1 및 제 2 부분 화소 위치 정보에 기초하여 일련의 프레임들의 블록/화소 움직임을 나타내는 구문 요소를 생성할 수 있다. 상술한 바와 같이, 부분 화소 위치 정보는 적어도 1/2 화소 또는 Q 화소 위치 정보일 수 있다.

[0098] 다음으로, 단계 (540) 에서, 비디오 인코더 (20) 는 단계 (535) 에서 생성된 구문 요소에 기초하여 비디오 데이터의 블록을 인코딩할 수 있다.

[0099] 여기에 개시된 정보 및 신호들은 임의의 다양한 상이한 기술 및 방법을 이용하여 표현될 수도 있다. 예를 들어, 위의 설명 전체에 걸쳐 언급될 수도 있는 데이터, 명령, 커맨드, 정보, 신호, 비트, 심볼, 및 칩은 전압, 전류, 전자기파, 자기장 또는 자기입자, 광학장 (optical field) 또는 광학 입자, 또는 이들의 임의의 조합에 의해 표현될 수도 있다.

[0100] 여기에 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리 블록, 모듈, 회로, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이 양자의 조합으로 구현될 수도 있다. 하드웨어와 소프트웨어의 이러한 상호교환가능성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트, 블록, 모듈, 회로, 및 단계가 일반적으로 그들의 기능성의 측면에서 위에서 설명되었다. 그러한 기능이 하드웨어 또는 소프트웨어로 구현될지 여부는, 전체 시스템에 부과된 특정 애플리케이션 및 설계 제약에 달려 있다. 당업자는 설명된 기능성을 각각의 특정 응용에 대해 다른 방식으로 구현할 수도 있지만, 이러한 구현 결정이 본 개시의 범위를 벗어나게 하는 것으로 해석되지 않아야 한다.

[0101] 여기에 기술된 실시형태들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수도 있다. 그러한 실시형태들은 범용 컴퓨터, 무선 통신 디바이스 핸드세트, 또는 무선 통신 디바이스 핸드세트 및 다른 디바이스들에서의 응용을 포함하는 다수의 사용들을 갖는 집적 회로 디바이스들과 같은 다양한 디바이스들 중의 임의의 것에서 구현될 수도 있다. 모듈들, 또는 컴포넌트들로서 설명된 임의의 특징들은 집적 로직 디바이스 (integrated logic device) 에서 함께 구현되거나 또는 이산이지만 연동적인 (interoperable) 로직 디바이스들로서 따로 구현될 수도 있다. 소프트웨어로 구현되면, 방법들은, 실행될 때, 위에서 설명된 방법들 중 하나 이상을 수행하는 명령들을 포함하는 프로그램 코드를 포함하는 컴퓨터 판독가능 데이터 저장 매체에 의해

적어도 부분적으로 실현될 수도 있다. 컴퓨터 판독가능 데이터 저장 매체는 패키징 재료들을 포함할 수도 있는 컴퓨터 프로그램 제품의 일부를 형성할 수도 있다. 컴퓨터 판독가능 매체는 메모리 또는 저장 저장 매체, 이를테면 RAM (random access memory) 이를테면, SDRAM (synchronous dynamic random access memory), ROM (read-only memory), NVRAM (non-volatile random access memory), EEPROM (electrically erasable programmable read-only memory), FLASH 메모리, 자기 또는 광학 데이터 저장 매체 등을 포함할 수도 있다. 추가로 또는 대안적으로 방법들은, 전파된 신호들 또는 파들과 같은, 명령들 또는 데이터 구조들의 형태의 프로그램 코드를 지니거나 통신하고, 컴퓨터에 의해 액세스, 판독 및/또는 실행될 수 있는 컴퓨터 판독가능 통신 매체에 의해 적어도 부분적으로 실현될 수도 있다.

[0102] 프로그램 코드는 프로세서에 의해 실행될 수도 있고, 프로세서는 하나 이상의 프로세서들, 이를테면 하나 이상의 DSP (digital signal processor), 범용 마이크로프로세서, ASIC (application specific integrated circuit), FPGA (field programmable logic array), 또는 다른 등가 집적 또는 이산 로직 회로를 포함할 수도 있다. 그러한 프로세서는 본 개시에 기재된 방법 및 프로세스들 중의 어느 것을 수행하도록 구성될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 다르게는, 프로세서는 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 또한, 프로세서는 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서, DSP 코어와 결합한 하나 이상의 마이크로프로세서, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 따라서, 본원에 사용된 용어 "프로세서" 는 임의의 전술한 구조, 전술한 구조의 임의의 조합, 또는 본원에 설명된 방법들의 구현에 적합한 임의의 다른 구조 또는 장치를 지칭할 수도 있다. 추가로, 몇몇 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성된 전용 소프트웨어 모듈 또는 하드웨어 모듈 내에 제공되거나 또는 결합된 비디오 인코더-디코더 (코덱) 에 포함될 수도 있다.

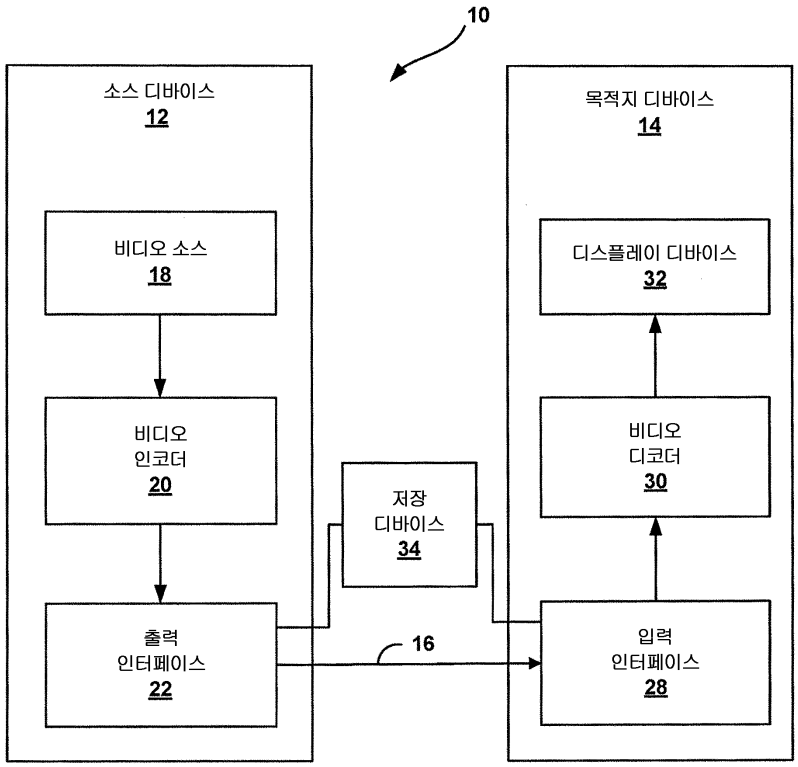
[0103] 본 발명의 실시 형태들은 특정 실시 형태에 대해 위에서 설명되었지만, 본 개시의 많은 변형이 가능하다. 예를 들어, 다양한 컴포넌트들의 수가 증가되거나 감소될 수도 있으며, 공급 전압을 결정하는 모듈들 및 단계들은 주파수, 또 다른 시스템 파라미터 또는 파라미터들의 조합을 결정하기 위하여 변경될 수도 있다. 추가적으로, 다양한 실시형태들의 특징들은 위에 설명된 것들과는 상이한 조합으로 조합될 수도 있다.

[0104] 당업자는 여기에 개시된 실시형태와 관련하여 설명된 다양한 예시적인 블록들 및 모듈들이 다양한 형태들에서 구현될 수 있다는 것을 인식할 것이다. 일부 블록들 및 모듈들은 일반적으로 그들의 기능의 측면에서 위에서 설명되었다. 그러한 기능이 어떻게 구현되는지는 전체 시스템에 부과된 설계 제약들에 의존한다. 당업자는 설명된 기능을 각각의 특정 애플리케이션에 대해 다른 방식으로 구현할 수 있지만, 이러한 구현 결정이 본 개시의 범위로부터 이탈을 야기하는 것으로 해석되지는 않아야 한다. 또한, 모듈, 블록 또는 단계 내에서 기능들의 그룹화는 설명의 용이를 위한 것이다. 특정 기능 또는 단계들은 본 개시로부터 벗어남이 없이 하나의 모듈 또는 블록으로부터 이동될 수 있거나 또는 모듈들 또는 블록들에 걸쳐 분산될 수 있다.

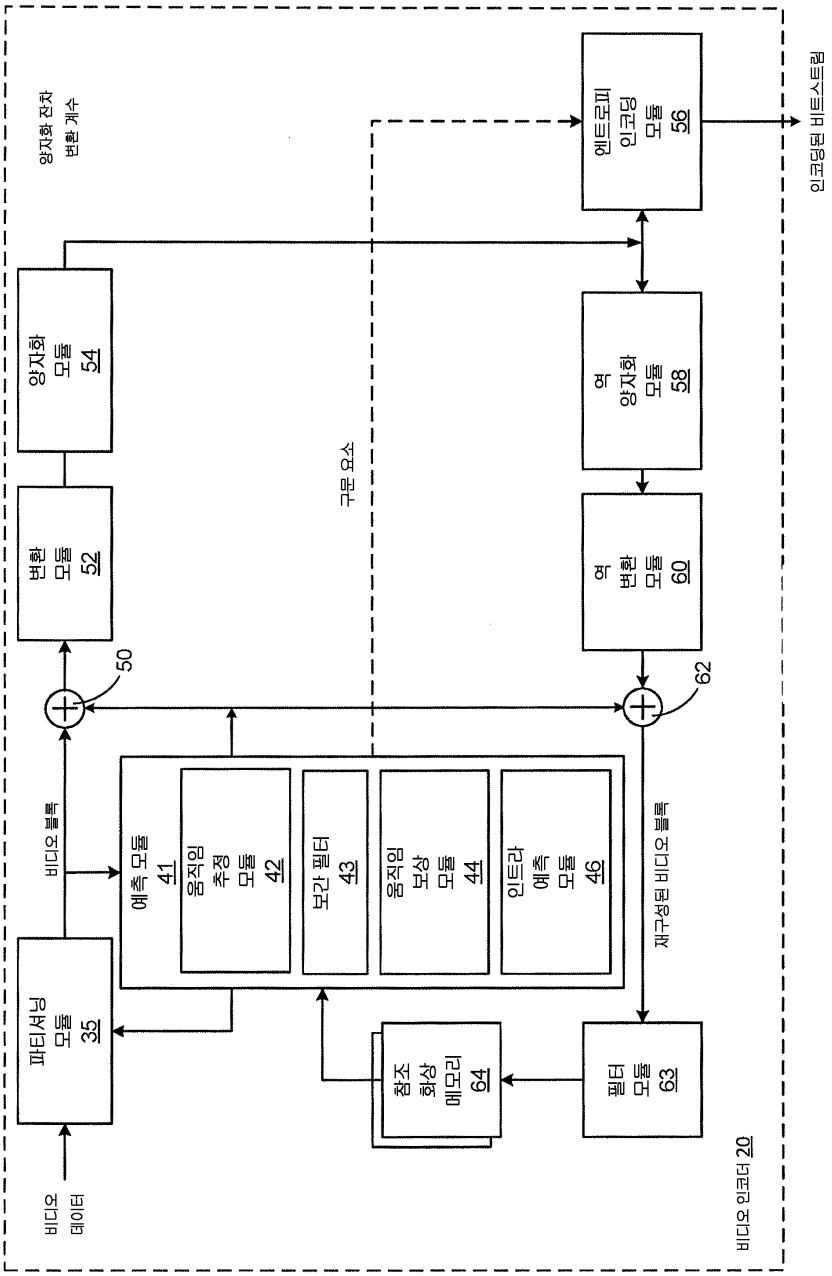
[0105] 개시된 실시형태의 위의 설명은 당업자가 본 개시를 제조 또는 사용하는 것을 가능하게 하기 위하여 제공된다. 이들 실시형태에 대한 다양한 변형은 당업자에게는 용이하게 명백할 것이며, 여기에 설명된 일반적인 원리는 본 개시의 사상 또는 범위를 벗어남이 없이 다른 실시형태에 적용될 수 있다. 따라서, 여기에 제시된 설명 및 도면들은 본 개시의 본 바람직한 구현들을 나타내고 따라서 본 개시에 의해 폭넓게 고려되는 요지를 나타낸다는 것이 이해되야 한다. 또한, 본 개시의 범위는 당업자에게 명백할 수도 있는 다른 실시형태를 완전히 포함하고 이에 따라 본 개시의 범위는 단지 첨부 청구항들에 의해만 제한된다는 것이 이해된다.

도면

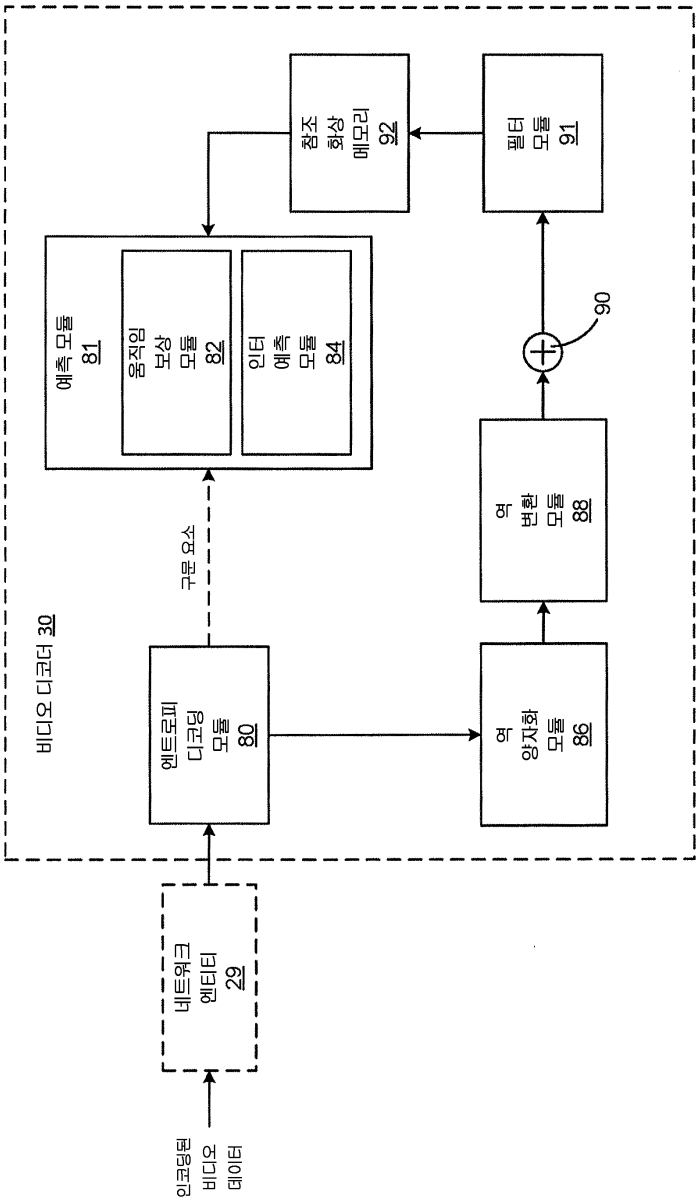
도면1


10
소스 디바이스
12
비디오 소스
18
비디오
인코더
20
출력
인터페이스
22
저장
디바이스
34
16
목적지 디바이스
14
디스플레이 디바이스
32
비디오
디코더
30
입력
인터페이스
28

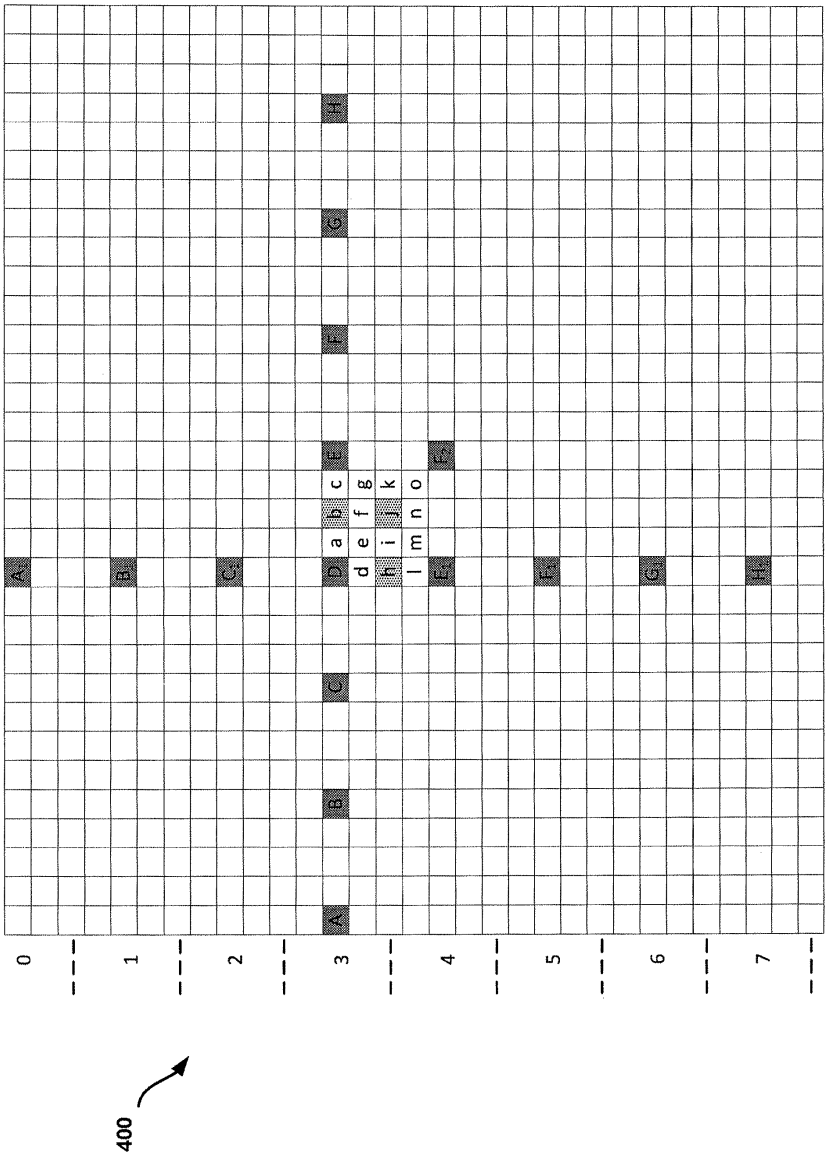
도면2


양자화 잔차
변환 계수
엔트로피
인코딩
모듈
56
인코딩된 비트스트림
양자화
모듈
54
역
양자화
모듈
58
구문 요소
변환
모듈
52
역
변환
모듈
60
50
62
비디오 블록
예측 모듈
41
움직임
추정
모듈
42
보간 필터
43
움직임
보상
모듈
44
인트라
예측
모듈
46
재구성된 비디오 블록
파티셔닝
모듈
35
참조
화상
메모리
64
필터
모듈
63
비디오
데이터
비디오 인코더 20

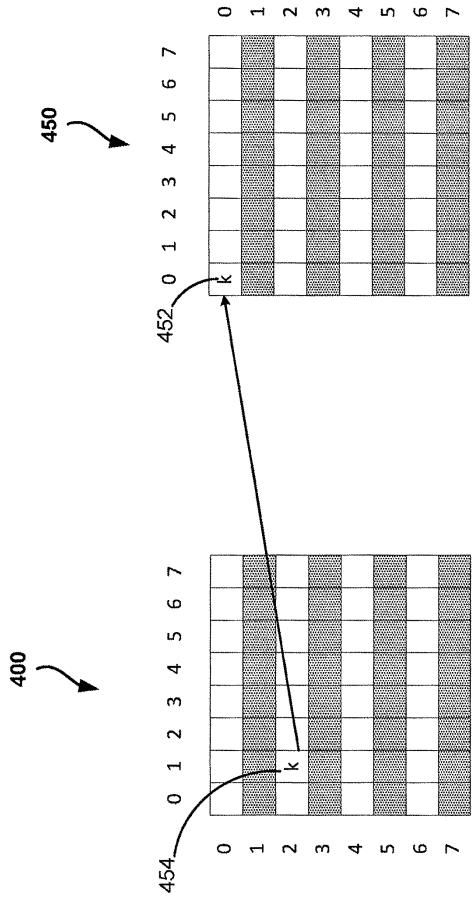
도면3


인코딩된 비디오 데이터
네트워크 엔티티 29
비디오 디코더 30
엔트로피 디코딩 모듈 80
구문 요소
예측 모듈 81
움직임 보상 모듈 82
인터 예측 모듈 84
역 양자화 모듈 86
역 변환 모듈 88
90
필터 모듈 91
참조 화상 메모리 92

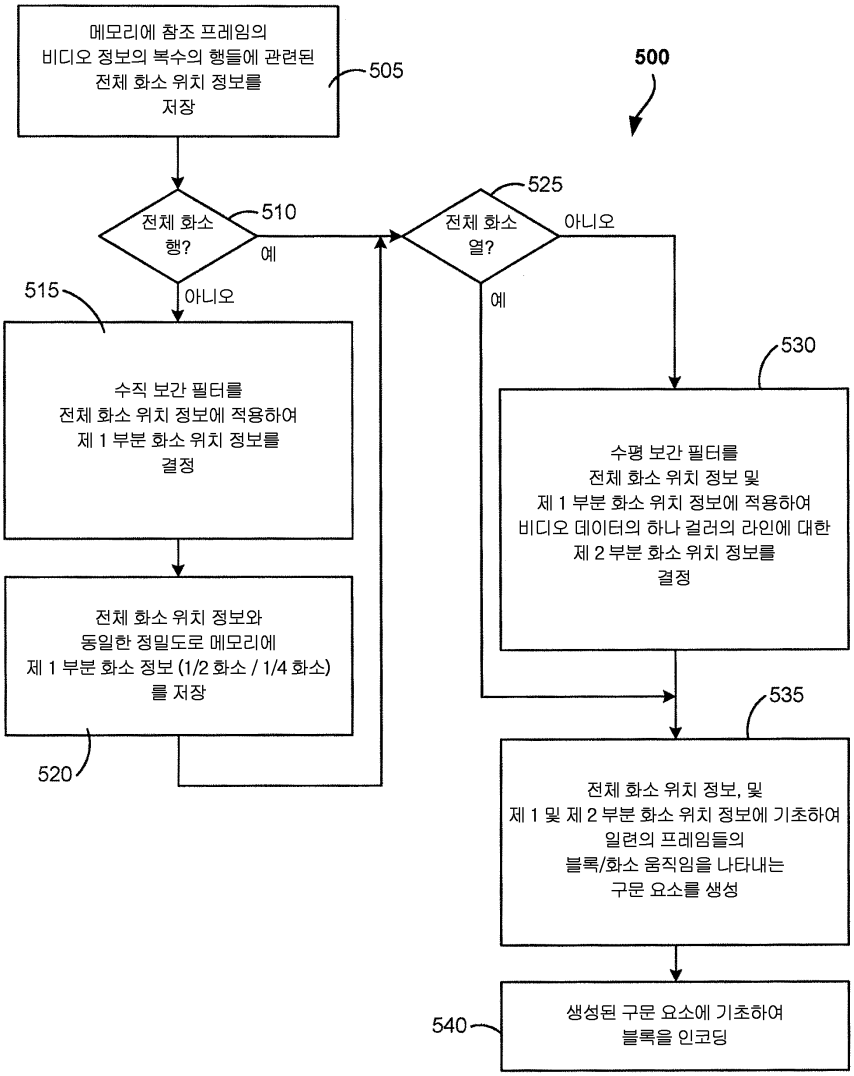
도면4a



도면4b



도면5


500
메모리에 참조 프레임의 비디오 정보의 복수의 행들에 관련된 전체 화소 위치 정보를 저장
505
전체 화소 행?
510
예
아니오
515
수직 보간 필터를 전체 화소 위치 정보에 적용하여 제 1 부분 화소 위치 정보를 결정
전체 화소 위치 정보와 동일한 정밀도로 메모리에 제 1 부분 화소 정보 (1/2 화소 / 1/4 화소)를 저장
520
525
전체 화소 열?
아니오
예
530
수평 보간 필터를 전체 화소 위치 정보 및 제 1 부분 화소 위치 정보에 적용하여 비디오 데이터의 하나 걸러의 라인에 대한 제 2 부분 화소 위치 정보를 결정
535
전체 화소 위치 정보, 및 제 1 및 제 2 부분 화소 위치 정보에 기초하여 일련의 프레임들의 블록/화소 움직임을 나타내는 구문 요소를 생성
540
생성된 구문 요소에 기초하여 블록을 인코딩