



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2021년09월30일

(11) 등록번호 10-2306845

(24) 등록일자 2021년09월23일

- (51) 국제특허분류(Int. Cl.)
H04N 19/523 (2014.01) **H04N 19/174** (2014.01)
H04N 19/43 (2014.01) **H04N 19/44** (2014.01)
H04N 19/513 (2014.01) **H04N 19/53** (2014.01)
H04N 19/70 (2014.01)
- (52) CPC특허분류
H04N 19/523 (2015.01)
H04N 19/174 (2015.01)
- (21) 출원번호 10-2016-7021329
- (22) 출원일자(국제) 2015년01월07일
 심사청구일자 2019년12월10일
- (85) 번역문제출일자 2016년08월03일
- (65) 공개번호 10-2016-0106127
- (43) 공개일자 2016년09월09일
- (86) 국제출원번호 PCT/US2015/010391
- (87) 국제공개번호 WO 2015/105816
 국제공개일자 2015년07월16일
- (30) 우선권주장
 61/925,108 2014년01월08일 미국(US)
 (뒷면에 계속)
- (56) 선행기술조사문헌
 JP2009171608 A*
 US20040001544 A1*
 WO2012042893 A1*
 *는 심사관에 의하여 인용된 문헌

- (73) 특허권자
 마이크로소프트 테크놀로지 라이선싱, 엘엘씨
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이
- (72) 발명자
 실킨 세르게이
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 엘씨에이-인터내셔널 패턴즈
 (8/1172) 마이크로소프트 코포레이션 내
- 사블린 세르게이
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 엘씨에이-인터내셔널 패턴즈
 (8/1172) 마이크로소프트 코포레이션 내
 (뒷면에 계속)
- (74) 대리인
 김태홍, 김진희

전체 청구항 수 : 총 18 항

심사관 : 박상철

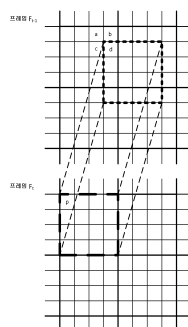
(54) 발명의 명칭 인코딩된 비트 스트림에서의 움직임 벡터 표현

(57) 요약

움직이는 이미지 데이터를 인코딩하는데 사용하기 위한 포맷은, 적어도 하나의 영역이 움직임 추정을 사용하여 인코딩되는 복수의 프레임을 포함하는 프레임 시퀀스; 상기 프레임 각각 또는 상기 프레임 각각 내의 하나 이상의 영역 각각에 대한 움직임 추정의 움직임 벡터를 나타내는 움직임 벡터 값들의 세트; 및 개개의 프레임 또는

(뒷면에 계속)

대표도 - 도7



영역의 개개의 움직임 벡터 값이 제 1 해상도 또는 제 2 해상도로 인코딩되는지를 나타내는, 상기 개개의 프레임 또는 영역 각각과 연관된 적어도 하나의 지표를 포함한다.

(52) CPC특허분류

H04N 19/43 (2015.01)
H04N 19/44 (2015.01)
H04N 19/513 (2015.01)
H04N 19/53 (2015.01)
H04N 19/70 (2015.01)

(30) 우선권주장

61/934,506 2014년01월31일 미국(US)
 14/530,625 2014년10월31일 미국(US)

(72) 발명자

주 유

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트
 프트 웨이 엘씨에이-인터내셔널 페턴츠 (8/1172)
 마이크로소프트 코포레이션 내

린 치룽

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트
 프트 웨이 엘씨에이-인터내셔널 페턴츠 (8/1172)
 마이크로소프트 코포레이션 내

리 밍치에

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트
 프트 웨이 엘씨에이-인터내셔널 페턴츠 (8/1172)
 마이크로소프트 코포레이션 내

설리번 개리 제이.

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트
 프트 웨이 엘씨에이-인터내셔널 페턴츠 (8/1172)
 마이크로소프트 코포레이션 내

명세서

청구범위

청구항 1

디코더에 있어서,

움직임 추정(motion estimation)을 사용하여 영역(region)이 인코딩되는 복수의 프레임들을 포함하는, 인코딩된 형태의 움직이는 이미지 데이터(moving image data)를 수신하는 입력부 - 상기 움직이는 이미지 데이터는 상기 프레임들 각각의 프레임과 연관되거나 또는 상기 프레임들 각각 내의 각각의 영역과 연관된 움직임 벡터 값들의 각각의 세트를 포함하고, 각각의 프레임 또는 영역의 움직임 벡터 값들 각각은 인코딩된 비트스트림 내의 움직임 벡터 필드 내에 포함되고, 상기 움직이는 이미지 데이터는, 상기 움직임 벡터 값들 각각이 제1 해상도로 인코딩되어 있는지 또는 제2 해상도로 인코딩되어 있는지 여부를 표시하는, 각각의 프레임들 또는 영역들 각각과 연관된 두 개의 각각의 지표(indicator)들을 헤더 정보 내에 더 포함하고, 상기 두 개의 각각의 지표들은 각각의 움직임 벡터들의 해상도를 2차원으로 표시하며, 상기 움직임 벡터 필드는, 상기 제1 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들의 크기에 비해, 상기 제2 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들에 대해 감소된 크기를 가지며, 상기 움직이는 이미지 데이터는 상기 움직임 벡터 값들의 제1 해상도 또는 제2 해상도가 영역에 대해(per region) 표시되는지 또는 프레임에 대해(per frame) 표시되는지 여부를 설정하기 위한 세팅을 포함함 -; 및

상기 움직임 벡터 값들에 기초하여 상기 프레임들 또는 영역들을 디코딩하는 움직임 예측 모듈

을 포함하고,

상기 디코딩은,

상기 헤더 정보로부터 상기 지표들 각각을 판독하는 것;

상기 세팅을 판독하고, 이에 따라 상기 움직임 벡터 값들을 해석하는 것;

상기 판독된 지표들로부터, 각각의 프레임 또는 영역의 상기 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 결정하는 것; 및

상기 움직임 벡터 값들을, 상기 제1 해상도로 표시된 경우 부화소(fractional pixel)들의 단위로, 또는 상기 제2 해상도로 표시된 경우에 응답하여 정수 화소(integer pixel)들의 단위로 해석하는 것

을 포함한 것인, 디코더.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 프레임들 중 적어도 하나는 복수의 영역들로 분할되고,

상기 움직이는 이미지 데이터는, 각각의 영역의 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 개별적으로 표시하기 위한, 상기 복수의 영역들 각각과 연관된 적어도 하나의 각각의 지표를 포함하며,

상기 움직임 예측 모듈은, 상기 지표들을 판독하여, 각각의 영역의 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 결정하고, 이에 따라 상기 움직임 벡터 값들 각각을 상기 부화소들의 단위 또는 정수 화소들의 단위로 해석하는 것인, 디코더.

청구항 4

제3항에 있어서,

상기 복수의 영역들은 H.26x 표준의 슬라이스들인 것인, 디코더.

청구항 5

제1항에 있어서,

상기 움직임 예측 모듈은, 상기 프레임들 또는 영역들 중 하나에 대해 각각의 지표들이 존재하지 않는 경우, 상기 움직임 벡터 값들 각각을 디폴트(default)로서 부화소들의 단위로 해석하는 것인, 디코더.

청구항 6

제1항에 있어서,

상기 움직임 벡터 값들 중 적어도 하나의 움직임 벡터 값은 델타 값의 측면에서 차등적으로(differentially) 인코딩되고, 상기 움직임 예측 모듈은 상기 움직임 벡터 값들 중 다른 움직임 벡터 값을 스케일링함으로써 상기 적어도 하나의 움직임 벡터 값을 예측하며,

상기 움직임 예측 모듈은, 상기 스케일링된 움직임 벡터 값을 정수 화소 해상도로 반올림(rounding)하지 않고서 상기 스케일링된 움직임 벡터 값에 대한 상기 델타 값의 가산을 수행함으로써, 상기 적어도 하나의 움직임 벡터 값을 정수 화소들의 단위로 해석하는 것인, 디코더.

청구항 7

제1항에 있어서,

상기 디코더는 상기 움직임 예측 모듈에 의한 상기 디코딩에 선행하는 무손실 인코딩 스테이지의 역(inverse)을 포함한 것인, 디코더.

청구항 8

컴퓨터 프로그램 제품을 포함하는 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스에 있어서, 상기 컴퓨터 프로그램 제품은 프로세서에 의한 실행에 응답하여 동작들을 수행하고, 상기 동작들은,

움직임 추정을 사용하여 영역이 인코딩되는 복수의 프레임들을 포함하는, 인코딩된 형태의 움직이는 이미지 데이터를 수신하는 동작 - 상기 움직이는 이미지 데이터는 상기 프레임들 각각의 프레임과 연관되거나 또는 상기 프레임들 각각 내의 각각의 영역과 연관된 움직임 벡터 값들의 각각의 세트를 포함하고, 각각의 프레임 또는 영역의 움직임 벡터 값들 각각은 인코딩된 비트스트림 내의 움직임 벡터 필드 내에 포함되고, 상기 움직이는 이미지 데이터는, 상기 움직임 벡터 값들 각각이 제1 해상도로 인코딩되어 있는지 또는 제2 해상도로 인코딩되어 있는지 여부를 표시하는, 각각의 프레임들 또는 영역들 각각과 연관된 두 개의 각각의 지표들을 헤더 정보 내에 더 포함하고, 상기 두 개의 각각의 지표들은 각각의 움직임 벡터들의 해상도를 2차원으로 표시하며, 상기 움직임 벡터 필드는, 상기 제1 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들의 크기에 비해, 상기 제2 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들에 대해 감소된 크기를 가지며, 상기 움직이는 이미지 데이터는 상기 움직임 벡터 값들의 상기 제1 해상도 또는 상기 제2 해상도가 영역에 대해 표시되는지 또는 프레임에 대해 표시되는지 여부를 설정하기 위한 세팅을 포함함 -;

상기 헤더 정보로부터 상기 지표들 각각을 판독하는 동작;

상기 세팅을 판독하고, 이에 따라 상기 움직임 벡터 값들을 해석하는 동작;

상기 판독된 지표들로부터, 각각의 프레임 또는 영역의 상기 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 결정하는 동작;

상기 움직임 벡터 값들 각각을, 상기 제1 해상도로 표시된 경우 부화소들의 단위로, 또는 상기 제2 해상도로 표시된 경우 정수 화소들의 단위로 해석하는 동작; 및

상기 해석에 따라 상기 부화소들의 단위 또는 상기 정수 화소들의 단위로 해석된 상기 움직임 벡터 값들에 기초하여 상기 프레임들 또는 영역들을 디코딩하는 동작

을 포함한 것인, 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스.

청구항 9

삭제

청구항 10

제8항에 있어서,

상기 프레임들 중 적어도 하나는 복수의 영역들로 분할되고,

상기 움직이는 이미지 데이터는, 각각의 영역의 상기 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 개별적으로 표시하기 위한, 상기 복수의 영역들 각각과 연관된 적어도 하나의 각각의 지표를 포함하며,

상기 컴퓨터 프로그램 제품은 또한, 상기 지표들을 판독하여, 각각의 영역의 상기 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 결정하고, 이에 따라 상기 움직임 벡터 값들 각각을 상기 부화소들의 단위 또는 정수 화소들의 단위로 해석하는 것을 포함하는 동작들을 수행하는 것인, 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스.

청구항 11

제10항에 있어서,

상기 움직이는 이미지 데이터는 상기 움직임 벡터 값들의 해상도가 영역에 대해 표시되는지 또는 프레임에 대해 표시되는지 여부를 설정하기 위한 세팅을 더 포함하며,

상기 컴퓨터 프로그램 제품은 또한, 상기 세팅을 판독하고, 이에 따라 상기 움직임 벡터 값들을 해석하는 것을 포함하는 동작들을 수행하는 것인, 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스.

청구항 12

제8항에 있어서,

상기 컴퓨터 프로그램 제품은 또한, 상기 프레임들 또는 영역들 중 하나에 대해 각각의 지표들이 존재하지 않는 경우, 상기 움직임 벡터 값들 각각을 디폴트로서 부화소들의 단위로 해석하는 것을 포함하는 동작들을 수행하는 것인, 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스.

청구항 13

네트워크 하드웨어 디바이스에 있어서,

프로세서에 의한 실행에 응답하여 인코딩된 비트스트림을 생성하는 프로그램이 저장되어 있는 컴퓨터 판독가능 저장 하드웨어 메모리 디바이스를 포함하고,

상기 인코딩된 비트스트림은,

움직임 추정을 사용하여 영역이 인코딩되는 복수의 프레임들을 포함하는 프레임들의 시퀀스;

상기 프레임들 각각 또는 상기 프레임들 각각 내의 각각의 영역에 대한 상기 움직임 추정의 움직임 벡터들을 나타내는 움직임 벡터 값들의 각각의 세트 - 각각의 프레임 또는 영역의 움직임 벡터 값들 각각은 상기 인코딩된 비트스트림 내의 움직임 벡터 필드 내에 포함되고, 상기 움직임 벡터 값들의 각각의 세트는, 제1 해상도로 인코딩된 움직임 벡터 값들이 부화소 단위들을 나타내고 제2 해상도로 인코딩된 움직임 벡터 값들이 정수 화소 단위들을 나타내도록 하는 프로토콜에 따라 인코딩되며, 상기 프로토콜에 따라, 상기 움직임 벡터 필드는, 상기 제1 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들의 크기에 비해, 상기 제2 해상도로 인코딩된 움직임 벡터 값들을 갖는 프레임들 또는 영역들에 대해 감소된 크기를 가짐 -; 및

상기 프레임들 또는 영역들 각각과 연관된, 헤더 정보 내의 두 개의 각각의 지표들 - 상기 두 개의 각각의 지표들은, 각각의 프레임 또는 영역의 상기 움직임 벡터 값들 각각이 영역에 대해 표시되거나 또는 프레임에 대해 표시된 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 표시하며, 각각의 움직임 벡터들의 해상도를 2차원으로 표시하고, 디코더로 하여금, 상기 지표를 판독할 수 있게 하고, 상기 지표를 상기 헤더 정보 내에 포함시킴으로써 상기 움직임 벡터 값들 각각을 부화소들의 단위 또는 정

수 화소들의 단위로 해석할 수 있게 함 -
을 포함한 것인 네트워크 하드웨어 디바이스.

청구항 14

제13항에 있어서,

상기 인코딩된 비트스트림은, 상기 복수의 프레임들 또는 영역들 중 상기 제1 해상도로 인코딩된 프레임들 또는 영역들, 및 상기 복수의 프레임들 또는 영역들 중 상기 제2 해상도로 인코딩된 다른 프레임들 또는 영역들을 포함하며,

상기 두 개의 각각의 지표들은 상기 복수의 프레임들 또는 영역들 각각에 대해 상기 해상도를 개별적으로 표시한 것인, 네트워크 하드웨어 디바이스.

청구항 15

제13항에 있어서,

상기 프레임들 중 적어도 하나는 복수의 영역들로 분할되며,

상기 인코딩된 비트스트림은, 각각의 영역의 상기 움직임 벡터 값들이 상기 제1 해상도로 인코딩되어 있는지 또는 상기 제2 해상도로 인코딩되어 있는지 여부를 개별적으로 표시하기 위한, 상기 복수의 영역들 각각과 연관된 적어도 하나의 각각의 지표를 포함한 것인, 네트워크 하드웨어 디바이스.

청구항 16

제15항에 있어서,

상기 영역들은 H.26x 표준의 슬라이스들인 것인, 네트워크 하드웨어 디바이스.

청구항 17

제15항에 있어서,

상기 인코딩된 비트스트림은 상기 움직임 벡터 값들의 해상도가 영역에 대해 표시되는지 또는 프레임에 대해 표시되는지 여부를 설정하기 위한 세팅을 더 포함한 것인, 네트워크 하드웨어 디바이스.

청구항 18

제13항에 있어서,

상기 인코딩된 비트스트림 내에 포함된 상기 움직임 벡터 값들 각각은 또한 무손실 인코딩 기술에 따라 인코딩된 것인, 네트워크 하드웨어 디바이스.

청구항 19

제13항에 있어서,

상기 프레임들 또는 영역들 중 하나에 대해 각각의 지표들이 존재하지 않는 경우, 상기 움직임 벡터 값들 각각이 디폴트로서 부화소들의 단위로 해석되는 것인, 네트워크 하드웨어 디바이스.

청구항 20

제13항에 있어서,

상기 인코딩된 비트스트림은 상기 움직임 벡터 값들의 해상도가 영역에 대해 표시되는지 또는 프레임에 대해 표시되는지 여부를 설정하기 위한 세팅을 포함하며,

상기 세팅은 판독가능하고, 이에 따라 상기 움직임 벡터 값들이 해석되는 것인, 네트워크 하드웨어 디바이스.

발명의 설명

기술분야

배경기술

- [0001] 현대 통신 시스템들에서, 비디오 신호는 유선 네트워크 및/또는 무선 네트워크와 같은 매체, 대개 인터넷과 같은 패킷 기반 네트워크를 통해 하나의 단말기로부터 다른 단말기에 전송될 수 있다. 예를 들어, 비디오는 데스크톱 또는 랩톱 컴퓨터, 태블릿 또는 스마트 폰과 같은 사용자 단말기에서 실행되는 VoIP(voice over Internet Protocol) 클라이언트 애플리케이션으로부터 처리되는 VoIP 호출(call)의 일부일 수 있다.
- [0002] 통상적으로, 비디오의 프레임은 네트워크를 통한 전송을 위해 이들을 압축하기 위해, 송신 단말기에서 인코더에 의해 인코딩된다. 주어진 프레임에 대한 인코딩은 블록이 동일한 프레임에서의 다른 블록에 관하여 인코딩되는 인트라 프레임(intra frame) 인코딩을 포함할 수 있다. 이 경우, 타겟 블록은 해당 블록과 이웃 블록 간의 차이(잔차) 면에서 인코딩된다. 대안적으로, 일부 프레임에 대한 인코딩은 타겟 프레임에서의 블록이 통상적으로 움직임 예측에 기초하여, 이전 프레임에서의 대응 부분에 관하여 인코딩되는 인터 프레임(inter frame) 인코딩을 포함할 수 있다. 이 경우, 타겟 블록은 해당 블록과 예측될 대응 부분 간의 오프셋, 및 해당 블록과 예측되는 대응 부분 간의 차이(잔차)를 식별하는 움직임 벡터 면에서 인코딩된다. 수신기에 있는 대응하는 디코더가 디코더 측의 스크린으로의 출력을 위해 수신된 비디오 신호의 프레임을 압축 해제하기 위해, 적절한 타입의 예측에 기초하여 상기 수신된 비디오 신호의 프레임을 디코딩한다.
- [0003] 비디오를 인코딩(압축)할 때, 움직임 벡터는 현재 프레임에 대한 인터 프레임 예측을 생성하기 위해 사용된다. 인코더는 먼저 현재 블록(타겟 블록)과 가장 일치하는 이전에 인코딩된 프레임에서의 유사 블록(기준 블록)을 검색하고, 인코딩된 비트 스트림의 일부로서 디코더에 기준 블록과 타겟 블록 간의 변위를 시그널링한다. 변위는 통상적으로 수평 및 수직의 x 및 y 좌표로서 표현되고, 움직임 벡터로 지칭된다.
- [0004] 기준 "블록"은 사실은 기준 프레임의 실제 블록 위치에 있는 것으로 제한되지 않고, 즉, 타겟 블록과 동일한 그리드로 한정되는 것이 아니라, 움직임 벡터만큼 타겟 블록의 위치에 관해 오프셋된 기준 프레임의 대응 크기 부분이다. 현재 표준에 따라, 움직임 벡터는 부화소 해상도(fractional pixel resolution)로 표현된다. 예를 들어, H.264 표준에서, 각각의 움직임 벡터는 $\frac{1}{4}$ 화소 해상도로 표현된다. 그래서 한 예로서, 현재 프레임의 16×16 블록이 타겟 블록 위치의 1 화소 좌측에 있는 이전 프레임의 다른 16×16 블록으로부터 예측될 경우, 그 움직임 벡터는 (4,0) 이다. 또는, 타겟 블록이 타겟 블록의 좌측으로 단지 화소의 $\frac{3}{4}$ 에 있는 기준 블록으로부터 예측될 경우, 움직임 벡터는 (3,0) 이다. 부화소 위치에 있는 기준 블록은 실제로 그 자체가 존재하는 것이 아니라, 기준 프레임의 화소들 간의 보간에 의해 생성된다. 서브 화소 움직임 벡터는 압축 효율 면에서 상당한 성능을 달성할 수 있다.
- ### 발명의 내용
- [0005] 그러나, 부화소 해상도를 사용하는 것은 움직임이 정수 화소 해상도(integer pixel resolution)에서 추정되었던 경우보다 움직임 벡터를 인코딩하기 위해 더 많은 비트를 초래하고, 또한 최상의 일치 기준을 검색하는데 더 많은 처리 자원을 초래한다. 비디오 코딩의 경우, 이것은 가치가 있을 수 있고, 예컨대, 더 많이 일치된 잔차의 감소된 크기는 일반적으로 움직임 벡터를 인코딩하는데 초래되는 비트보다 클 수 있기 때문에, 또는 획득된 품질은 자원을 조정하기 위해 고려될 수 있기 때문에, 가치가 있을 수 있다. 그러나, 인코딩될 모든 움직임은 이미지는 비디오(즉, 카메라에서 캡처됨)가 아니다. 본원에서는, 카메라보다는 스크린에서 캡처된 움직임은 이미지를 인코딩(압축)할 때, 인코딩된 비트 스트림의 움직임 벡터의 대부분이 일반적으로 정수 화소를 가리킬 것이고, 부화소 위치에는 발견되는 경향이 거의 없다는 것이 인식된다. 따라서, 인코더는 보통 $\frac{1}{4}$ 화소 단위로 비트 스트림에서 움직임 벡터를 나타내지만, 애플리케이션을 공유 또는 기록하는 스크린의 경우, 단지 1 화소 단위로 움직임 벡터를 인코딩함으로써, 대역폭은 실제로 품질의 과도한 손실 없이 절약될 수 있다.
- [0006] 그럼에도 불구하고, 분수 움직임 벡터가 보통의 비디오(카메라에 의해 캡처됨) 또는 아마 다른 움직임은 이미지(예컨대, 애니메이션)에 여전히 유용하다는 것을 고려하면, 움직임 벡터는 융통성 있는 방법으로 시그널링될 수 있다. 비디오 소스가 캡처된 스크린에서 오는 경우, 움직임 벡터는 1 화소 단위로 시그널링될 수 있지만, 보통의 비디오 및/또는 다른 움직임은 이미지의 경우, 부화소 단위가 여전히 사용될 수 있다.
- [0007] 보다 일반적으로, 예컨대, 인코더의 설계자가 임의의 원하는 트레이드 오프 또는 효과를 구현하기를 원하는 방

법에 따라, 부화소 또는 정수 화소 움직임 벡터 해상도가 사용되는지의 여부에 대한 제어를 갖는 것이 유용할 수 있는 여러 상황이 있을 수 있다. 예컨대, 아마도 일부 비디오 또는 애니메이션은 자신들의 특징의 일부 양태로 인해, 움직임 추정에서 정수 화소 해상도로 더욱 효율적으로 제공될 것이고, 다른 비디오 또는 다른 타입의 움직이는 이미지는 부화소 해상도로 더욱 효율적으로 제공될 수 있다.

- [0008] 따라서, 본원에 개시된 일 양태에 따라, 움직이는 이미지 데이터를 인코딩하는데 사용하기 위한 포맷이 제공되고, 그것에 의하여, 상기 포맷에 따라 인코딩된 움직이는 이미지 데이터는,
- [0009] · 적어도 하나의 영역이 움직임 추정을 사용하여 인코딩되는 복수의 프레임을 포함하는 프레임 시퀀스;
- [0010] · 상기 프레임 각각 또는 상기 프레임 각각 내의 하나 이상의 영역 각각에 대한 움직임 추정의 움직임 벡터를 나타내는 움직임 벡터 값의 세트; 및
- [0011] · 개개의 프레임 또는 영역의 개개의 움직임 벡터 값이 제 1 해상도 또는 제 2 해상도로 인코딩되는지를 나타내는, 상기 개개의 프레임 또는 영역 각각과 연관된 적어도 하나의 지표를 포함한다.
- [0012] 제 1 해상도로 인코딩된 움직임 벡터 값은 많은 수의 미세한 단계(finer step)를 갖는 스케일로 표현되고, 제 2 해상도로 인코딩된 움직임 벡터 값은 적은 수의 대강의 단계(coarser step)를 갖는 스케일로 표현되는 프로토콜에 따라 움직임 벡터 값은 인코딩되고, 이에 의해, 인코딩된 비트 스트림에 평균적으로 더 적은 비트를 초래한다. 대강의 단계는 정수 화소 단위를 나타내며, 미세한 단계는 부화소 단위를 나타낸다.
- [0013] 본원에 개시된 다른 양태에 따라, 포맷 또는 프로토콜에 따라 인코딩된 움직이는 이미지 데이터의 비트 스트림을 전달하는 네트워크 요소 또는 컴퓨터 판독 가능 저장 매체가 제공된다.
- [0014] 실시예에서, 제 1 해상도로 인코딩된 복수의 프레임 또는 영역의 일부 및 제 2 해상도로 인코딩된 복수의 프레임 또는 영역의 다른 부분을 포함하는 비트 스트림이 제공될 수 있고, 개개의 지표는 상기 복수의 (인터 프레임 인코딩된) 프레임 또는 영역 각각에 대해 개별적으로 해상도를 나타낸다.
- [0015] 실시예에서, 각각의 프레임 또는 영역의 움직임 벡터 값 각각의 값은 인코딩된 비트 스트림의 움직임 벡터 필드에 포함될 수 있고, 상기 프로토콜에 따라, 움직임 벡터 필드는 움직임 벡터가 제 2 해상도로 인코딩되는 프레임 또는 영역에 대해 감소된 크기를 가질 수 있다.
- [0016] 본원에 개시된 다른 양태에 따라, 인코딩된 형태의 움직이는 이미지 데이터를 수신하기 위한 입력부, 및 움직임 예측 모듈을 포함하는 디코더가 제공된다. 본원에 개시된 실시예들 중 일부에 따르면, 움직이는 이미지 데이터는 적어도 하나의 영역이 포맷 또는 프로토콜에 기반을 둔 움직임 추정을 사용하여 인코딩되는 복수의 프레임 (즉, 인터 프레임 인코딩된 프레임)을 포함한다. 움직임 예측 모듈은 움직임 벡터 값에 기초하여 상기 (인터 프레임 인코딩된) 프레임 또는 영역을 디코딩한다. 이것은 개개의 프레임 또는 영역의 움직임 벡터 값이 제 1 해상도 또는 제 2 해상도로 인코딩되는지, 제 1 해상도가 부화소 단위로 움직임 벡터 값을 해석하는 경우인지, 제 2 해상도가 정수 화소 단위로 움직임 벡터 값을 해석하는 경우인지를 결정하기 위해 지표 각각을 판독하는 것을 포함한다.
- [0017] 실시예에서, 움직이는 이미지 데이터는 상기 프레임 또는 영역 각각과 연관된 두 개의 지표를 포함할 수 있고, 두 개의 지표는 2 차원으로 움직임 벡터의 해상도를 나타내고, 움직임 예측 모듈은 두 개의 지표 모두를 판독하고, 그에 따라 움직임 벡터 값을 해석하도록 구성될 수 있다.
- [0018] 실시예에서, 상기 프레임 중 적어도 일부 프레임 각각은 다수의 영역으로 분할될 수 있고; 움직이는 이미지 데이터는 각각의 영역의 움직임 벡터 값이 제 1 해상도 또는 제 2 해상도로 인코딩되는지를 개별적으로 나타내기 위해 다수의 영역 각각과 연관된 적어도 하나의 지표를 포함할 수 있고; 움직임 예측 모듈은 각각의 영역의 움직임 벡터 값이 제 1 해상도 또는 제 2 해상도로 인코딩되는지를 결정하기 위해 지표를 판독하고, 그에 따라 움직임 벡터 값을 상기 부화소 또는 정수 화소 단위로 해석하도록 구성될 수 있다. 실시예에서, 상기 영역은 H.26x 표준의 슬라이스일 수 있다.
- [0019] 실시예에서, 움직이는 이미지 데이터는 움직임 벡터 값의 해상도가 영역마다 또는 프레임마다 나타나는지를 설정하기 위해 설정부(setting)를 더 포함할 수 있고, 움직임 예측 모듈은 그 설정부를 판독하고, 그에 따라 움직임 벡터 값을 해석하도록 구성될 수 있다.
- [0020] 다른 실시예들에서, 움직임 예측 모듈은, 지표가 상기 프레임 또는 영역 중 하나에 대해 존재하지 않는 경우, 움직임 벡터의 값을 디폴트(default)로 부화소 단위로 해석하도록 구성될 수 있다.

[0021] 또 다른 실시예에서, 움직임 벡터를 포함하는 움직이는 이미지 데이터는 또한 무손실 인코딩 방법에 따라 인코딩될 수 있다. 디코더는 움직임 예측 모듈에 의한 상기 디코딩 이전에 무손실 인코딩 스테이지의 역을 포함할 수 있다.

[0022] 또 다른 양태에 따르면, 컴퓨터 프로그램 제품이 제공되고, 상기 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 저장 매체에 구현되고, 실행되는 경우 본 명세서에 개시된 임의의 실시예에 따른 디코더의 동작을 수행하도록 구성된다.

[0023] 이 요약은 아래의 상세한 설명에서 더욱 기술되는 개념들의 선택을 간략화된 형태로 소개하기 위해서 제공된다. 요약은 주장되는 대상의 중요 특징 또는 필수 특징을 식별하기 위한 것도 아니고, 주장되는 대상의 범위를 한정하기 위해 사용되는 것도 아니다. 주장되는 대상은 본원에 언급된 단점들 중 일부 또는 모두를 해결하는 구현에 한정되는 것도 아니다.

도면의 간단한 설명

[0024] 본 개시의 이해를 돕기 위해 그리고 실시예들이 실행될 수 있는 방법을 보여주기 위해, 첨부 도면을 예로서 참조한다.

도 1은 비디오 스트림의 개략도이다.

도 2는 통신 시스템의 개략적 블록도이다.

도 3은 인코딩된 비디오 스트림의 개략도이다.

도 4는 인코더의 개략적 블록도이다.

도 5는 디코더의 개략적 블록도이다.

도 6은 인터 프레임 인코딩 방식의 개략도이다.

도 7은 다른 인터 프레임 인코딩 방식의 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0025] 도 1은 카메라에서 캡처되어, 공간 분할로 분할된 입력 비디오 신호의 개략도를 제공하는 것으로, 입력 비디오 신호는 인코딩된 비트 스트림을 생성하기 위해 비디오 인코더에 의해 인코딩될 것이다. 신호는 시간적으로 복수의 프레임(F)으로 분할된 움직이는 비디오 이미지를 포함하고, 각각의 프레임은 시간적으로 상이한 개개의 순간($\dots t-1, t, t+1 \dots$)에서 이미지를 나타낸다. 각 프레임 내에서, 프레임은 각각이 복수의 화소를 나타내는 복수의 분할로 공간적으로 분할된다. 이러한 분할은 블록으로 지칭될 수 있다. 특정 방식에서, 프레임을 상이한 레벨의 블록으로 분할 및 세분된다. 예를 들어, 각각의 프레임은 매크로 블록(macroblock; MB)으로 분할될 수 있고, 각각의 매크로 블록은 블록(b)으로 분할될 수 있으며, 예컨대, 각각의 블록은 프레임 내의 8×8 화소의 영역을 나타내고, 각각의 매크로 블록은 2×2 블록(16×16 화소)의 영역을 나타낸다. 특정 방식에서, 각각의 프레임은 또한 각각이 복수의 매크로 블록을 포함하는 독립적으로 디코딩 가능한 슬라이스(S)로 분할될 수 있다. 슬라이스(S)는 일반적으로 임의의 형태를 취할 수 있고, 예컨대, 각각의 슬라이스는 한 행 이상의 매크로 블록이거나, 불규칙하게 또는 임의로 정의된 매크로 블록의 선택(예컨대, 이미지에서 관심 영역(ROI)에 대응함)일 수 있다.

[0026] 용어 "화소"에 관해, 다음에서, 용어는 픽처 어레이에 대한 샘플링 그리드에서 샘플 및 샘플링 위치를 나타내기 위해 사용된다(때때로, 문헌에서, 용어 "화소"는 하나의 공간 위치에 대응하는 세 개의 색상 컴포넌트 모두를 나타내기 위해 대신 사용되며, 때때로, 이것은 단일 어레이에서 단일 정수 샘플 값 또는 단일 위치를 나타내기 위해 사용된다). 샘플링 그리드의 해상도는 루마 및 크로마 샘플링 어레이 사이에서 대개 상이하다. 실시예에서, 다음은 4:4:4 표현에 적용될 수 있지만, 이것은 잠재적으로, 예를 들어, 4:2:2 및 4:2:0 에도 적용될 수 있다.

[0027] 임의의 주어진 표준이 용어 블록 또는 매크로 블록에 특정 의미를 제공할 수 있지만, 용어 블록은 또한 인트라 또는 인터 예측과 같은 인코딩 및 디코딩 동작이 수행되는 레벨에서 프레임의 분할을 나타내기 위해서 당해 분야에서 대개 더욱 일반적으로 사용되며, 이것은 특별히 달리 언급되지 않는 한 본원에서 사용될 일반적인 의미인 것을 주의한다. 예를 들어, 본원에 나타나는 블록은 사실은 H.26x 표준에서 블록 또는 매크로 블록이라고 불리는 분할일 수 있고, 다양한 인코딩 및 디코딩 스테이지가 논의되고 있는 표준, 애플리케이션 및/또는 인코딩

모드에 적절한 임의의 분할 레벨에서 동작할 수 있다.

- [0028] 캡처된 바와 같은 입력 신호의 블록은 보통 공간 도메인으로 표현되고, 여기서 각각의 색 공간 채널이 그 블록 내에서 공간 위치의 함수로서 표현된다. 예를 들어, YUV 색 공간에서, 휘도(Y) 및 색차(U, V) 채널 각각은 직교 좌표 x 및 y 의 함수, $Y(x,y)$, $U(x,y)$ 및 $V(x,y)$ 로서 표현될 수 있고; 또는 RGB 색 공간에서, 적색(R), 녹색(G), 및 청색(B) 채널 각각은 직교 좌표의 함수, $R(x,y)$, $G(x,y)$, $B(x,y)$ 로서 표현될 수 있다. 이 표현에서, 각각의 블록 또는 부분은 상이한 공간 좌표, 예컨대, x 및 y 좌표에서, 화소 값들의 세트로 표현되어, 색 공간의 각각의 채널은 이산 세트의 화소 위치 각각에서 그 채널의 크기 면으로 표현된다.
- [0029] 그러나, 양자화 이전에, 블록은 인코딩 프로세스의 일부로서 변환 도메인 표현, 통상적으로 공간 주파수 도메인 표현(때때로, 주파수 도메인으로 언급됨)으로 변환될 수 있다. 주파수 도메인에서, 블록에서의 각각의 색 공간 채널은 2 차원 각각에서 공간 주파수의 함수(1/길이의 치수)로서 표현된다. 예를 들어, 이것은 수평 및 수직 방향에서 각각 파수 k_x 및 k_y 로 나타낼 수 있어서, 결국 채널은 YUV 공간에서 $Y(k_x, k_y)$, $U(k_x, k_y)$ 및 $V(k_x, k_y)$; 또는 RGB 공간에서 $R(k_x, k_y)$, $G(k_x, k_y)$, $B(k_x, k_y)$ 로 표현될 수 있다. 따라서, 이산 세트의 화소 위치 각각에서 크기 면으로 색 공간 채널을 나타내는 대신에, 변환은 각각의 색 공간 채널을 그 블록을 구성하는 이산 세트의 공간 주파수 컴포넌트 각각과 연관된 계수 면으로, 즉, 블록에 걸쳐 상이한 주파수의 공간 변화에 대응하는 이산 세트의 공간 주파수 성분 각각의 진폭 면으로 나타난다. 이러한 변환에 대한 가능성은 푸리에 변환, 이산 코사인 변환(Discrete Cosine Transform; DCT), 카루넨 루베 변환(Karhunen-Loeve Transform; KLT), 또는 기타 변환을 포함한다.
- [0030] 도 2의 블록도는 본 개시의 기술이 사용될 수 있는 통신 시스템의 일례를 제공한다. 통신 시스템은 제 1 송신 단말기(12), 및 제 2 수신 단말기(22)를 포함한다. 예를 들어, 각각의 단말기(12, 22)는 휴대 전화 또는 스마트폰, 태블릿, 랩톱 컴퓨터, 데스크톱 컴퓨터, 또는 텔레비전 세트, 셋톱 박스, 스테레오 시스템 등과 같은 다른 가전 제품 중 하나를 포함할 수 있다. 제 1 및 제 2 단말기(12, 22)는 각각 통신 네트워크(32)에 작동적으로 결합되고, 제 1 송신 단말기(12)는 이에 의해 제 2 수신 단말기(22)에 의해 수신될 신호를 전송하도록 구성된다. 물론, 송신 단말기(12)는 또한 수신 단말기(22)로부터 신호를 수신할 수 있고, 그 반대로 가능하지만, 설명을 위해, 전송은 제 1 단말기(12)의 관점에서 본원에서 설명되고, 수신은 제 2 단말기(22)의 관점에서 설명된다. 통신 네트워크(32)는, 예를 들어, 광역 인터넷 및/또는 로컬 영역 네트워크 및/또는 이동 셀룰러 네트워크와 같은 패킷 기반 네트워크를 포함할 수 있다.
- [0031] 제 1 단말기(12)는 플래시 메모리 또는 다른 전자 메모리, 자기 저장 장치, 및/또는 광학 저장 장치와 같은 컴퓨터 판독 가능 저장 매체(14)를 포함한다. 제 1 단말기(12)는 또한 하나 이상의 실행 유닛을 갖는 프로세서 또는 CPU 형태의 처리 장치(16), 송신기(18)를 갖는 유선 또는 무선 모델과 같은 트랜스미버, 비디오 카메라(15), 및 스크린 (17)(즉, 디스플레이 또는 모니터)를 포함한다. 카메라(15) 및 스크린(17) 각각은 단말기(12)의 나머지 부분과 동일한 케이싱 내에 수용될 수도 있고 수용되지 않을 수도 있다(예컨대, 후자의 경우 동글 또는 무선 라우터를 포함하는 송신기(18)가 심지어 내부 또는 외부에 있을 수 있음). 저장 매체(14), 비디오 카메라(15), 스크린(17), 및 송신기(18)는 각각 처리 장치(16)에 작동적으로 결합되며, 상기 송신기(18)는 유선 또는 무선 링크를 통해 네트워크(32)에 작동적으로 결합된다. 유사하게, 제 2 단말기(22)는 전자, 자기, 및/또는 광학 저장 장치와 같은 컴퓨터 판독 가능 저장 매체(24); 및 하나 이상의 실행 유닛을 갖는 CPU 형태의 처리 장치(26)를 포함한다. 제 2 단말기는 적어도 하나의 수신기(28)를 갖는 유선 또는 무선 모델과 같은 트랜스미버, 및 단말기(22)의 나머지 부분과 동일한 케이싱 내에 수용될 수도 있고 수용되지 않을 수도 있는 스크린(25)을 포함한다. 제 2 단말기의 저장 매체(24), 스크린(25), 및 수신기(28)는 각각 처리 장치(26)에 작동적으로 결합되며, 상기 수신기(28)는 유선 또는 무선 링크를 통해 네트워크(32)에 작동적으로 결합된다.
- [0032] 제 1 단말기(12) 상의 저장 매체(14)는 움직이는 이미징 데이터를 인코딩하기 위한 인코더를 적어도 저장하고, 상기 인코더는 처리 장치(16)에서 실행되도록 구성된다. 인코더가 실행되면, 비디오 카메라(15)로부터 "미가공(raw)"(비인코딩) 입력 비디오 스트림을 수신하고, 이것은 비디오 스트림을 낮은 비트 레이트 스트림으로 압축하기 위해서 비디오 스트림을 인코딩하도록 동작하며, 송신기(18) 및 통신 네트워크(32)를 통한 전송을 위해 상기 인코딩된 비디오 스트림을 제 2 단말기(22)의 수신기(28)에 출력한다. 제 2 단말기(22) 상의 저장 매체(24)는 그 자신의 처리 장치(26)에서 실행되도록 구성된 비디오 디코더를 적어도 저장한다. 디코더가 실행되면, 수신기(28)로부터 인코딩된 비디오 스트림을 수신하고, 이를 스크린(25)으로 출력하기 위해 디코딩한다.
- [0033] 인코더 및 디코더는 또한 스크린 공유 스트림을 비롯한 다른 타입의 움직이는 이미지 데이터를 인코딩 및 디코딩하기 위해 동작 가능하다. 스크린 공유 스트림은 한 명 이상의 다른 원격 사용자들이 인코더 측에 있는 사용

자가 스크린을 통해 보는 것을 볼 수 있도록, 또는 그 스크린의 사용자가 나중에 한 명 이상의 다른 사용자에게 재생하기 위해 스크린 상에 무슨 일이 있었는지를 기록할 수 있도록 인코더 측에 있는 스크린(17)에서 캡처된 이미지 데이터이다. 송신 단말기(12) 및 수신 단말기(22) 사이에 수행된 호출(call)의 경우, 송신 단말기(12)에 있는 스크린(17)의 움직이는 콘텐츠는 인코딩되어 수신 단말기(22)의 스크린(25) 상에 디코딩 및 디스플레이되도록 라이브로(실시간으로) 전송될 것이다. 예를 들어, 인코더 측 사용자는 자신이 자신의 운영 체제의 데스크탑, 또는 일부 애플리케이션을 작동하는 방법을 다른 사용자와 공유하기를 원할 수 있다.

[0034] 스크린 공유 스트림이 스크린 등으로부터 캡처되는 경우, 이는 그렇게 하기 위해 어떤 특정한 메커니즘으로도 한정되지 않는다는 것을 주의한다. 예컨대, 데이터는 스크린(17)의 스크린 버퍼로부터 판독되거나, 또는 스크린(17) 상의 디스플레이용 애플리케이션으로부터 또는 운영 체제로부터 출력되는 동일한 그래픽 데이터의 인스턴스를 수신함으로써 캡처될 수 있다.

[0035] 도 3은 송신 단말기(12) 상에서 작동하는 인코더에서부터 수신 단말기(22) 상에서 작동하는 디코더에 전송되는 바와 같은 인코딩된 비트 스트림(33)의 개략도를 제공한다. 비트 스트림(33)은 임의의 연관된 움직임 벡터와 함께 그 프레임 또는 슬라이스의 블록에 대한 인코딩된 샘플을 포함하는 각각의 프레임 또는 슬라이스에 대한 인코딩된 이미지 데이터(34)를 포함한다. 하나의 애플리케이션에서, 비트 스트림은 송신 단말기(12)와 수신 단말기(22) 간의 VoIP 호출(VoIP 호출은 또한 비디오 및 스크린 공유를 포함할 수 있다)과 같은 라이브(실시간) 호출의 일부로서 전송될 수 있다. 비트 스트림(33)은 또한 각각의 프레임 또는 슬라이스와 연관된 헤더 정보(36)를 포함한다. 실시예에서, 헤더(36)는 움직임 벡터의 해상도를 나타내는 적어도 하나의 플래그(37)의 형태로 적어도 하나의 부가적인 요소를 포함하도록 구성되고, 이는 이하에서 보다 상세히 논의될 것이다.

[0036] 도 4는 송신 단말기(12) 상에 구현될 수 있는 인코더를 도시한 블록도이다. 인코더는 메인 인코더 모듈(40)을 포함하고, 메인 인코더 모듈(40)은 이산 코사인 변환(DCT) 모듈(51), 양자화기(53), 역 변환 모듈(61), 역 양자화기(63), 인트라 예측 모듈(41), 인터 예측 모듈(43), 스위치(47), 감산 스테이지(-)(49), 및 무손실 인코딩 스테이지(65)를 포함한다. 인코더는 또한 인터 예측 모듈(43)에 결합된 제어 모듈(50)을 포함한다. 이러한 모듈 또는 스테이지들 각각은 송신 단말기의 저장 매체(14) 상에 저장되거나 그 처리 장치(16) 상에서 실행하기 위해 구성된 코드의 일부분으로서 구현될 수 있지만, 전체적으로 또는 부분적으로 전용 하드웨어 회로에 구현되는 이들 중 일부 또는 전부의 가능성은 배제되지 않는다.

[0037] 감산 스테이지(49)는 복수의 프레임(F)을 통해 복수의 블록을 포함하는 입력 신호의 인스턴스를 수신하도록 구성된다. 입력 스트림은 카메라(15)로부터 수신되거나, 또는 스크린(17) 상에 디스플레이되는 것으로부터 캡처된다. 인트라 또는 인터 예측 모듈(41, 43)은 다른 이미 인코딩된 블록 또는 대응 크기의 기준 부분으로부터의 예측에 기초하여, 인코딩될 현재(타겟) 블록의 예측 버전을 생성한다. 예측 버전은 감산 스테이지(49)의 입력에 공급되고, 감산 스테이지(49)에서, 이것은 공간 도메인의 입력 신호(즉, 실제 신호)로부터 감산되어 블록의 예측 버전과 실제 입력 신호의 대응 블록 간의 차이를 나타내는 잔차 신호를 생성한다.

[0038] 인트라 예측 모드에서, 인트라 예측 모듈(41)은 동일한 프레임의 다른 이미 인코딩된 블록, 통상적으로 이웃 블록으로부터의 예측에 기초하여 인코딩될 현재(타겟) 블록의 예측 버전을 생성한다. 인트라 프레임 인코딩을 수행할 때, 아이디어는 프레임 내의 이미지 데이터의 부분이 동일한 프레임 내의 다른 부분과 얼마나 상이한지에 대한 측정치만 인코딩하여 전송하는 것이다. 그런 다음, 그 부분은 디코더에서 예측될 수 있고(시작하기 위한 일부 절대 데이터가 제공됨), 그래서 실제 데이터 자체 보다는 예측과 실제 데이터 간의 차이를 전송하는 것만이 필요하다. 차이 신호는 통상적으로 크기가 작아서, (무손실 압축 스테이지(65)의 동작으로 인해-아래 참조) 인코딩하는데 더 적은 비트를 사용한다.

[0039] 인터 예측 모드에서, 인터 예측 모듈(43)은 현재 블록과는 상이한 프레임에서의 다른 이미 인코딩된 기준 부분으로부터의 예측에 기초하여 인코딩될 현재(타겟) 블록의 예측 버전을 생성하고, 여기서 기준 부분은 블록 크기를 갖지만, 인터 예측 모듈(43)에 의해 예측되는 움직임 벡터만큼 공간 도메인에서 타겟 블록에 관하여 오프셋된다(인터 예측은 또한 움직임 예측 또는 움직임 추정으로 지칭될 수 있다). 인터 예측 모듈(43)은 공간 도메인에서 타겟 프레임 이외의 하나 이상의 프레임에서 복수의 가능한 움직임 벡터만큼 오프셋된 복수의 후보 기준 부분을 검색하고, 적당한 메트릭에 따라 타겟 블록에 대한 잔차를 최소화하는 후보를 선택함으로써, 주어진 타겟 블록에 대한 최적의 기준을 선택한다. 인터 예측 모듈(43)은 인트라 프레임 예측 스테이지(41) 대신에, 스위치(47)에 의해 피드백 경로로 스위칭되어, 이에 따라서, 다른 프레임의 블록들에 관하여 인터 프레임을 인코딩하기 위해서, 하나의 프레임과 다른 프레임의 블록들 사이에 피드백 루프가 생성된다. 즉, 잔차는 이제 인터 예측된 블록과 실제 입력 블록 간의 차이를 나타낸다. 이것은 통상적으로 인트라 프레임 인코딩보다 인코딩에 훨

원 더 적은 비트를 사용한다.

- [0040] 잔차 신호(예측이 입력 신호로부터 감산된 이후의 잔차 블록을 포함함)의 샘플은 감산 스테이지(49)로부터 출력되고, 변환(DCT) 모듈(51) (또는 다른 적절한 변환)을 통해, 이러한 잔차 값들은 주파수 도메인으로 변환되고, 그런 다음, 양자화기(53)에서, 변환된 값들은 실질적으로 이산 양자화 인덱스로 변환된다. 상기 변환 모듈(51) 및 양자화 모듈(53)에 의해 생성된 잔차의 양자화되고 변환된 인덱스는 물론, 예측 모듈들(41, 43)에서 사용되는 예측의 표시, 및 인터 예측 모듈(43)에 의해 생성된 임의의 움직임 벡터는 인코딩된 비디오 스트림(33)(도 3의 요소(34) 참조)에 포함하기 위해 모두 출력된다; 골롬 인코더 또는 엔트로피 인코더와 같은 추가의 무손실 인코딩 스테이지(65)를 통해, 움직임 벡터, 및 변환되고 양자화된 인덱스는 당해 분야에 공지된 무손실 인코딩 기술을 사용하여 더욱 압축된다.
- [0041] 양자화되고 변환된 신호의 인스턴스가 또한 역 양자화기(63) 및 역 변환 모듈(61)을 통해 피드백되어, 인코딩되는 현재 타겟 블록이 이전에 인코딩된 블록의 역 양자화 및 역 변환 버전에 기초하여 예측되었던 동일한 방식으로, 인코딩될 후속 블록을 예측하는데 있어서 선택된 예측 모듈(41 또는 43)에 의한 사용을 위해 블록의 예측 버전(디코더에서 볼 수 있음)을 생성한다. 스위치(47)는 현재 인코딩되는 프레임 또는 블록에 대해 사용된 인코딩에 적절하게 인트라 예측 모듈(41) 또는 인터 예측 모듈(43) 중 어느 하나의 입력에 역 양자화기(63)의 출력을 통과시키도록 구성된다.
- [0042] 도 5는 수신 단말기(22) 상에 구현될 수 있는 디코더를 도시한 블록도이다. 디코더는 역 무손실 인코딩(95), 역 양자화 스테이지(83), 역 DCT 변환 스테이지(81), 스위치(70), 인트라 예측 스테이지(71), 및 움직임 보상 스테이지(73)를 포함한다. 이러한 모듈 또는 스테이지들 각각은 수신 단말기의 저장 매체(24) 상에 저장되거나 그 처리 장치(26) 상에서 실행하기 위해 구성된 코드의 일부분으로서 구현될 수 있지만, 전체적으로 또는 부분적으로 전용 하드웨어 회로에 구현되는 이들 중 일부 또는 전부의 가능성은 배제되지 않는다.
- [0043] 역 양자화기(81)는 수신기(28) 및 역 무손실 인코딩 스테이지(95)를 통해 인코더로부터의 인코딩된 신호(33)를 수신하도록 구성된다. 역 양자화기(81)는 잔차 신호(잔차 블록을 포함함)의 역 양자화된 샘플로 인코딩된 신호의 양자화 인덱스를 변환하고, 역 양자화된 샘플을 역 DCT 모듈(81)에 통과시키며, 여기서 역 양자화된 샘플은 주파수 도메인에서 다시 공간 도메인으로 변환된다. 그런 다음, 스위치(70)는 디코딩되는 현재의 프레임 또는 블록에 사용되는 예측 모드에 적절하게 인트라 또는 인터 예측 모듈(71 또는 73)에 역 양자화된 공간 도메인 잔차 샘플을 통과시키고, 인트라 또는 인터 예측 모듈(71, 73)은 블록을 디코딩하기 위해 각각 인트라 또는 인터 예측을 사용한다. 어떤 모드를 사용할지는 예측의 표시 및/또는 인코딩된 비트 스트림(33)에서 인코딩된 샘플(34)과 함께 수신된 임의의 움직임 벡터를 사용하여 결정된다. 이 스테이지에 이어, 디코딩된 블록은 수신 단말기(22)에서 스크린(25)을 통해 재생되도록 출력된다.
- [0044] 위에서 말한 바와 같이, 종래의 표준에 따른 코덱은 1/4 화소의 해상도로 움직임 예측을 수행하고, 이는 움직임 벡터가 또한 1/4 화소 스텝 면에서 표현된다는 것을 의미한다. 1/4 화소 해상도 움직임 추정의 예가 도 6에 도시된다. 이 예에서, 타겟 블록의 좌측 상부 코너의 화소(p)는 화소들(a, b, c 및 d) 간의 보간으로 예측되고, 타겟 블록의 다른 화소들도 또한 하나의 프레임의 타겟 블록과 다른 프레임의 기준 부분(이러한 블록들은 도 6에서 굵은 점선으로 도시됨) 간의 오프셋에 따라, 기준 프레임에 있는 화소들의 그룹 간의 유사한 보간에 기초하여 예측될 것이다. 그러나, 이러한 단위로 움직임 추정을 수행하는 것은 아래에 논의된 바와 같은 결과를 갖는다.
- [0045] 무손실 코더(65) 및 디코더(95)를 참조하면, 무손실 코딩은 정보를 버리는 것(양자화 같은 것)이 아니라, 무손실 인코딩 스테이지(65)에 의해 인코딩될 데이터에, 상이한 값들이 발생할 가능성 또는 상이한 값들이 발생하는 빈도에 따라 상이한 값들을 나타내기 위해 상이한 길이의 코드워드를 사용함으로써 작동하는 압축 형태이다. 예를 들어, 1을 직면하기 이전의 코드워드에서의 선두 0의 수는 코드워드의 길이를 나타낼 수 있으므로, 1은 가장 짧은 코드워드이고, 010 및 011은 그 다음 짧은 코드워드이며, 그 다음은 00100 ..., 등이다. 따라서, 가장 짧은 코드워드는 균일한 코드워드 길이가 사용되었을 경우에 필요한 것보다 훨씬 짧지만, 가장 긴 코드워드는 그것보다 길다. 그러나, 가장 짧은 코드워드에 가장 빈번하거나 가능성이 많은 값을 할당하고, 긴 코드워드에 가능성이 적거나 덜 빈번하게 발생하는 값만을 할당함으로써, 결과 비트 스트림(33)은 균일한 코드워드 길이가 사용되었던 경우보다 평균적으로 인코딩된 값 당 더 적은 비트를 초래하므로, 어떠한 추가의 정보도 폐기하지 않고 압축을 달성한다.
- [0046] 무손실 인코딩 스테이지(65) 이전의 인코더(40)의 대부분이 무손실 코딩 스테이지(65)를 통과하기 전에 많은 값들을 가능한 한 작게 만들려고 노력하도록 설계된다. 그들이 더 자주 발생할 때, 작은 값은 큰 값보다 인코딩된

비트 스트림(33)에 낮은 비트 레이트를 초래할 것이다. 이것은 절대 샘플이 아니라 잔차가 인코딩되는 이유이다. 많은 샘플이 변환 도메인에서 제로 또는 작은 계수로 변환하는 경향이 있기 때문에, 이것은 변환(51)을 뒷받침하는 근거이다.

[0047] 유사한 고려 사항이 움직임 벡터의 인코딩에 적용될 수 있다.

[0048] 예를 들어, H.264/MPEG-4 Part 10, 및 H.265/HEVC에서, 움직임 벡터는 지수 곱셈 코딩으로 인코딩된다. 다음 표는 움직임 벡터 값 및 인코딩된 비트를 도시한다.

[0049]	값	코드워드	발생된 비트 수
[0050]	0	1	1
[0051]	1	010	3
[0052]	2	011	3
[0053]	3	00100	5
[0054]	4	00111	5
[0055]	5	0001000	7
[0056]	...		

[0057] 위의 표로부터, 값이 크면 클수록, 더 많은 비트가 사용된다는 것을 알 수 있다. 이것은 움직임 벡터의 해상도가 크면 클수록, 더 많은 비트가 초래한다는 것을 의미한다. 예컨대, 1/4 화소 해상도를 사용하면, 1 화소의 오프셋은 인코딩된 비트 스트림에서 5 비트를 초래하는 4의 값으로 표현되어야 한다.

[0058] 비디오(카메라에서 캡처됨)를 인코딩하는데 있어서, 더 정밀한 해상도가 낮은 비용의 잔차 기준에 대한 검색에 더 많은 기회를 제공할 수 있기 때문에, 움직임 벡터의 이러한 해상도의 비용은 가치가 있을 수 있다. 그러나, 스크린에서 캡처된 움직이는 이미지의 경우, 대부분의 공간 변위는 전체 화소 변위에 있는 경향이 있고, 부화소 위치에 있는 경향은 거의 없으므로, 대부분의 움직임 벡터는 정수 화소 값을 가리키는 경향이 있고, 부화소 값을 가리키는 경향은 거의 없는 것으로 본원에서 관찰된다.

[0059] 이것에 기초하여, 스크린에서 캡처된 이미지 데이터에 대한 움직임 벡터를 1 화소의 해상도로 인코딩하는 것이 바람직할 수 있다. 어떠한 비트도 이러한 콘텐츠에 대한 움직임 벡터의 분수 부분에 사용될 필요가 없다는 사실을 고려하면, 이는 이러한 콘텐츠를 인코딩하는데 초래되는 비트 레이트가 감소될 수 있다는 것을 의미한다.

[0060] 예를 들어, 인코더는 일반적으로 1/4 화소 오프셋 단위로 비트 스트림에서의 움직임 벡터를 해석하지만, 인코더는 사실 이 해상도를 포기하고 대신에 정수 화소 오프셋 단위로 스크린 코딩 애플리케이션에 대한 움직임 벡터를 인코딩하여 비트 레이트를 절약할 수 있다. 이것은 움직임 벡터의 정밀도를 4 배로 감소시킬 것이지만, 이러한 정밀도는 일반적으로 화면 공유 또는 기록 애플리케이션에 덜 가치 있으며, 이는 또한 벡터를 코딩하는데 필요한 비트 수를 감소시킨다. 타겟 블록의 1 화소 왼쪽의 기준 블록으로부터의 현재(타겟) 블록을 예측하기 위해, 움직임 벡터는 (4,0) 대신 (1,0)일 것이다. 상기 곱셈 인코딩을 사용하면, 이것은 움직임 벡터를 인코딩하기 위해 초래되는 비트가 (00111, 1)에서부터 (010, 1)로 변화하므로, 이 경우에 2 비트가 절약된다는 것을 의미한다.

[0061] 또한, 실시예에서, 감소된 해상도의 움직임 벡터는 또한 움직임 벡터 검색을 정수 값으로 제한하여, 이에 의해 검색에 의해 초래되는 처리 자원을 감소시킴으로써, 인코더에서 수행되는 움직임 추정의 복잡성을 감소시킬 수 있다. 대안적으로, 일반 검색을 수행하고 결과의 움직임 벡터를 정수 값으로 반올림하는 것이 가능하다.

[0062] 도 7은 오직 전체 화소의 해상도로 제한된 움직임 벡터의 예를 도시하고, 움직임 벡터는 오직 전체 화소 스텝으로 제한된다. 도 6과는 대조적으로, 화소(p)는 어떠한 보간 없이 오직 단일의 전체 화소로부터 예측된다. 대안적으로, 이것은 한 프레임의 타겟 블록과 다른 프레임의 기준 부분(다시 한 번, 굵은 점선으로 도시됨) 간의 오프셋에 따라 화소(b, c, d) 또는 다른 화소로부터 예측될 수 있지만, 제한으로 인해, 이것은 화소들 간의 보간으로 예측되지 않을 수 있다. 주의: 임의의 주어진 블록의 경우, 도 6의 예로서 예시된 바와 같은 1/4 화소 예측은 그것이 가장 낮은 잔차를 제공하는 경우 어떠한 보간 없이 전체 화소 오프셋을 생성하게 될 수 있다. 그러나, 이것은 그렇게 하도록 제한되지 않았고, 상당한 크기의 이미지 위에서 모든 블록에 대해 발생하는 것은 매우 가능성이 없을 것이다.

- [0063] 분수 움직임 벡터 값이 여전히 카메라 캡처된 콘텐츠에 매우 유용할 수 있다는 것을 고려하면, 실시예에서, 인코더(40)에는 움직임 예측 모듈(43)에 결합된 제어기(50)가 제공되고, 상기 제어기(50)는 융통성 있는 방법으로 움직임 벡터 해상도를 선택하도록 구성되고, 소스 데이터가 캡처된 스크린(17)에서 오고 어떠한 부화소 움직임도 없는 경우, 움직임 벡터는 단지 전체 화소의 단위로 인코딩되어 전송되지만; 카메라 콘텐츠 비디오의 경우, 움직임 벡터는 여전히 부화소 정밀도로 인코딩되어 전송된다.
- [0064] 이를 위해, 제어기(50)는 캡처되는 콘텐츠의 타입이 스크린 콘텐츠라는 사실을 나타내는 성능 휴리스틱(performance heuristic)을 측정하도록 구성될 수 있다. 이에 응답하여, 그 다음에 스크린 콘텐츠 코딩에 부화소 움직임 보상을 사용하지 않는다. 대안적으로, 제어기(50)는 어떤 타입의 데이터가 인코딩을 위해 인코더에 제공되지에 관해 운영 체제 또는 애플리케이션으로부터의 지시를 수신할 수 있고, 제어기(50)는 그것에 기초하여 모드를 선택할 수 있다. 다른 옵션은 이력 데이터에 기초하여 선택될 수 있다. 선택은 프레임 단위로 행해질 수 있거나, 모드는, 예컨대, 슬라이스 단위로, 프레임 내의 상이한 영역에 대해 개별적으로 선택될 수 있다.
- [0065] 따라서, 프레임 또는 슬라이스를 인코딩하기 전에, 인코더는 이력 통계 데이터, 애플리케이션 타입에 대한 지식, 멀티 패스 분석, 또는 일부 다른 이러한 기술과 같은 요인들에 기초하여 움직임 벡터 해상도를 결정할 수 있다. 인코더가 단지 전체 화소 움직임 추정을 사용하기로 결정하면, 부화소 검색은 생략된다. 스케일링된 움직임 벡터 예측이 분수 부분을 가지면, 예측은 정수 값으로 반올림될 수 있다.
- [0066] 다른 실시예에서, 제어는 선택적으로 벡터의 수평 또는 수직 컴포넌트에 대해 별도로 적용될 수 있다. 이것은 수평으로 또는 수직으로 스케일링되는 스크린 비디오를 인코딩하기에 유용할 수 있다. 예를 들면, 인코더가 수평으로 또는 수직으로 크기가 조정된 스크린 공유와 협력하는 경우를 고려한다. 이 경우, 움직임 벡터의 컴포넌트들 중 하나는 분수 부분을 가질 것이지만, 다른 컴포넌트는 분수 부분을 갖지 않을 것이다. 비트 레이트를 제어하기 위해, 인코더는 미리 정의된 범위에서 양자화 파라미터(quantization parameter; QP)를 조정할 수 있다. QP가 허용된 상한계에 도달하고, 비트 레이트가 여전히 너무 높은 경우, 인코더는 내부 리사이저(resizer)를 트리거할 수 있다. 그것은 수평으로 입력 픽처를 크기 조정하고, 디코더 측에서, 디코딩된 픽처를 다시 원래의 폭으로 늘린다. 따라서, 수평 및 수직 움직임 벡터를 상이하게 처리하는데 또한 유용한 상황이 발생할 수 있다.
- [0067] 정수 화소 단위 또는 스텝의 감소된 해상도 스케일을 통해 움직임 벡터를 나타내어, 종래의 코덱에 비해 연관된 비트 레이트 절감을 달성하기 위해서, 움직임 벡터를 시그널링하기 위한 프로토콜은 미래 코덱 표준을 위해 업데이트되어야 할 것이다. 실시예에서, 이것은 H.265 (HEVC, 고효율 비디오 코딩) 표준에 대한 업데이트로서 구현될 수 있다. 캡처된 스크린 콘텐츠를 인코딩하는 경우, 인코딩된 데이터(34)의 포맷은 각각의 움직임 벡터에 대한 감소된 크기의 움직임 벡터 필드가 제공될 것이다. 정수 화소 모드로 인코딩된 스크린 캡처 스트림의 경우, 관련 데이터(34)는 비트 스트림(33)에 정수 움직임 벡터를 포함할 것이고, 일 실시예에서는 비트 스트림(33)에 오직 정수 움직임 벡터만을 포함할 것이다.
- [0068] 실시예에서, 이것은 선택적일 것이고, 부화소(예컨대, $\frac{1}{4}$ 화소) 또는 정수 화소 해상도가 관련 프레임 또는 슬라이스(다시 한 번, 도 3 참조)의 인코딩에 사용되는지 여부를 나타내기 위해 헤더(36)에 또한 포함된 플래그(37)를 이용한다. 수평 및 수직 해상도가 별도로 선택될 수 있다면, 두 개의 플래그(37)가 프레임 또는 슬라이스마다 요구될 것이다.
- [0069] 다음은 H.265 표준에 대한 업데이트에 기초하여 예시적인 실시예를 개시한다. 수정은 움직임 벡터가 인코딩된 비트 스트림(33)에서 감소된 정수 화소 스케일로 표현되도록 할 수 있고, 수평 및 수직 컴포넌트에서 움직임 벡터 해상도를 시그널링하기 위해서, 압축된 스트림의 헤더 정보(36)에 슬라이스마다 두 개의 플래그(37)를 추가한다.
- [0070] 수정은 헤더 레벨이 아닌 구문 또는 파싱(parsing) 프로세스를 변경하는 것이 아니라, 움직임 벡터 차이를 정수로 해석하고 스케일링된 MV 예측을 정수 값으로 반올림함으로써 디코딩 프로세스를 수정하는 것이다. 수정은 테스트되는 스크린 콘텐츠 시퀀스에 대해 7 % 정도까지 많이, 평균적으로 대략 2% 만큼, 코딩 효율을 증가시키는 것으로 밝혀졌고, 이것은 또한 인코딩 및 디코딩 프로세스의 복잡성을 감소시킬 수 있다.
- [0071] 움직임 벡터의 해석에 대한 해상도를 나타내기 위해 하이-레벨 지표가 (SPS, PPS 및/또는 슬라이스 헤더 레벨에) 추가된다.
- [0072] 디코딩 프로세스에서, 움직임 벡터가 전체 화소 해상도에 있는 것으로 나타나고, 스케일링된 움직임 벡터 예측이 분수 부분을 갖는 경우, 이 실시예에서, 예측은 정수 값으로 반올림된다. 움직임 벡터의 차이는 $\frac{1}{4}$ 샘플 오프셋보다는 정수 오프셋으로 간단히 해석된다. 모든 다른 디코딩 프로세스는 동일하게 유지된다. 파싱 프로세스

(헤더 레벨 아래)도 또한 변하지 않는다. 움직임 벡터가 전체 샘플 정밀도로 코딩되고 입력 이미지 데이터가 4:2:2 또는 4:2:0 샘플링을 사용하는 경우, 크로마 움직임 벡터는 $\frac{1}{2}$ 샘플 크로마 움직임 변위를 생성할 통상의 방식으로 유도될 수 있다. 대안적으로, 크로마 움직임 벡터는 또한 정수 값으로 반올림될 수 있다.

[0073] 움직임 벡터 해상도를 시그널링하기 위한 구문 변경에 관해서, 실시예에서는, 3 개의 가능한 모드가 있고, 모드 지표를 이용하여 픽처 파라미터 세트(picture parameter set; PPS)와 같은 높은 (프레임마다) 헤더 레벨에서, 또는 대안적으로 시퀀스 파라미터 세트(sequence parameter set; SPS)와 같은 더욱 높은 레벨에서 시그널링된다. 모드 지표는 3 개의 선택 가능한 값을 갖는다. 하나의 값은 움직임 벡터가 항상 $\frac{1}{4}$ 샘플 오프셋(PPS를 사용하는 픽처에서)을 나타내는 것으로 해석되고 있다는 것을 나타낸다. 제 2 값은 움직임 벡터가 항상 정수 오프셋(PPS를 사용하는 픽처에서)을 나타내는 것으로 해석되고 있다는 것을 나타낸다. 제 3 값은 낮은 레벨에서 적용될 해석을 적응적으로 선택하는 낮은 레벨의 신호가 있다는 것을 나타낸다. 이러한 낮은 레벨 표시는 이 제 3 모드에서 동작하는 경우에만 존재할 슬라이스 헤더의 플래그에 있을 수 있다.

[0074] 이를 구현하기 위해, 실시예에서, 새로운 2 비트 지표가 움직임 벡터 해상도 제어 모드를 나타내기 위해 PPS 확장에 포함될 것이다. 이 지표는 motion_vector_resolution_control_idc로 지칭될 수 있다. 이 모드가 0인 경우, 움직임 벡터는 $\frac{1}{4}$ 화소 정밀도로 인코딩되고, 모든 디코딩 프로세스는 변하지 않은 상태로 유지된다. 이 모드가 1인 경우, PPS를 나타내는 슬라이스의 모든 움직임 벡터는 전체 화소 정밀도로 인코딩된다. 그리고, 이 모드가 2인 경우, 움직임 벡터 해상도는 슬라이스 헤더의 플래그에 의해 슬라이스 단위로 제어된다. motion_vector_resolution_control_idc가 존재하지 않으면, 이 값은 0으로 추정된다. motion_vector_resolution_control_idc가 2와 같으면, slice_motion_vector_resolution_flag로 불리는 추가의 플래그가 슬라이스 헤더에 시그널링된다. 이 플래그가 0이면, 이 슬라이스의 움직임 벡터는 $\frac{1}{4}$ 화소 정밀도로 인코딩되고, 이 플래그가 1이면, 움직임 벡터는 전체 화소 정밀도로 인코딩된다. 이 플래그가 존재하지 않으면, 이 값은 motion_vector_resolution_control_idc의 값과 동일한 것으로 추정된다.

[0075] 수정된 PPS 구문은 다음과 같이 나타난다:

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
dependent_slice_segments_enabled_flag	u(1)
output_flag_present_flag	u(1)
num_extra_slice_header_bits	u(3)
sign_data_hiding_enabled_flag	u(1)
cabac_init_present_flag	u(1)
...	
lists_modification_present_flag	u(1)
log2_parallel_merge_level_minus2	ue(v)
slice_segment_header_extension_present_flag	u(1)
pps_extension1_flag	u(1)
if(pps_extension1_flag) {	
if(transform_skip_enabled_flag)	
log2_max_transform_skip_block_size_minus2	ue(v)
luma_chroma_prediction_enabled_flag	u(1)
motion_vector_resolution_control_idc	u(2)
chroma_qp_adjustment_enabled_flag	u(1)
if(chroma_qp_adjustment_enabled_flag) {	
diff_cu_chroma_qp_adjustment_depth	ue(v)
chroma_qp_adjustment_table_size_minus1	ue(v)
for(i = 0; i <= chroma_qp_adjustment_table_size_minus1; i++) {	
cb_qp_adjustment[i]	se(v)
cr_qp_adjustment[i]	se(v)
}	
}	
pps_extension2_flag	u(1)
}	
if(pps_extension2_flag)	

[0076]

while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbps_trailing_bits()	
}	

[0077]

[0078] 수정된 슬라이스 헤더 구문은 다음과 같이 나타난다:

slice_segment_header() {	Descriptor
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
...	
if(slice_type == P slice_type == B) {	
if(motion_vector_resolution_control_idc == 2)	
slice_motion_vector_resolution_flag	u(1)
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
...	

[0079]

[0080]

앞서 언급된 스케일링에 관해서, 이것은, 예를 들면, HEVC(H. 265)에서 발생할 수 있는 것이다. 아이디어는, 움직임 벡터가 일부 다른 프레임에 코딩하는데 사용되는 경우, 이것은 (i) 현재 픽처와 (ii) 그 기준 픽처 간의 상대 위치 변위 면에서 동등한 움직임 벡터가 무엇인지 계산될 수 있다는 것이다. 이것은 다른 픽처의 동일 위치 부분에서의 움직임 벡터에 의해 나타난 변위의 상대 위치에 기초하고, (iii) 해당 픽처와 (iv) 기준 픽처로서 참조되는 픽처 간의 상대 위치 변위에 기초한다. 즉, 움직임 벡터 자체는 예측에 기초하여 인코딩될 수 있고, 현재 타겟 프레임의 타겟 블록에 대한 움직임 벡터는 이전 프레임의 공간적으로 대응하는 블록에 대한 이전의 움직임 벡터에 관하여 인코딩된다. 그런 다음, 타겟 블록에 대한 움직임 벡터는 예측된 버전과 실제 버전 간의 차이(예컨대, 통상적으로 델타) 면에서 오직 인코딩된 비트 스트림(33)에 시그널링된다. 그런 다음, 디코더는 디코더 측에서 동일한 예측을 수행하여 상기 델타를 추가함으로써 인코딩된 움직임 벡터를 재생성할 수 있다.

[0081]

그러나, 코딩된 데이터의 시간적 프레임 레이트는 항상 일정하지 않고, 또한 픽처가 비트 스트림에 코딩되는 순서와 픽처가 캡처되고 디스플레이되는 순서 간의 차이가 있을 수 있다. 이러한 시간적 관계는 계산될 수 있고, 그런 다음, 움직임 벡터가 기본적으로 동일한 방향에서 동일한 움직임 속도를 나타내도록 움직임 벡터를 스케일링하기 위해 사용될 수 있다. 즉, 움직임 벡터의 예측 버전은 기준 벡터와 정확히 동일하지 않고, 오히려 그것의 스케일링된 버전이다. 이것은 시간적 움직임 벡터 예측으로 알려져있다.

[0082]

HEVC 표준에서 현재 움직임 벡터(motion vector; MV) 디코딩 프로세스는 다음과 같이 요약될 수 있다.

[0083]

1. MV가 디코더에서 예측되는 방법 및 움직임 벡터 델타(motion vector delta; MVD)가 인코더로부터 인코딩된 비트 스트림(이것은 일부 구문 표시를 포함할 수 있다)에 전송되었는지의 여부를 결정한다.

[0084]

2. 예측된 움직임 벡터(predicted motion vector; PMV)를 생성한다. 이것은 한 쌍의 정수(px, py)를 생성한다. 이 정수는 루마 도메인에서 1/4 샘플 위치 정밀도로 오프셋을 나타내는 것으로 가정된다.

[0085]

3. 어떠한 MVD도 전송되지 않는 경우, MV 값(mx, my)를 (px, py)로 설정하고, 그렇지 않으면, 한 쌍의 정수(px, py)를 (dx, dy)로서 디코딩하고, MV 값(mx, my)를 (px + dx, py + dy)로 설정한다. MV는 루마 도메인에서 1/4 샘플 오프셋을 나타내는 것으로 해석된다. 주의: 비디오가 크로마 채널에서보다 루마 채널에서의 상이한 해상도를 이용하는 4:2:2 또는 4:2:0 와 같은 포맷인 경우, 루마 구별만 중요하고, 대신에 비디오가, 예를 들어, 4:4:4 포맷인 경우, 오프셋은 모든 샘플링 그리드에 대해 1/4 샘플 단위를 갖는 것으로 해석된다.

[0086]

4. 그 다음의 PMV를 생성(그리고 디블록킹 필터링을 제어하는 것 등)하기 위한 다음의 사용을 위해 최종 값(mx, my)을 저장한다.

[0087]

본 개시의 실시예에서, 정수 화소 움직임 벡터 모드가 사용되는 경우, 스케일링된 움직임 벡터는 정수 화소 해상도에 대한 제한을 수용하기 위해 디코딩 프로세스에서 반올림될 수 있다. 예를 들어, 움직임 벡터 디코딩 프

로세스는 다음과 같이 수정될 수 있다.

- [0088] 1. MV가 디코더에서 예측되는 방법 및 움직임 벡터 델타(motion vector delta; MVD)가 인코더로부터 인코딩된 비트 스트림(이것은 일부 구문 표시를 포함할 수 있다)에 전송되었는지의 여부를 결정한다.
- [0089] 2. 예측된 움직임 벡터(predicted motion vector; PMV)를 생성한다. 이것은 한 쌍의 정수(px, py)를 생성한다. 이 정수는 루마 도메인에서 1/4 샘플 위치 정밀도로 오프셋을 나타내는 것으로 가정된다.
- [0090] 3. 정수 정밀도 모드에서 동작하는 경우, 다음을 수행한다.
- [0091] a. 예측 값을 정수 오프셋에 대응하도록 만들기 위해 예측 값을 반올림한다 - 예컨대, $(rx, ry) = (\text{round}(px/4), \text{round}(py/4))$ 생성한다.
- [0092] b. 어떠한 MVD도 전송되지 않는 경우, MV 값(mx, my)를 $(4*rx, 4*ry)$ 로 설정하고, 그렇지 않으면, 한 쌍의 정수(px, py)를 (dx, dy)로서 디코딩하고, MV 값(mx, my)를 $(4*(rx + dx), 4*(ry + dy))$ 로 설정한다. 그런 다음, MV는 루마 도메인에서 1/4 샘플 오프셋을 나타내는 것으로 해석된다(주의, 또다시, 비디오가 4:2:2 또는 4:2:0와 같은 포맷인 경우, 루마 구별만 중요하다).
- [0093] 4. 그렇지 않으면(즉, 1/4 샘플 정밀도 모드에서 동작하는 경우), 다음을 수행한다. 어떠한 MVD도 전송되지 않는 경우, MV 값(mx, my)를 (px, py)로 설정하고, 그렇지 않으면, 한 쌍의 정수(px, py)를 (dx, dy)로서 디코딩하고, MV 값(mx, my)를 $(px + dx, py + dy)$ 로 설정한다. MV는 루마 도메인에서 1/4 샘플 오프셋을 나타내는 것으로 또다시 해석된다.
- [0094] 5. 그 다음의 PMV를 생성(그리고 디블록킹 필터링을 제어하는 것 등)하기 위한 다음의 사용을 위해 최종 값(mx, my)을 저장한다.
- [0095] 그러나, 상기 프로세스는 모든 가능한 실시예들에 필요하지 않다. 위에서 설명한 방법은 움직임 벡터 예측의 스케일링이 1/4 화소 도메인에서 발생하도록 디코더가 처음부터 1/4 화소 도메인에서 동작하는 것을 가정한다. 그러나, 사실, 디코더는 이 스테이지에서 단위가 무엇을 의미하는지를 반드시 알아야 할 필요는 없다. 따라서, 실제로 이미지를 생성하기 위해 사용될 나중까지 단위에 어떠한 해석도 하지 않는 대안적인 방법을 취하는 것이 가능하다.
- [0096] 다음은 이러한 대안적인 방법의 실시예에 따른 디코딩 프로세스를 설명한다. 위와 같이, 움직임 벡터 값 중 적어도 하나는 상이하게 인코딩될 수 있고, 움직임 예측 모듈(73)은 상기 움직임 벡터 값 중 다른 값을 스케일링함으로써 움직임 벡터 값을 예측하도록 구성된다. 움직임 벡터 값 중 적어도 하나는 델타 값의 측면으로 상이하게 인코딩될 수 있고, 움직임 예측 모듈(73)은 스케일링된 움직임 벡터 값을 생성하도록 상기 움직임 벡터 값 중 다른 값을 스케일링하고, 그런 다음 상기 스케일링된 움직임 벡터 값에 델타 값을 가산함으로써, 움직임 벡터 값을 예측하도록 구성될 수 있다. 그러나, 상기와는 달리, 움직임 예측 모듈(73)은 또한, 스케일링 이후에 정수 화소 해상도로 움직임 벡터 값을 반올림하지 않고, 상기 정수 화소 단위(정수 모드가 사용되는 경우)로 움직임 벡터 값을 해석하도록 구성된다. 이것은 스케일링의 단위에 해석을 하는 것을 삼가함으로써 가능하게 된다. 움직임 예측 모듈(73)은 정수 화소 해상도로 스케일링된 움직임 벡터 값을 반올림하지 않고 스케일링된 움직임 벡터 값에 델타 값의 가산을 수행함으로써, 상기 정수 화소 단위로 움직임 벡터 값을 해석하도록 구성될 수 있다. 결과적인 움직임 벡터 값은 이미지를 생성하기 위해 사용될 때만 오직 정수 또는 부화소 단위 측면으로 해석된다. 결과 값은 또한 저장된 값에 대한 해석을 정수 단위 또는 분수 단위 중 어느 하나의 측면으로 하지 않고, 하나 이상의 그 다음의 움직임 벡터를 예측하는데 사용하기 위해 저장된다.
- [0097] 예를 들어, 이와 같은 대안에 따라, 움직임 벡터 디코딩 프로세스는 다음과 같이 구현될 수 있다.
- [0098] 1. MV가 디코더에서 예측될 방법 및 움직임 벡터 델타(motion vector delta; MVD)가 인코더로부터 인코딩된 비트 스트림(이것은 일부 구문 표시를 포함할 수 있다)에 전송되었는지의 여부를 결정한다.
- [0099] 2. 예측된 움직임 벡터(predicted motion vector; PMV)를 생성한다. 이것은 한 쌍의 정수(px, py)를 생성한다. 그러나, 이것이 나타내는 것에 대해 걱정하지 않는다. 즉, 이 스테이지에서 단위에 대해 어떠한 해석도 하지 않는다.
- [0100] 3. 어떠한 MVD도 전송되지 않는 경우, MV 값(mx, my)를 (px, py)로 설정하고, 그렇지 않으면, 한 쌍의 정수(px, py)를 (dx, dy)로서 디코딩하고, MV 값(mx, my)를 $(px + dx, py + dy)$ 로 설정한다. 정수 정밀도 모드에서 동작하는 경우, 루마 도메인에서 전체 정수 오프셋을 나타내는 것으로서, 즉, 1/4 샘플 오프셋 단위로 값

($4 \times mx$, $4 \times my$)을 나타내는 것으로서 MV를 해석한다. 그렇지 않으면(즉, $1/4$ 샘플 정밀도 모드에서 동작하는 경우), 루마 도메인에서 $1/4$ 샘플 오프셋을 나타내는 것으로서, 즉, $1/4$ 샘플 오프셋 단위로 값(mx , my)을 나타내는 것으로서 MV를 해석한다(주의, 또다시, 비디오가 4:2:2 또는 4:2:0 와 같은 포맷인 경우, 루마 구별만 중요하다).

- [0101] 4. 그 다음의 PMV를 생성(그리고 디블록킹 필터링을 제어하는 것 등)하기 위한 다음의 사용을 위해 최종 값(mx , my)을 저장한다. 또다시, 이것이 나타내는 것에 대해 걱정하지 않는다.
- [0102] 따라서, 이전에 설명한 반올림 단계(3a)는 이제 제거된다. 오직 숫자의 해석만이 수정된다. 이것은 MV 예측 프로세스의 어떠한 수정도 불필요하기 때문에, 디코딩 프로세스를 덜 복잡하게 만든다.
- [0103] 위에서 설명한 반올림은 $1/4$ (또는 분수) 해상도에서 정수 해상도로의 반올림을 나타낸다. 본 표준에는, 또한 움직임 벡터 예측의 일부로서 더욱 미세한 해상도에서 $1/4$ 해상도로의 반올림도 있고, 위의 대안적인 실시예에서, 예비 반올림이 여전히 존재하는 것을 배제하지 않는다.
- [0104] 이와 같이 동작하는 것은 일부 부작용이 있을 수 있다. 하나의 부작용은 일부 픽처가 다른 것과는 상이한 모드를 사용하는 경우, 시간적 MV 예측을 위한 MV 예측 프로세스에서 사용되는 스케일링이 정확하지 않을 수 있다는 것이다. 그러나, 이것은 매우 작은 효과일 것이다. 특히, 상이한 모드에서 동작하는 이러한 상황을 갖는 것이 매우 드물기 때문에, 또한 시간적 MV 예측이 일반적으로 단지 매우 작은 장점만을 가지고 있기 때문이다. 다른 부작용은 디블록킹 필터 제어의 결과가 MV 값에 의존할 수 있다는 것이고, MV 값은 상이한 스케일 인자로 해석된다. 그러나, 이 효과도 또한 매우 작고, 특히, 디블록킹이 사용되지 않거나, 또는 스크린 콘텐츠를 코딩할 때 많은 효과를 갖지 않을 수 있기 때문이다.
- [0105] 또 다른 가능성은 정수 움직임만을 사용할 때마다 시간적 움직임 벡터 예측을 사용하지 않는 것이다. 인코더가 그 기능의 사용을 사용하지 않게 하는 구문이 HEVC에 이미 있다. 그것은 차이가 정수 값 또는 분수 값으로서 코딩되는지의 여부에 따라 상이하게 동작하는 특별한 프로세스를 디코더가 가져야 하는 필요성을 방지하기 위한 가능한 방법일 것이다. 그래도 이러한 사용 경우에 시간적 움직임 벡터 예측에서 얻은 이득은 작을 수 있어(또는 0일 수 있음), 그것을 사용하지 않는 것은 바람직하지 않아도 된다.
- [0106] 상기 실시예들은 단지 예로서 설명되었다는 것을 이해할 것이다.
- [0107] 예를 들어, 상기에서 블록 면에서 설명되었지만, 이것은 반드시 어떤 특정 표준에서 블록이라고 언급되는 분할로 한정되는 것은 아니다. 예를 들어, 본원에 나타난 블록은 H.26x 표준에서의 블록 또는 매크로 블록이라고 언급되는 분할일 수 있다.
- [0108] 본원에 개시된 일반적인 기술 및 임의의 특정 코덱 또는 표준으로 한정된 본 개시의 범위는 기존 표준의 컨텍스트 또는 기존 표준에 대한 업데이트 중 어느 하나에서 구현될 수 있고, H264 또는 H.265와 같은 H.26x 표준이든 임의의 다른 표준이든 맞춤형 코덱으로 구현될 수 있다. 또한, 본 개시의 범위는 RGB, YUV 측면으로든 또는 다른 측면으로든 비디오 샘플의 임의의 특정 표현으로 특별히 제한되지 않는다. 본 개시의 범위는 임의의 특정 양자화로도 한정되지 않고, DCT 변환으로도 한정되지 않는다. 예컨대, 카루넬 루베 변환(KLT)과 같은 대안적인 변환이 사용되거나, 어떠한 변환도 사용되지 않을 수 있다. 또한, 본 개시는 임의의 특정 종류의 네트워크를 통한 통신 또는 VoIP 통신으로 한정되는 것이 아니라, 데이터를 전달할 수 있는 임의의 네트워크 또는 매체에 사용될 수 있다.
- [0109] 움직임 벡터 오프셋이 화소의 정수 등으로 제한되거나 제한되지 않는 경우, 이것은 움직임 추정을 색 공간 채널 중 임의의 하나 또는 두 개로 나타내거나, 움직임 추정을 세 개의 색 채널 모두로 나타낼 수 있다.
- [0110] 또한, 본 발명은 정수 화소 해상도와 $1/4$ 화소 해상도 간의 선택으로 한정되지 않는다. 일반적으로, 본원에서 설명된 기술은 정수 화소 해상도와 임의의 부화소 해상도, 예컨대, $1/2$ 화소 해상도 간의 선택, 또는 정수 화소 해상도와 복수의 상이한 부화소 모드 간의 선택, 예컨대, 정수와, $1/2$ 모드와, $1/4$ 모드 간의 선택에 적용될 수 있다.
- [0111] 또한, 본 개시의 범위는 인코딩된 비디오 및/또는 스크린 캡처 스트림이 네트워크를 통해 전송되는 애플리케이션으로도 한정되지 않고, 스트림이 라이브 스트림인 애플리케이션으로도 한정되지 않는다. 예를 들어, 다른 애플리케이션에서, 스트림은 광 디스크, 하드 디스크 또는 다른 자기 저장 장치, 또는 "플레이스" 메모리 스틱 또는 다른 전자 메모리와 같은 저장 디바이스에 저장될 수 있다. 그러므로, 스크린 공유 스트림은 반드시 라이브 공유를 의미해야 하는 것은 아님을 주의한다(그래도 그것은 확실히 하나의 옵션이기는 하다). 대안적으로 또는 부

가적으로, 그것은 하나 이상의 다른 사용자와 나중에 공유하기 위해 저장되거나, 캡처된 이미지 데이터는 공유되는 것이 아니라, 오히려 단지 그 시점에 스크린을 이용하는 사용자에게 기록될 수 있다. 일반적으로, 스크린 캡처는 캡처된 인코더 측 스크린 콘텐츠로 구성된 임의의 움직이는 이미지 데이터일 수 있고, 한 명 이상의 다른 사용자와 공유(라이브이든 아니든)하기 위해 또는 캡처 사용자의 이익을 위해 단순히 기록하기 위해 또는 단지 보관(아마도, 결과적으로 봤을 때 실제로 결코 다시 볼 수 없음)을 위해 임의의 적합한 수단(그것은 하나의 옵션이기는 하지만, 반드시 스크린 버퍼로부터의 획득에 의한 것은 아니다)에 의해 캡처될 수 있다.

[0112] 코덱은 반드시 단지 스크린 캡처 데이터 및 비디오를 인코딩하는 것으로 한정되는 것은 아님을 또한 주의한다. 실시예에서, 다른 타입의 움직이는 이미지 데이터, 예컨대, 애니메이션을 또한 인코딩할 수 있다. 이러한 다른 타입의 움직이는 이미지 데이터는 부화소 모드 또는 정수 화소 모드로 인코딩될 수 있다.

[0113] 또한, 인터 프레임 인코딩은 반드시 항상 이전 프레임에 관하여 인코딩되어야 하는 것은 아니고, 보다 일반적인 일부 코덱들은 타겟 프레임이 아닌, 타겟 프레임 이전의 또는 타겟 프레임 앞에 있는 상이한 프레임에 관한 인코딩을 허용할 수 있다(적합한 발송 버퍼를 가정함)는 것을 주의한다.

[0114] 또한, 전술한 바와 같이, 움직임 벡터 자체가 상이하게 인코딩될 수 있다는 것을 주의한다. 인코딩된 비트 스트림에서 시그널링되는 움직임 벡터가 화소의 정수 등으로 제한되는 경우, 이것은 상이하게 인코딩된 형태의 움직임 벡터가 제한(예컨대, 텔타)된다는 것을 의미한다.

[0115] 또한, 디코더는 반드시 최종 사용자 단말기에 구현되어야 하는 것도 아니고, 수신 단말기에서 즉각적인 소비를 위해 움직이는 이미지 데이터를 출력해야 하는 것도 아니다. 대안적인 실시예에서, 수신 단말기는, 디코딩되거나 변환된 형태의 움직이는 이미지 데이터를 다른 단말기에 출력하기 위해, 또는 나중의 소비를 위해 디코딩된 데이터를 저장하기 위해, 디코더 소프트웨어를 실행하는 서버와 같은 중간 단말기일 수 있다. 유사하게, 인코더는 최종 사용자 단말기에 구현되어야 하는 것도 아니고, 송신 단말기에서 비롯되는 움직이는 이미지 데이터를 인코딩해야 하는 것도 아니다. 다른 실시예에서, 송신 단말기는, 예를 들어, 비인코딩되거나 다르게 코딩된 형태의 움직이는 이미지 데이터를 다른 단말기로부터 수신하기 위해, 그 데이터를 서버에 저장하기 위해 인코딩하거나 트랜스코딩하기 위해, 또는 수신 단말기에 포워딩하기 위해, 인코더 소프트웨어를 실행하는 서버와 같은 중간 단말기일 수 있다.

[0116] 일반적으로, 본 명세서에 기술된 기능들 중 임의의 기능들은, 소프트웨어, 펌웨어, 하드웨어(예컨대, 고정 로직 회로), 또는 이러한 구현예들의 조합을 사용하여 구현될 수 있다. 본 명세서에서 이용되는 바와 같은 용어 "모듈", "기능", "컴포넌트" 및 "로직"은 일반적으로 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합을 나타낸다. 소프트웨어 구현예의 경우에, 모듈, 기능, 또는 로직은 프로세서(예컨대, CPU 또는 CPU들) 상에서 실행될 때 명시된 작업을 수행하는 프로그램 코드를 나타낸다. 프로그램 코드는 하나 이상의 컴퓨터 판독 가능 메모리 디바이스에 저장될 수 있다. 본 명세서에 기술된 기술들의 특징은 플랫폼 독립적이고, 이것은 기술들이 다양한 프로세서를 갖는 다양한 상업용 컴퓨팅 플랫폼 상에 구현될 수 있다는 것을 의미한다.

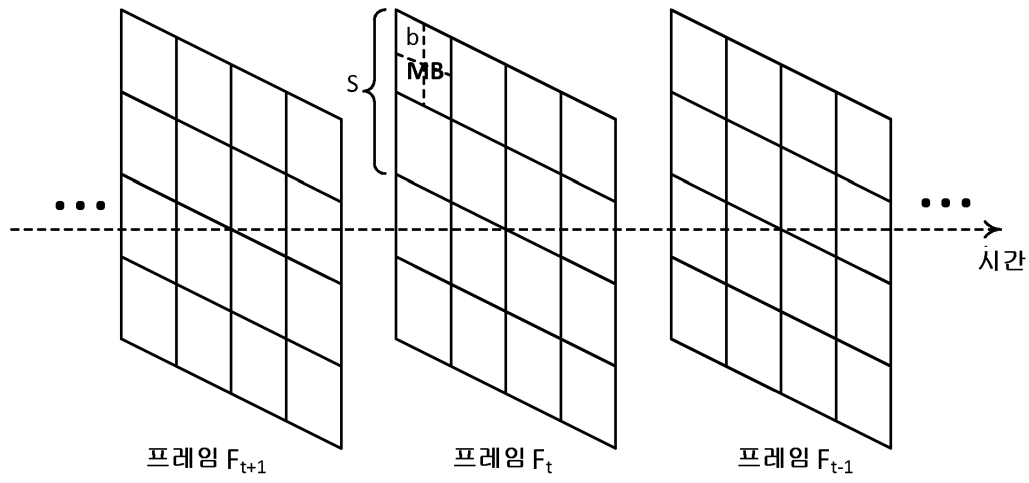
[0117] 예를 들어, 단말기는 사용자 단말기의 하드웨어로 하여금 동작들, 예컨대, 프로세서 기능 블록들 등을 수행하게 하는 엔티티(예컨대, 소프트웨어)를 포함할 수 있다. 예를 들어, 단말기는 사용자 단말기, 보다 상세하게는, 사용자 단말기의 운영 체제 및 연관된 하드웨어로 하여금 동작들을 수행하게 하는 명령어들을 유지하도록 구성될 수 있는 컴퓨터 판독 가능 매체를 포함할 수 있다. 이와 같이, 명령어들은 운영 체제 및 연관된 하드웨어를 동작들을 수행하도록 구성하는 기능을 하고, 이러한 방식으로 운영 체제 및 연관된 하드웨어를 기능들을 수행하도록 변환시킨다. 명령어들은 다양한 상이한 구성들을 통해 컴퓨터 판독 가능 매체에 의해 단말기에 제공될 수 있다.

[0118] 컴퓨터 판독 가능 매체의 하나의 이러한 구성은 신호 전달 매체(signal bearing medium)이고, 따라서 네트워크를 통하는 등에 의해, 명령어들을 (예컨대, 반송파로서) 컴퓨팅 디바이스로 전송하도록 구성된다. 컴퓨터 판독 가능 매체는 또한 컴퓨터 판독 가능 저장 매체로서 구성될 수 있고, 따라서 신호 전달 매체가 아니다. 컴퓨터 판독 가능 저장 매체의 예는 랜덤 액세스 메모리(random-access memory; RAM), 읽기 전용 메모리(read-only memory; ROM), 광 디스크, 플래시 메모리, 하드 디스크 메모리, 그리고 명령어들 및 다른 데이터를 저장하기 위해 자기, 광, 및 기타 기법들을 사용할 수 있는 다른 메모리 디바이스들을 포함한다.

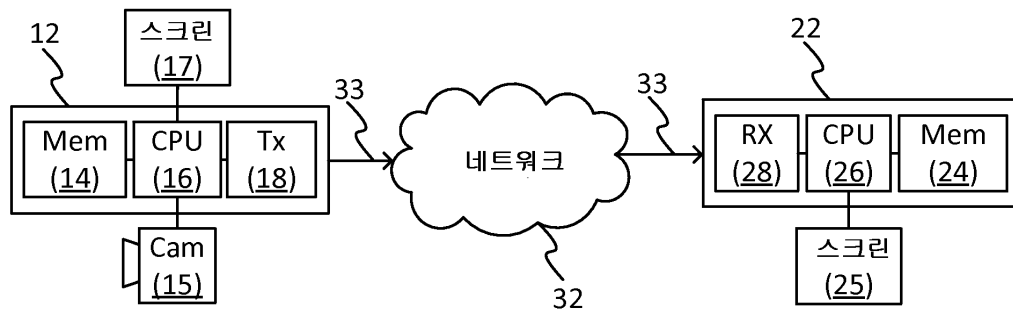
[0119] 대상이 구조적 특징 및/또는 방법론적 행위에 특정한 언어로 설명되어 있지만, 첨부된 특허 청구 범위에 정의된 대상은 반드시 위에서 설명한 특정한 특징 또는 행위로 제한되는 것이 아님이 이해될 것이다. 오히려, 위에서 설명한 특정한 특징 및 행위는 특허 청구 범위를 구현하는 예시적인 형태로 개시된다.

도면

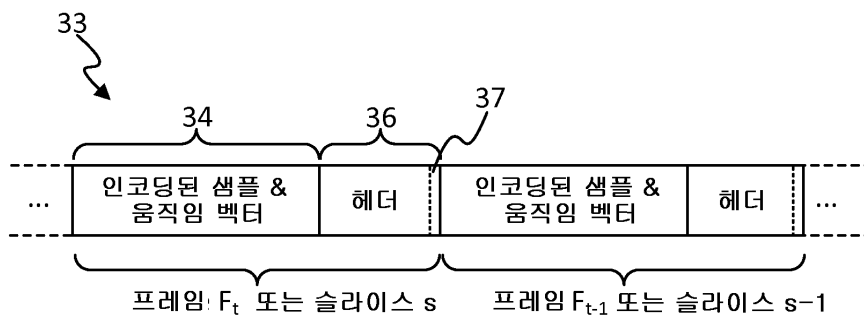
도면1



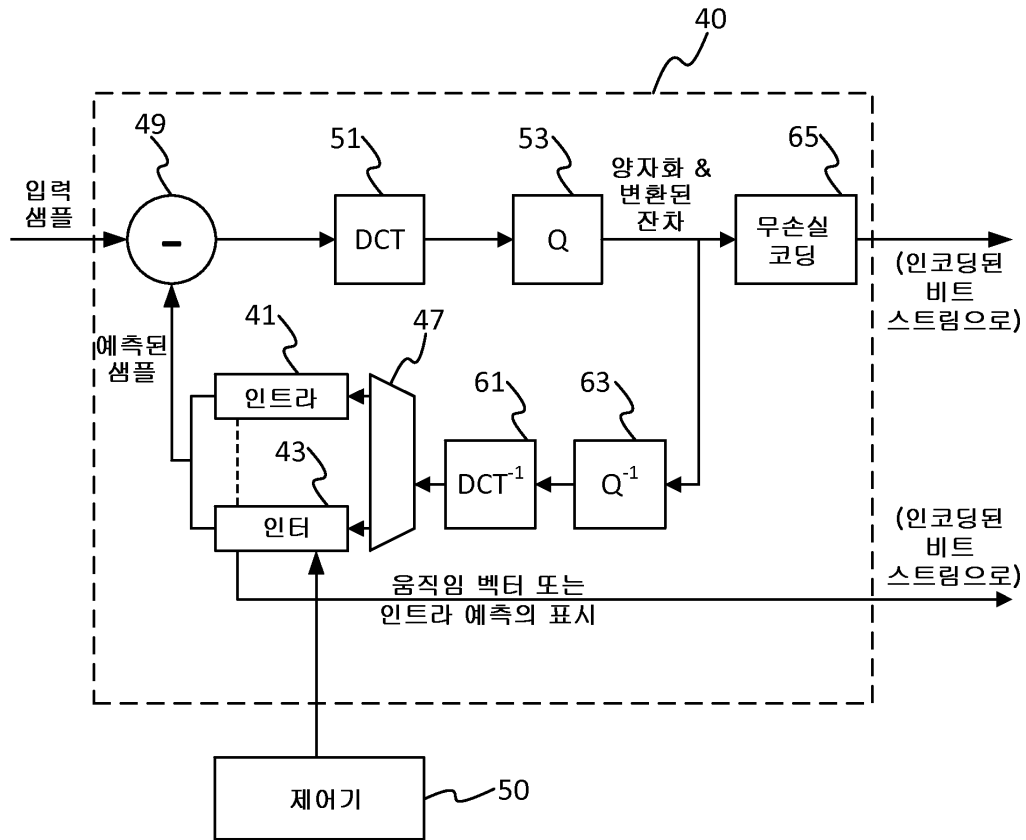
도면2



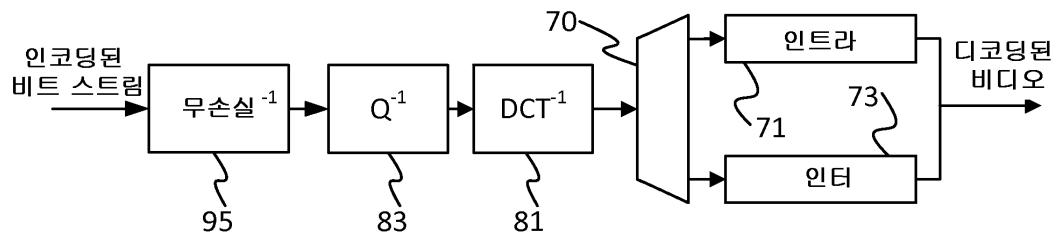
도면3



도면4



도면5



도면6

