



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년11월02일
(11) 등록번호 10-1792041
(24) 등록일자 2017년10월25일

- (51) 국제특허분류(Int. Cl.)
H04N 19/147 (2014.01) H04N 19/51 (2014.01)
- (21) 출원번호 10-2012-7012382
- (22) 출원일자(국제) 2010년10월04일
심사청구일자 2015년10월05일
- (85) 번역문제출일자 2012년05월14일
- (65) 공개번호 10-2012-0093288
- (43) 공개일자 2012년08월22일
- (86) 국제출원번호 PCT/US2010/002670
- (87) 국제공개번호 WO 2011/046587
국제공개일자 2011년04월21일
- (30) 우선권주장
61/251,508 2009년10월14일 미국(US)
- (56) 선행기술조사문헌
JP2004048552 A*
(뒷면에 계속)
- (73) 특허권자
톱슨 라이센싱
프랑스 92130 이씨레블리노 잔 다르크 뢰 1-5
- (72) 발명자
구오, 리웨이
미국 08536 뉴저지주 플레인스보로 미들섹스 블러
바드 에이퍼티. 311 100
쟁, 윤페이
미국 08536 뉴저지주 플레인스보로 웨일 리지 드
라이브 123
(뒷면에 계속)
- (74) 대리인
양영준, 전경석, 백만기

전체 청구항 수 : 총 11 항

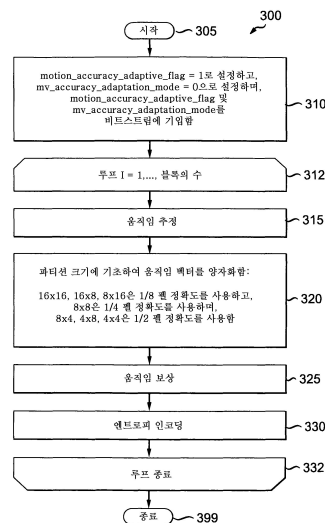
심사관 : 이상래

(54) 발명의 명칭 움직임 정보의 적응적 코딩 방법 및 장치

(57) 요약

움직임 정보의 적응적 코딩 방법 및 장치가 제공된다. 장치는 움직임 벡터를 사용하여 적어도 화상에서의 블록을 인코딩하는 인코더(100)를 포함한다. 적응적 움직임 벡터 정확도 방식은 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준을 포함한다.

대표도 - 도3



(72) 발명자

인, 팽

미국 14850 뉴욕주 이타카 존 스트리트 6

솔, 호엘

미국 08542 뉴저지주 프린스턴 1/2 리 애비뉴 94

루, 시아오안

미국 08540 뉴저지주 프린스턴 케네디 코트 30

수, 치안

미국 95762 캘리포니아주 엘 도라도 힐스 컨커디아
드라이브 4795

(56) 선행기술조사문헌

KR1020040075317 A*

JP2004328799 A*

JP2006313950 A*

W02009088038 A1*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

움직임 벡터를 사용하여 화상 내의 블록을 인코딩하는 인코더(100)

를 포함하고,

상기 블록을 인코딩하는 데 사용되는 상기 움직임 벡터의 정확도를 선택하는 데 적응적 움직임 벡터 정확도 방식(adaptive motion vector accuracy scheme)이 사용되고, 상기 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 움직임 벡터 방향 적응에 기초하고,

상기 선택 기준은 움직임 벡터 성분 방향을 포함하고,

상기 선택 기준은 이웃한 블록들의 움직임 벡터들에 기초하는 움직임 벡터 성분 방향을 포함하고, 상기 블록을 인코딩하는데 사용되는 상기 움직임 벡터의 정확도는 상기 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하게 되도록 선택되며, 상기 수직 성분 및 상기 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분(dominant component)(520)으로서 선택되는 장치.

청구항 2

비디오 인코더에서의 방법으로서,

움직임 벡터를 사용하여 화상 내의 블록을 인코딩하는 단계

를 포함하고,

상기 블록을 인코딩하는 데 사용되는 상기 움직임 벡터의 정확도를 선택하는 데 적응적 움직임 벡터 정확도 방식이 사용되고, 상기 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 움직임 벡터 방향 적응에 기초하고,

상기 선택 기준은 움직임 벡터 성분 방향을 포함하고,

상기 선택 기준은 이웃한 블록들의 움직임 벡터들에 기초하는 움직임 벡터 성분 방향을 포함하고, 상기 블록을 인코딩하는데 사용되는 상기 움직임 벡터의 정확도는 상기 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하게 되도록 선택되며, 상기 수직 성분 및 상기 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분으로서 선택되는 방법.

청구항 3

삭제

청구항 4

삭제

청구항 5

제2항에 있어서, 상기 우세 성분은,

(i) 상기 움직임 벡터가 정수 움직임 벡터일 때, 상기 움직임 벡터의 진폭(520),

(ii) 상기 블록에 대한 움직임 보상 파티션의 형상(520),

(iii) 상기 블록에 대한 예측된 움직임 벡터(520),

(iv) 상기 블록과 관련한 이웃 블록들의 움직임 벡터(520), 및

(v) 상기 화상 및 하나 이상의 다른 화상 중 적어도 하나의 화상에 관한 전역(global) 움직임 정보(520) - 상기 화상 및 상기 하나 이상의 다른 화상은 동일한 비디오 시퀀스에 포함되어 있음 -

중 적어도 하나에 따라 결정되는 것인 방법.

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

제2항에 있어서, 상기 블록을 인코딩하는 데 사용된 상기 움직임 벡터의 정확도는 인코딩된 비트스트림으로 명시적으로 신호되는 것인 방법.

청구항 11

제2항에 있어서, 상기 블록을 인코딩하는 데 사용된 상기 움직임 벡터의 정확도는 상기 화상에서의 또는 상기 화상을 포함하는 시퀀스에서의 이전에 디코딩된 비디오로부터 추론(1120)되는 것인 방법.

청구항 12

움직임 벡터를 사용하여 화상 내의 블록을 디코딩하는 디코더(200)

를 포함하고,

상기 블록을 디코딩하는 데 사용되는 상기 움직임 벡터의 정확도를 선택하는 데 적응적 움직임 벡터 정확도 방식이 사용되고, 상기 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 움직임 벡터 방향 적응에 기초하고,

상기 선택 기준은 움직임 벡터 성분 방향을 포함하고,

상기 선택 기준은 이웃한 블록들의 움직임 벡터들에 기초하는 움직임 벡터 성분 방향을 포함하고, 상기 블록을 디코딩하는데 사용되는 상기 움직임 벡터의 정확도는 상기 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하게 되도록 선택되며, 상기 수직 성분 및 상기 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분으로서 선택되는 장치.

청구항 13

비디오 디코더에서의 방법으로서,

움직임 벡터를 사용하여 화상 내의 블록을 디코딩하는 단계

를 포함하고,

상기 블록을 디코딩하는 데 사용되는 상기 움직임 벡터의 정확도를 선택하는 데 적응적 움직임 벡터 정확도 방식이 사용되고, 상기 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 움직임 벡터 방향 적응에 기초하고,

상기 선택 기준은 움직임 벡터 성분 방향을 포함하고,

상기 선택 기준은 이웃한 블록들의 움직임 벡터들에 기초하는 움직임 벡터 성분 방향을 포함하고, 상기 블록을 디코딩하는데 사용되는 상기 움직임 벡터의 정확도는 상기 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하게 되도록 선택되며, 상기 수직 성분 및 상기 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분으로서 선택되는 것을 특징으로 하는 방법.

청구항 14

삭제

청구항 15

삭제

청구항 16

제13항에 있어서, 상기 우세 성분은,

(i) 상기 움직임 벡터가 정수 움직임 벡터일 때, 상기 움직임 벡터의 진폭(620),

(ii) 상기 블록에 대한 움직임 보상 파티션의 형상(620),

(iii) 상기 블록에 대한 예측된 움직임 벡터(620),

(iv) 상기 블록과 관련한 이웃 블록들의 움직임 벡터(620), 및

(v) 상기 화상 및 하나 이상의 다른 화상 중 적어도 하나의 화상에 관한 전역 움직임 정보(620) - 상기 화상 및 상기 하나 이상의 다른 화상은 동일한 비디오 시퀀스에 포함되어 있음 -

중 적어도 하나에 따라 결정되는 것인 방법.

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

제13항에 있어서, 상기 블록을 디코딩하는 데 사용된 상기 움직임 벡터의 정확도는 인코딩된 비트스트림으로 명시적으로 수신되는 것인 방법.

청구항 22

제13항에 있어서, 상기 블록을 디코딩하는 데 사용된 상기 움직임 벡터의 정확도는 상기 화상에서의 또는 상기 화상을 포함하는 시퀀스에서의 이전에 디코딩된 비디오로부터 추론(1220)되는 것인 방법.

청구항 23

비디오 신호 데이터가 인코딩되어 있는 컴퓨터 판독가능 저장 매체로서,

움직임 벡터를 사용하여 인코딩된, 화상 내의 블록

을 포함하고,

상기 블록을 인코딩하는 데 사용되는 상기 움직임 벡터의 정확도를 선택하는 데 적응적 움직임 벡터 정확도 방식이 사용되고, 상기 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 움직임 벡터 방향 적응에 기초하고,

상기 선택 기준은 움직임 벡터 성분 방향을 포함하고,

상기 선택 기준은 이웃한 블록들의 움직임 벡터들에 기초하는 움직임 벡터 성분 방향을 포함하고, 상기 블록을 인코딩하는데 사용되는 상기 움직임 벡터의 정확도는 상기 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하게 되도록 선택되며, 상기 수직 성분 및 상기 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분

으로서 선택되는 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

- [0001] <관련 출원의 상호 참조>
- [0002] 본 출원은 2009년 10월 14일자로 출원된 미국 가특허 출원 제61/251,508호를 기초로 우선권을 주장하며, 이 미국 출원은 참조 문헌으로서 그 전체 내용이 본 명세서에 포함된다.
- [0003] 본 발명은 일반적으로 비디오 인코딩 및 디코딩에 관한 것으로서, 보다 상세하게는 움직임 정보의 적응적 코딩 방법 및 장치에 관한 것이다.

배경 기술

- [0004] 움직임 보상은 많은 비디오 코딩 프레임워크에서 중요한 구성요소이다. 움직임 보상은 압축을 위해 시간 중복성을 이용하는 비디오 코딩에서 중요한 역할을 한다. 이는 움직임 정보를 사용하여 비디오 컬러 데이터를 추론하는 방식이다.
- [0005] 비디오 신호에서의 움직임은 많은 방식으로 표현될 수 있다. 가장 널리 쓰이는 표현은 변위 기반 표현인 움직임 벡터이다. 움직임 벡터가 모든 유형의 움직임을 표현하기에 충분히 정확하지는 않지만, 간소성 및 사용 용이성 특성으로 인해 움직임 벡터가 많은 비디오 관련 응용에서 널리 쓰이고 있다. 움직임 정보를 기술하는 데 더 나은 정확도를 달성하기 위해, 영상 장치의 제한된 공간 및 시간 샘플링 레이트로 인한 앨리어싱을 제거하는 데 서브펠 정확도(sub-pel accuracy)의 움직임 벡터가 종종 선호된다.
- [0006] 움직임 보상의 성능은 움직임 벡터의 정확도 및 관련 보간 프로세스(서브펠 정확도 움직임이 관여되어 있는 경우)에 크게 의존한다.
- [0007] 움직임 벡터의 정확도를 향상시키는 것은 움직임 보상의 품질을 향상시킬 수 있지만, 보다 높은 정확도의 움직임 벡터를 코딩하는 비용도 역시 증가된다. 따라서, 움직임 벡터 정확도의 향상은 코딩 비용의 증가의 대가로 오며, 그 결과 코딩된 비디오를 전송하는 데 부가의 대역폭(또는 코딩된 비디오를 저장하는 데 부가의 메모리)이 필요하게 된다. 최신 기술의 비디오 코딩 표준인 ISO/IEC(International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-4) Part 10 AVC(Advanced Video Coding) 표준/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 권고안(이후부터, "MPEG-4 AVC 표준"이라고 함)에서, 움직임 벡터는 1/4-펠 정확도를 가지며 그 중요성으로 인해 무손실 압축된다. 1/4-펠 정확도 움직임 벡터는 이전의 코딩 표준보다 코딩 효율을 향상시키는 양호한 타협안이다. 그렇지만, 대부분의 코딩 표준은 움직임 정보와 비디오 콘텐츠 사이의 관계를 고려함이 없이 균일한 움직임 벡터 정확도를 사용한다. 예를 들어, MPEG-4 AVC 표준은 비디오 화상에서의 모든 곳, 비디오 시퀀스에서의 모든 화상, 및 모든 비디오 시퀀스에 대해 1/4 펠 정확도를 사용한다.
- [0008] 1/4-펠 정확도를 갖는 움직임 벡터를 이용함으로써, 움직임 벡터 정확도의 향상으로 인해 과거의 표준보다 더 많은 코딩 이득이 달성가능하다. 1/4-펠 정확도의 움직임 벡터에서, 움직임 보상 프로세스는 적합한 보간 필터에 의존한다. MPEG-4 AVC 표준에서, 6-탭 선형 필터는 1/2-펠 보간 스테이지에서 적용되고, 선형 보간은 1/4-펠 스테이지에서 사용된다. 움직임 보상의 성능을 추가로 향상시키기 위해, 프레임별로 각각의 서브펠 위치에 대해 보간 필터를 업데이트함으로써 움직임 보상 오차를 감소시키는 데 적응적 보간 필터(adaptive interpolation filter, AIF)가 적용된다. 그렇지만, 이들 방식 모두는 단지 움직임 보상 오차를 감소시키는 것만을 고려하고 있고, 따라서 1/4-펠 정확도를 갖는 움직임 벡터의 비용을 감소시키지 않았다.
- [0009] 실제 움직임이 단지 정수 정확도(integer accuracy)일 때, 1/4-펠 정확도의 움직임 벡터를 코딩하는 것이 필요하지 않으며, 많은 비트를 낭비한다. 따라서, 이러한 균일 정확도 방식은 레이트-왜곡 비용(rate-distortion cost)의 점에서 최적이지 않다.
- [0010] 움직임 벡터에서의 중복성을 감소시켜 더 나은 코딩 성능을 얻기 위해 연구가 수행되었다. 예를 들어, 제1 종래 기술의 방식에서, MPEG-4 AVC 표준에서의 무손실 방식 대신에 움직임 벡터의 손실 압축을 가능하게 해주는 움직임 벡터 양자화 방식이 기술되어 있다. 게다가, 이 방식은, MPEG-4 AVC 표준의 다른 기존의 모드와 함께, 부가의 코딩 모드(QMV 모드라고 함)를 추가한다. QMV 모드에서, 엔트로피 인코딩 이전에 파티션의 움직임 벡터

가 양자화될 것이다. 공간 적응(spatial adaptation)을 실현하기 위해 다양한 매크로블록에서 양자화 스텝 Qv가 상이할 수 있다. QMV 모드는 레이트 왜곡(rate distortion)에 기초하여 움직임 벡터를 상이한 정확도로 표현하는 것의 적응을 달성할 수 있다. Qv 값 및 QMV 모드 정보를 전송하는 데 소비되는 부가의 비용이 움직임 벡터에서의 레이트 절감으로 이룩한 이득을 상쇄해 버릴 수 있을 것이다.

[0011]

<개요>

[0012]

움직임 정보의 적응적 코딩 방법 및 장치에 관한 본 발명에 의해 종래 기술의 이들 및 기타 결점 및 단점이 해결된다.

[0013]

본 발명의 한 측면에 따르면, 장치가 제공된다. 이 장치는 움직임 벡터를 사용하여 적어도 화상에서의 블록을 인코딩하는 인코더를 포함한다. 적응적 움직임 벡터 정확도 방식은 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준(non-rate-distortion-based criteria)을 포함한다.

[0014]

본 발명의 다른 측면에 따르면, 비디오 인코더에서의 방법이 제공된다. 이 방법은 움직임 벡터를 사용하여 적어도 화상에서의 블록을 인코딩하는 단계를 포함한다. 적응적 움직임 벡터 정확도 방식은 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준을 포함한다.

[0015]

본 발명의 또 다른 측면에 따르면, 장치가 제공된다. 이 장치는 움직임 벡터를 사용하여 적어도 화상에서의 블록을 디코딩하는 디코더를 포함한다. 적응적 움직임 벡터 정확도 방식은 블록을 디코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준을 포함한다.

[0016]

본 발명의 또 다른 측면에 따르면, 비디오 디코더에서의 방법이 제공된다. 이 방법은 움직임 벡터를 사용하여 적어도 화상에서의 블록을 디코딩하는 단계를 포함한다. 적응적 움직임 벡터 정확도 방식은 블록을 디코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준을 포함한다.

[0017]

본 발명의 이들 및 기타 측면, 특징 및 이점이 첨부 도면과 관련하여 기술된 예시적인 실시예에 대한 이하의 상세한 설명으로부터 명백하게 될 것이다.

도면의 간단한 설명

[0018]

본 발명은 이하의 예시적인 도면들에 따라 더 잘 이해될 수 있다.

도 1은 본 발명의 일 실시예에 따른, 본 발명이 적용될 수 있는 예시적인 비디오 인코더를 나타낸 블록도.

도 2는 본 발명의 일 실시예에 따른, 본 발명이 적용될 수 있는 예시적인 비디오 디코더를 나타낸 블록도.

도 3은 본 발명의 일 실시예에 따른, 파티션 크기에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 4는 본 발명의 일 실시예에 따른, 파티션 크기에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

도 5는 본 발명의 일 실시예에 따른, 움직임 벡터 방향에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 6은 본 발명의 일 실시예에 따른, 움직임 벡터 방향에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

도 7은 본 발명의 일 실시예에 따른, 양자화 파라미터에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 8은 본 발명의 일 실시예에 따른, 양자화 파라미터에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

도 9는 본 발명의 일 실시예에 따른, 명시적 시그널링(explicit signaling)을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 10은 본 발명의 일 실시예에 따른, 명시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

도 11은 본 발명의 일 실시예에 따른, 암시적 시그널링(implicit signaling)을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 12는 본 발명의 일 실시예에 따른, 암시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

도 13은 본 발명의 일 실시예에 따른, 움직임 벡터 진폭에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법을 나타낸 흐름도.

도 14는 본 발명의 일 실시예에 따른, 암시적 시그널링을 갖는 움직임 벡터 진폭에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법을 나타낸 흐름도.

발명을 실시하기 위한 구체적인 내용

- [0019] 본 발명은 움직임 정보의 적응적 코딩 방법 및 장치에 관한 것이다.
- [0020] 이 설명은 본 발명에 대해 기술한다. 따라서, 당업자라면, 본 명세서에 명시적으로 기술되거나 도시되어 있지 않지만, 본 발명을 구현하고 또 본 발명의 사상 및 범위 내에 포함되는 다양한 구성들을 안출할 수 있다는 것을 잘 알 것이다.
- [0021] 본 명세서에 상술되는 모든 일례 및 조건적 화법은 기술을 발전시키기 위해 발명자들이 기여한 개념들 및 본 발명을 독자가 이해하는 데 도움을 주도록 교육적 목적을 위한 것이며, 이러한 구체적으로 상술된 예들 및 조건들로 제한되는 것으로 해석되어서는 안 된다.
- [0022] 게다가, 본 발명의 원리, 측면 및 실시예는 물론 본 발명의 구체적인 예들을 기술하는 본 명세서의 모든 설명이 본 발명의 구조적 등가물 및 기능적 등가물 둘 다를 포괄하는 것으로 보아야 한다. 그에 부가하여, 이러한 등가물이 현재 공지된 등가물은 물론 장래에 개발되는 등가물, 즉 구조에 상관없이 동일한 기능을 수행하는 개발된 임의의 요소도 포함하는 것으로 보아야 한다.
- [0023] 따라서, 예를 들어, 당업자라면 본 명세서에 제공된 블록도가 본 발명을 구현하는 예시적인 회로의 개념도를 나타낸다는 것을 잘 알 것이다. 이와 마찬가지로, 어떤 플로우차트, 흐름도, 상태 천이도, 의사코드 등이, 사실상 컴퓨터 판독가능 매체에 표현되고 따라서 컴퓨터 또는 프로세서(이러한 컴퓨터 또는 프로세서가 명시적으로 도시되어 있는 그렇지 않은 간에)에 의해 실행될 수 있는 다양한 프로세스를 나타낸다는 것을 잘 알 것이다.
- [0024] 도면에 도시된 다양한 요소들의 기능이 전용 하드웨어는 물론 적절한 소프트웨어와 관련하여 소프트웨어를 실행할 수 있는 하드웨어를 사용하여 제공될 수 있다. 프로세서에 의해 제공될 때, 이들 기능은 하나의 전용 프로세서에 의해, 하나의 공유 프로세서에 의해, 또는 복수의 개별 프로세서(이들 중 일부는 공유될 수 있음)에 의해 제공될 수 있다. 게다가, "프로세서" 또는 "제어기"라는 용어의 명시적 사용이 소프트웨어를 실행할 수 있는 하드웨어만을 말하는 것으로 해석되어서는 안되며, 디지털 신호 처리기(DSP) 하드웨어, 소프트웨어를 저장하는 판독 전용 메모리(ROM), 랜덤 액세스 메모리(RAM), 및 비휘발성 저장 장치(이들로 한정되지 않음)를 암시적으로 포함할 수 있다.
- [0025] 기타 종래의 하드웨어(conventional hardware) 및/또는 커스텀 하드웨어(custom hardware)도 역시 포함될 수 있다. 이와 마찬가지로, 도면들에 도시된 임의의 스위치는 단지 개념적인 것이다. 이들의 기능은 프로그램 로직의 동작을 통해, 전용 로직을 통해, 프로그램 제어와 전용 로직의 상호작용을 통해, 또는 심지어 수동으로 수행될 수 있으며, 문맥으로부터 보다 구체적으로 이해되는 바와 같이 특정의 기법이 구현자에 의해 선택될 수 있다.
- [0026] 본 명세서의 청구항에서, 지정된 기능을 수행하는 수단으로서 표현된 임의의 요소는, 예를 들어, a) 그 기능을 수행하는 회로 요소들의 조합 또는 b) 기능을 수행하기 위해 소프트웨어를 실행하는 적절한 회로와 결합된 임의의 형태의 소프트웨어(따라서, 펌웨어, 마이크로코드, 기타를 포함함)를 비롯하여, 그 기능을 수행하는 임의의 방식을 포함하는 것으로 보아야 한다. 이러한 청구항에 의해 정의된 본 발명은 다양한 인용된 수단들에 의해 제공된 기능들이 청구항이 요구하는 방식으로 결합되어 합해진다는 사실에 존재한다. 따라서, 그 기능들을 제공할 수 있는 어떤 수단이라도 본 명세서에 제공된 기능들과 등가인 것으로 간주된다.

- [0027] 본 명세서에서 본 발명의 "일 실시예" 또는 "실시예"는 물론 그의 다른 변형이라고 하는 것은 이 실시예와 관련 하여 기술된 특성의 특징, 구조, 특성, 기타가 본 발명의 적어도 하나의 실시예에 포함되어 있다는 것을 의미한다. 따라서, 본 명세서에 걸쳐 여러 곳에서 나오는 "일 실시예에서" 또는 "실시예에서"라는 문구가 반드시 모두 동일한 실시예를 말하는 것은 아니다.
- [0028] "/", "및/또는" 그리고 "~중 적어도 하나" 중 임의의 표현의 사용(예를 들어, "A/B", "A 및/또는 B"와 "A와 B 중 적어도 하나"의 경우)이 첫 번째로 열거된 옵션(A)만의 선택, 또는 두 번째로 열거된 옵션(B)만의 선택, 또는 양쪽 옵션(A 및 B) 둘 다의 선택을 포괄하는 것으로 보아야 한다는 것을 잘 알 것이다. 추가의 일례로서, "A, B 및/또는 C"와 "A, B 및 C 중 적어도 하나"의 경우에, 이러한 구문은 첫 번째로 열거된 옵션(A)만의 선택, 또는 두 번째로 열거된 옵션(B)만의 선택, 또는 세 번째로 열거된 옵션(C)만의 선택, 또는 첫 번째 및 두 번째로 열거된 옵션(A 및 B)만의 선택, 또는 첫 번째 및 세 번째로 열거된 옵션(A 및 C)만의 선택, 또는 두 번째 및 세 번째로 열거된 옵션(B 및 C)만의 선택, 또는 3개의 옵션(A 및 B 및 C) 모두의 선택을 포괄하는 것으로 보아야 한다. 당업자라면 잘 알 것인 바와 같이, 열거된 동수의 항목들에 대해 이것이 확장될 수 있다.
- [0029] 게다가, 본 발명의 하나 이상의 실시예가 본 명세서에서 MPEG-4 AVC 표준과 관련하여 기술되어 있지만, 본 발명이 이 표준으로만 제한되지 않고 따라서 본 발명의 사상을 유지하면서 다른 비디오 코딩 표준, 권고안 및 이들의 확장(MPEG-4 AVC 표준의 확장은 물론, 독점적 및 장래의 표준 또는 방식을 포함함)과 관련하여 이용될 수 있다는 것을 잘 알 것이다.
- [0030] 또한, 본 명세서에서 사용되는 바와 같이, 단어 "화상" 및 "영상"은 서로 바꾸어 사용될 수 있고, 정지 영상 또는 비디오 시퀀스로부터의 화상을 말한다. 공지된 바와 같이, 화상은 프레임 또는 필드일 수 있다.
- [0031] 그에 부가하여, 본 명세서에서 사용되는 바와 같이, "신호한다"라는 단어는 대응하는 디코더에 무언가를 알려주는 것을 말한다. 예를 들어, 인코더는 어느 특성의 움직임 벡터 정확도가 인코더측에서 사용되었는지를 디코더에게 알려주기 위해 주어진 움직임 벡터 정확도를 신호할 수 있다. 이러한 방식으로, 동일한 움직임 벡터 정확도가 인코더측 및 디코더측 둘 다에서 사용될 수 있다. 따라서, 예를 들어, 인코더는 특성의 움직임 벡터 정확도를 디코더로 전송함으로써 디코더가 동일한 특성의 움직임 벡터 정확도를 사용할 수 있도록 할 수 있거나, 디코더가 이미 특성의 움직임 벡터 정확도는 물론 다른 것들도 가지고 있는 경우, 단지 디코더가 특성의 움직임 벡터 정확도를 알고 선택할 수 있게 해주기 위해 시그널링이 (전송하는 일 없이) 사용될 수 있다. 임의의 실제 움직임 벡터 정확도의 전송을 피함으로써, 비트 절감이 실현될 수 있다. 시그널링이 각종의 방식으로 달성될 수 있다는 것을 잘 알 것이다. 예를 들어, 정보를 대응하는 디코더에 신호하기 위해 하나 이상의 구문 요소, 플래그 등이 사용될 수 있다.
- [0032] 게다가, 본 명세서에서 사용되는 바와 같이, "국소 화상 영역(local picture region)"이라는 어구는 비디오 시퀀스의 서브셋 신호(subset signal)를 말한다. 국소 화상 영역은 다수의 연속적인 프레임, 단일 프레임, 다수의 시간적으로 및/또는 공간적으로 이웃한 블록, 및/또는 다수의 시간적으로 및/또는 공간적으로 이웃한 픽셀일 수 있다.
- [0033] 또한, 본 명세서에서 사용되는 바와 같이, "전역 움직임 정보(global motion information)"라는 어구는 "화상 영역"에서의 우세 움직임(dominant motion)을 말한다. 본 명세서에서 사용되는 바와 같이, "화상 영역"이라는 어구는 동일한 장면에 속하는 다수의 프레임, 단일 프레임, 및/또는 단일 프레임 내의 일부분을 말한다. 전역 움직임 정보의 몇몇 예들이 다음과 같이 제공된다. 일례에서, 특성의 화상 영역 내의 모든 블록에 대한 움직임을 추정하고, 전역 움직임 정보는 이들 블록에서 가장 통상적인 움직임이다. 다른 예에서, 특성의 화상 영역 내의 모든 블록에 대한 움직임을 추정하고, 전역 움직임 정보는 모든 이들 블록에 걸쳐 평균된 움직임이다. 또 다른 예에서, 특성의 화상 영역 내의 모든 블록에 대한 움직임을 추정하고, 전역 움직임 정보는 모든 이들 블록 중에서의 메디안 움직임(median motion)이다.
- [0034] 도 1을 참조하면, 본 발명이 적용될 수 있는 예시적인 비디오 인코더가 전체적으로 참조 번호(100)로 나타내어져 있다. 비디오 인코더(100)는 결합기(combiner)(185)의 비반전 입력과 신호 통신을 하는 출력을 갖는 프레임 순서화 버퍼(frame ordering buffer)(110)를 포함하고 있다. 결합기(185)의 출력은 변환기 및 양자화기(125)의 제1 입력에 연결되어 신호 통신을 한다. 변환기 및 양자화기(125)의 출력은 엔트로피 코더(entropy coder)(145)의 제1 입력 그리고 역변환기 및 역양자화기(150)의 제1 입력에 연결되어 신호 통신을 한다. 엔트로피 코더(145)의 출력은 결합기(190)의 제1 비반전 입력에 연결되어 신호 통신을 한다. 결합기(190)의 출력은 출력 버퍼(135)의 제1 입력에 연결되어 신호 통신을 한다.

- [0035] 인코더 제어기(105)의 제1 출력은 프레임 순서화 버퍼(110)의 제2 입력, 역변환기 및 역양자화기(150)의 제2 입력, 화상-타입 결정 모듈(picture-type decision module)(115)의 입력, 매크로블록-타입(MB-타입) 결정 모듈(MB-type decision module)(120)의 제1 입력, 인트라 예측 모듈(intra prediction module)(160)의 제2 입력, 디블로킹 필터(165)의 제2 입력, 움직임 보상기(motion compensator)(170)의 제1 입력, 움직임 추정기(motion estimator)(175)의 제1 입력, 및 기준 화상 버퍼(reference picture buffer)(180)의 제2 입력에 연결되어 신호 통신을 한다.
- [0036] 인코더 제어기(105)의 제2 출력은 SEI(Supplemental Enhancement Information) 삽입기(130)의 제1 입력, 변환기 및 양자화기(125)의 제2 입력, 엔트로피 코더(145)의 제2 입력, 출력 버퍼(135)의 제2 입력, 및 SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set) 삽입기(140)의 입력에 연결되어 신호 통신을 한다.
- [0037] SEI 삽입기(130)의 출력은 결합기(190)의 제2 비반전 입력에 연결되어 신호 통신을 한다.
- [0038] 화상-타입 결정 모듈(115)의 제1 출력은 프레임 순서화 버퍼(110)의 제3 입력에 연결되어 신호 통신을 한다. 화상-타입 결정 모듈(115)의 제2 출력은 매크로블록-타입 결정 모듈(120)의 제2 입력에 연결되어 신호 통신을 한다.
- [0039] SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set) 삽입기(140)의 출력은 결합기(190)의 제3 비반전 입력에 연결되어 신호 통신을 한다.
- [0040] 역양자화기 및 역변환기(150)의 출력은 결합기(119)의 제1 비반전 입력에 연결되어 신호 통신을 한다. 결합기(119)의 출력은 인트라 예측 모듈(160)의 제1 입력 및 디블로킹 필터(165)의 제1 입력에 연결되어 신호 통신을 한다. 디블로킹 필터(165)의 출력은 기준 화상 버퍼(180)의 제1 입력에 연결되어 신호 통신을 한다. 기준 화상 버퍼(180)의 출력은 움직임 추정기(175)의 제2 입력 및 움직임 보상기(170)의 제3 입력에 연결되어 신호 통신을 한다. 움직임 추정기(175)의 제1 출력은 움직임 보상기(170)의 제2 입력에 연결되어 신호 통신을 한다. 움직임 추정기(175)의 제2 출력은 엔트로피 코더(145)의 제3 입력에 연결되어 신호 통신을 한다.
- [0041] 움직임 보상기(170)의 출력은 스위치(197)의 제1 입력에 연결되어 신호 통신을 한다. 인트라 예측 모듈(160)의 출력은 스위치(197)의 제2 입력에 연결되어 신호 통신을 한다. 매크로블록-타입 결정 모듈(120)의 출력은 스위치(197)의 제3 입력에 연결되어 신호 통신을 한다. 스위치(197)의 제3 입력은 스위치의 "데이터" 입력(제어 입력, 즉 제3 입력과 비교한 것임)이 움직임 보상기(170) 또는 인트라 예측 모듈(160)에 의해 제공되어야 하는지를 결정한다. 스위치(197)의 출력은 결합기(119)의 제2 비반전 입력 및 결합기(185)의 반전 입력에 연결되어 신호 통신을 한다.
- [0042] 프레임 순서화 버퍼(110)의 제1 입력 및 인코더 제어기(105)의 입력은, 입력 화상을 수신하기 위해, 인코더(100)의 입력으로서 이용가능하다. 게다가, SEI(Supplemental Enhancement Information) 삽입기(130)의 제2 입력은, 메타데이터를 수신하기 위해, 인코더(100)의 입력으로서 이용가능하다. 출력 버퍼(135)의 출력은, 비트스트림을 출력하기 위해, 인코더(100)의 출력으로서 이용가능하다.
- [0043] 도 2를 참조하면, 본 발명이 적용될 수 있는 예시적인 비디오 디코더가 전체적으로 참조 번호(200)로 나타내어져 있다. 비디오 디코더(200)는 엔트로피 디코더(245)의 제1 입력에 연결되어 신호 통신을 하는 출력을 갖는 입력 버퍼(210)를 포함하고 있다. 엔트로피 디코더(245)의 제1 출력은 역변환기 및 역양자화기(250)의 제1 입력에 연결되어 신호 통신을 한다. 역변환기 및 역양자화기(250)의 출력은 결합기(225)의 제2 비반전 입력에 연결되어 신호 통신을 한다. 결합기(225)의 출력은 디블로킹 필터(265)의 제2 입력 및 인트라 예측 모듈(260)의 제1 입력에 연결되어 신호 통신을 한다. 디블로킹 필터(265)의 제2 출력은 기준 화상 버퍼(280)의 제1 입력에 연결되어 신호 통신을 한다. 기준 화상 버퍼(280)의 출력은 움직임 보상기(270)의 제2 입력에 연결되어 신호 통신을 한다.
- [0044] 엔트로피 디코더(245)의 제2 출력은 움직임 보상기(270)의 제3 입력, 디블로킹 필터(265)의 제1 입력, 및 인트라 예측기(260)의 제3 입력에 연결되어 신호 통신을 한다. 엔트로피 디코더(245)의 제3 출력은 디코더 제어기(205)의 입력에 연결되어 신호 통신을 한다. 디코더 제어기(205)의 제1 출력은 엔트로피 디코더(245)의 제2 입력에 연결되어 신호 통신을 한다. 디코더 제어기(205)의 제2 출력은 역변환기 및 역양자화기(250)의 제2 입력에 연결되어 신호 통신을 한다. 디코더 제어기(205)의 제3 출력은 디블로킹 필터(265)의 제3 입력에 연결되어 신호 통신을 한다. 디코더 제어기(205)의 제4 출력은 인트라 예측 모듈(260)의 제2 입력, 움직임 보상기(270)의 제1 입력, 및 기준 화상 버퍼(280)의 제2 입력에 연결되어 신호 통신을 한다.

[0045] 움직임 보상기(270)의 출력은 스위치(297)의 제1 입력에 연결되어 신호 통신을 한다. 인트라 예측 모듈(260)의 출력은 스위치(297)의 제2 입력에 연결되어 신호 통신을 한다. 스위치(297)의 출력은 결합기(225)의 제1 비반전 입력에 연결되어 신호 통신을 한다.

[0046] 입력 버퍼(210)의 입력은, 입력 비트스트림을 수신하기 위해, 디코더(200)의 입력으로서 이용가능하다. 디블로킹 필터(265)의 제1 출력은, 출력 화상을 출력하기 위해, 디코더(200)의 출력으로서 이용가능하다.

[0047] 앞서 살펴본 바와 같이, 본 발명은 움직임 정보의 적응적 코딩 방법 및 장치에 관한 것이다. 따라서, 본 발명에 따르면, 움직임 정보와 비디오 콘텐츠 사이의 상관을 더 잘 이용함으로써 비디오 코딩 성능을 향상시키기 위해 적응적 움직임 정보 표현 및 압축 방식이 이용된다. 이 방식은, 적응을 위해 부가적인 비트 오버헤드를 발생시키는 일 없이(또는 최소한 부가적인 비트 오버헤드를 제한하여), 움직임 필드(motion field), 비디오 콘텐츠, 코딩 모드, 및 코딩 효율을 고려함으로써 적응적으로 움직임 벡터를 상이한 정확도 레벨로 표현한다.

[0048] 파티션 크기 적응

[0049] 전형적인 블록-기반 비디오 코딩 방식에서, 화상은 다수의 비중첩 블록으로 분할된다. 최적의 블록 형상 및 크기는 비디오 콘텐츠 및 코딩 방식에 의존한다. MPEG-4 AVC 표준은 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 및 4x4 블록을 지원한다. 알 수 있는 바와 같이, 큰 블록은 작은 블록보다 더 많은 픽셀을 가진다. 움직임 보상 오차는 각각의 픽셀로부터의 오차에 의해 영향을 받는다. 블록이 보다 많은 픽셀을 포함하는 경우, 각각의 픽셀로부터의 오차가 균일하다고 가정할 때, 그 블록은 더 큰 보상 오차를 가질 가능성이 비교적 더 높다. 따라서, 작은 블록에 비해 큰 블록에 대해 더 높은 정확도의 움직임 벡터를 사용하는 것을 선호한다. 따라서, 일 실시예에서, 움직임 벡터 정확도를 파티션 크기에 맞춰 적응시킨다.

[0050] 일반적으로, 큰 블록의 움직임 벡터에 대해 더 높은 정확도를 사용할 수 있는데, 그 이유는 큰 블록이 비디오에서 더 많은 면적을 차지하고, 올바르게 보상되지 않는 경우, 많은 양의 왜곡의 원인이 될 가능성이 높기 때문이다. 표 1은 본 발명의 실시예에 따른, 상이한 블록 크기를 상이한 정확도 레벨로 분류하는 것을 보여주고 있다. 물론, 본 발명이 상기한 분류로 제한되지 않고, 따라서 본 발명의 사상을 유지하면서, 본 발명의 개시 내용에 따라 다른 분류도 사용될 수 있다는 것을 잘 알 것이다.

표 1

[0051]

레벨	파티션 크기	움직임 벡터 정확도
0	16x16, 16x8, 8x16	1/8 펄
1	8x8	1/4 펄
2	8x4, 4x8, 4x4	1/2 펄

[0052] 각각의 블록의 움직임 벡터가 그 레벨의 대응하는 정확도로 표현될 것이다. 이미 전송된 파티션 크기에 기초하여, 어떤 부가의 비트 레이트도 움직임 벡터 정확도 적응에 소비되지 않는다.

[0053] 도 3을 참조하면, 파티션 크기에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법이 전체적으로 참조 번호(300)로 나타내어져 있다. 방법(300)은 제어를 기능 블록(310)으로 넘기는 시작 블록(305)을 포함한다. 기능 블록(310)은 motion_accuracy_adaptive_flag = 1로 설정하고, mv_accuracy_adaptation_mode = 0으로 설정하며, motion_accuracy_adaptive_flag 및 mv_accuracy_adaptation_mode를 비트스트림에 기입하고, 제어를 루프 한계 블록(312)으로 넘긴다. 루프 한계 블록(312)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(315)으로 넘긴다. 기능 블록(315)은 움직임 추정을 수행하고, 제어를 기능 블록(320)으로 넘긴다. 기능 블록(320)은 다음과 같이 파티션 크기에 기초한 움직임 추정[기능 블록(315)에 의해 수행됨]으로부터 얻어진 움직임 벡터를 양자화하고, 그 후에 제어를 기능 블록(325)으로 넘긴다. 16x16, 16x8, 8x16 파티션 크기는 1/8 펄 정확도를 사용하고, 8x8 파티션 크기는 1/4 펄 정확도를 사용하며, 8x4, 4x8, 4x4 파티션 크기는 1/2 펄 정확도를 사용한다. 기능 블록(325)은 움직임 보상을 수행하고, 제어를 기능 블록(330)으로 넘긴다. 기능 블록(330)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(332)으로 넘긴다. 루프 한계 블록은 루프를 종료하고, 제어를 종료 블록(399)으로 넘긴다.

[0054] 도 4를 참조하면, 파티션 크기에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(400)로 나타내어져 있다. 방법(400)은 제어를 기능 블록(410)으로 넘기는

시작 블록(405)을 포함한다. 기능 블록(410)은 motion_accuracy_adaptive_flag 및 mv_accuracy_adaptation_mode를 파싱하고, 제어를 루프 한계 블록(412)으로 넘긴다. 루프 한계 블록(412)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(413)으로 넘긴다. 기능 블록(413)은 움직임 벡터(MV) 구문을 파싱하고, 제어를 결정 블록(415)으로 넘긴다. 결정 블록(415)은 motion_accuracy_adaptive_flag == 1이고 mv_accuracy_adaptation_mode == 0인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(420)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(417)으로 넘어간다. 기능 블록(420)은 다음과 같이 파티션 크기로 결정된 정확도에 기초하여 움직임 벡터를 디코딩하고, 그 후에 제어를 기능 블록(425)으로 넘긴다. 16x16, 16x8, 8x16 파티션 크기는 1/8 펄 정확도를 사용하고, 8x8 파티션 크기는 1/4 펄 정확도를 사용하며, 8x4, 4x8, 4x4 파티션 크기는 1/2 펄 정확도를 사용한다. 기능 블록(425)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(427)으로 넘긴다. 루프 한계 블록(427)은 루프를 종료하고, 제어를 종료 블록(499)으로 넘긴다. 기능 블록(417)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(425)으로 넘긴다.

[0055] 움직임 벡터 방향 적응

[0056] 대부분의 움직임 벡터 표현에서, 움직임 벡터는 수평 및 수직 방향 둘 다에서의 움직임을 기술하는 2차원 벡터이다. 보통, 움직임 벡터는 양 방향에서 동일한 정확도를 가진다. 그렇지만, 상이한 방향에서 동일한 정확도를 가질 필요는 없으며, 움직임에 관한 어떤 이전의 정보를 가지고 있을 때 특히 그렇다. 예를 들어, 비디오가 (카메라 패닝과 같이) 우세한 수평 움직임을 가지는 경우, 움직임 정보를 더 잘 표현하기 위해 수평 방향에서 더 높은 정확도를 제공할 수 있다. 또한, 높은 정확도의 움직임 방향을 신호하기 위해 정수 움직임 진폭, 파티션 형상, 움직임 벡터 예측기, 또는 전역 움직임 정보를 이용할 수 있다. 따라서, 일 실시예에서, 움직임 벡터의 하나 이상의 특징의 방향에 맞춰 움직임 벡터 정확도를 적응시킨다.

[0057] 일 실시예에서, 우세 움직임 방향을 도출하기 위해 움직임 벡터 예측기를 이용한다. 우세 움직임 방향에 (비우세 움직임 방향보다) 더 높은 움직임 벡터 정확도를 할당할 것이다. MPEG-4 AVC 표준에 기술된 바와 같이, 이웃 블록의 움직임 벡터를 검사함으로써 움직임 벡터 예측기 $mvp = [mvp_x, mvp_y]$ 가 획득될 수 있다. 다음과 같은 파라미터를 정의한다:

수학식 1

$$\theta(mvp) = \begin{cases} \frac{|mvp_y|}{|mvp_x|}, & |mvp_x| \neq 0 \\ Inf., & |mvp_x| = 0 \end{cases}$$

[0058]

[0059] $\theta(mvp)$ 를 검사함으로써, 어느 방향이 더 높은 정확도를 사용하는지를 다음과 같이 결정할 수 있고:

수학식 2

$$\begin{cases} res(mv_y) = 1/8, res(mv_x) = 1/4, \theta(mvp) \geq th1 \\ res(mv_y) = res(mv_x) = 1/4, th1 > \theta(mvp) > th2, \\ res(mv_y) = 1/4, res(mv_x) = 1/8, \theta(mvp) \leq th2 \end{cases}$$

[0060]

[0061] 여기서 $res(mv_x)$ 및 $res(mv_y)$ 는 mv_x 및 mv_y 의 각자의 분해능(즉, 정확도)이고, $th1$ 및 $th2$ 는 상이한 방향에서의 움직임 벡터의 정확도를 결정하는 2개의 임계값이다.

[0062] 이 방식을 사용함으로써, 부가적인 오버헤드를 발생시키지 않고 상이한 방향에서 움직임 벡터의 정확도를 적응시킬 수 있다.

[0063] 도 5를 참조하면, 움직임 벡터 방향에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는

예시적인 방법이 전체적으로 참조 번호(500)로 나타내어져 있다. 방법(500)은 제어를 기능 블록(510)으로 넘기는 시작 블록(505)을 포함한다. 기능 블록(510)은 `motion_accuracy_adaptive_flag = 1`로 설정하고, `mv_accuracy_adaptation_mode = 1`로 설정하며, `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 비트스트림에 기입하고, 제어를 루프 한계 블록(512)으로 넘긴다. 루프 한계 블록(512)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(515)으로 넘긴다. 기능 블록(515)은 움직임 추정을 수행하고, 제어를 기능 블록(520)으로 넘긴다. 기능 블록(520)은 우세 성분을 도출하고, 움직임 벡터 예측기, 이웃 움직임 벡터, 파티션 형상, 정수 움직임 진폭, 전역 움직임, 및/또는 레이트 왜곡 비용 중 하나 이상에 기초하여 `res_x` 및 `res_y`를 설정하며, 그 후에 제어를 기능 블록(525)으로 넘긴다. 기능 블록(525)은 `res_x` 및 `res_y`에 기초하여 움직임 벡터 성분을 양자화하고, 제어를 기능 블록(530)으로 넘긴다. 기능 블록(530)은 움직임 보상을 수행하고, 제어를 기능 블록(535)으로 넘긴다. 기능 블록(535)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(537)으로 넘긴다. 루프 한계 블록(537)은 루프를 종료하고, 제어를 종료 블록(599)으로 넘긴다.

[0064] 도 6을 참조하면, 움직임 벡터 방향에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(600)로 나타내어져 있다. 방법(600)은 제어를 기능 블록(610)으로 넘기는 시작 블록(605)을 포함한다. 기능 블록(610)은 `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 파싱하고, 제어를 루프 한계 블록(612)으로 넘긴다. 루프 한계 블록(612)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(613)으로 넘긴다. 기능 블록(613)은 움직임 벡터(MV) 구문을 파싱하고, 제어를 결정 블록(615)으로 넘긴다. 결정 블록(615)은 `motion_accuracy_adaptive_flag == 1`이고 `mv_accuracy_adaptation_mode == 1`인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(620)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(617)으로 넘어간다. 기능 블록(620)은 우세 성분을 도출하고, 움직임 벡터 예측기, 이웃 움직임 벡터, 파티션 형상, 정수 움직임 진폭, 전역 움직임, 및/또는 레이트 왜곡 비용 중 하나 이상에 기초하여 `res_x` 및 `res_y`를 설정하며, 그 후에 제어를 기능 블록(625)으로 넘긴다. 기능 블록(625)은 `res_x` 및 `res_y`에 기초하여 움직임 벡터를 디코딩하고, 제어를 기능 블록(630)으로 넘긴다. 기능 블록(630)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(632)으로 넘긴다. 루프 한계 블록(632)은 루프를 종료하고, 제어를 종료 블록(699)으로 넘긴다. 기능 블록(617)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(630)으로 넘긴다.

[0065] QP 적응적 움직임 벡터 정확도

[0066] 비디오 인코더는 인코딩된 비디오의 품질을 제어하기 위해 양자화 파라미터(QP)를 사용한다. 양자화 파라미터가 클 때, 기준 프레임(이전에 인코딩된 프레임의 재구성임)의 품질이 낮다. 상세하게는, 기준 프레임은 평탄한 경향이 있는데, 그 이유는 인코딩 프로세스에서 대부분의 세부이 제거되기 때문이다. 따라서, 작은 차이를 갖는 움직임 벡터가 아주 유사한 예측을 제공할 수 있고, 높은 정확도의 움직임 벡터가 필요하지 않다. 따라서, 일 실시예에서, 하나 이상의 양자화 파라미터에 맞춰 움직임 벡터 정확도를 적응시킨다.

[0067] 일 실시예에서, 움직임 벡터 정확도가 인코딩 양자화 파라미터(QP) 또는 양자화 스텝 크기에 맞춰 적응된다. `mv`가 움직임 추정에 의해 발견된 움직임 벡터이고 `mvp`가 예측된 움직임 벡터인 것으로 가정하자. 차이는 `mvd`로서 표시되고, 여기서 `mvd=mv-mvp`이다. `mvq`를 디코더로 전송될 양자화된 `mvd`라고 하고, 여기서 `mvq=Q(mvd,q_mv)`이고, `Q`는 `mvd` 양자화 프로세스이며, `q_mv`는 양자화 스텝 크기이다(예컨대, `q_mv = 0.5`는 1/2-펠 정확도를 의미하고, `q_mv = 0.25`는 1/4-펠 정확도를 의미하며, 이하 마찬가지이다). 움직임 벡터 정확도를 인코딩 QP의 함수 - 즉, `q_mv = f(QP)` - 라고 하자. 예를 들어, QP가 임계값보다 작을 때, `q_mv`의 작은 값이 선택된다. 그렇지 않고, QP가 임계값보다 클 때, `q_mv`의 큰 값이 선택된다.

[0068] 도 7을 참조하면, 양자화 파라미터에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법이 전체적으로 참조 번호(700)로 나타내어져 있다. 방법(700)은 제어를 기능 블록(710)으로 넘기는 시작 블록(705)을 포함한다. 기능 블록(710)은 `motion_accuracy_adaptive_flag = 1`로 설정하고, `mv_accuracy_adaptation_mode = 2`로 설정하며, `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 비트스트림에 기입하고, 제어를 루프 한계 블록(712)으로 넘긴다. 루프 한계 블록(712)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(715)으로 넘긴다. 기능 블록(715)은 움직임 추정을 수행하고, 제어를 기능 블록(720)으로 넘긴다. 기능 블록(720)은 인코딩 양자화 파라미터(QP)에 기초하여 움직임 벡터 정확도를 선택하고, 선택된 움직임 벡터 정확도를 사용하여 움직임 추정[기능 블록(715)에 의해 수행됨]으로부터 얻어진 움직임 벡터를 양자화하며, 제어를 기능

블록(725)으로 넘긴다. 기능 블록(725)은 움직임 보상을 수행하고, 제어를 기능 블록(730)으로 넘긴다. 기능 블록(730)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(732)으로 넘긴다. 루프 한계 블록(732)은 루프를 종료하고, 제어를 종료 블록(799)으로 넘긴다.

[0069] 도 8을 참조하면, 양자화 파라미터에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(800)로 나타내어져 있다. 방법(800)은 제어를 기능 블록(810)으로 넘기는 시작 블록(805)을 포함한다. 기능 블록(810)은 `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 파싱하고, 제어를 루프 한계 블록(812)으로 넘긴다. 루프 한계 블록(812)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(813)으로 넘긴다. 기능 블록(813)은 움직임 벡터(MV) 구문을 파싱하고, 제어를 결정 블록(815)으로 넘긴다. 결정 블록(815)은 `motion_accuracy_adaptive_flag == 1`이고 `mv_accuracy_adaptation_mode == 2`인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(820)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(817)으로 넘어간다. 기능 블록(820)은 양자화 파라미터(QP)로부터 움직임 벡터 정확도 함수를 획득하고, 수신된 움직임 벡터 인덱스로부터 움직임 벡터를 재구성하며, 제어를 기능 블록(825)으로 넘긴다. 기능 블록(825)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(827)으로 넘긴다. 루프 한계 블록은 루프를 종료하고, 제어를 종료 블록(899)으로 넘긴다. 기능 블록(817)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(825)으로 넘긴다.

[0070] 콘텐츠 적응적 움직임 벡터 정확도

[0071] 비디오 신호에서의 평탄한 영역에 대해, 작은 차이를 갖는 움직임 벡터는 아주 유사한 예측을 제공할 수 있고, 따라서 높은 정확도의 움직임 벡터로부터의 이점이 제한된다. 한편, 객체 에지 및 텍스처링된 영역에 대해, 예측과 현재의 신호 사이의 약간의 불일치는 예측 오차를 크게 증가시킬 수 있고, 따라서 정확한 움직임 벡터가 아주 바람직하다. 이 관계를 고려하여, 일 실시예에서, 화상(또는 시퀀스) 콘텐츠에 따라 움직임 벡터 정확도를 적응시킨다.

[0072] 일 실시예에서, 움직임 벡터 정확도가 화상 콘텐츠에 따라 적응적이다. `S`를 비디오 시퀀스의 서브셋 신호라고 하자. `S`는 다수의 연속적인 프레임, 단일 프레임 또는 다수의 이웃 블록일 수 있다. `h(S)`를 `S`의 복잡도 함수라고 정의하자. 예를 들어, `h(S)`가 `S`에서의 픽셀의 분산, 재구성된 잔차의 분산, 또는 `S`에서의 에지의 배향 및 강도일 수 있다. `S`에 대한 움직임 벡터 정확도 `q_mv`의 값이 `h(S)`에 기초하여 선택된다. 예를 들어, 콘텐츠가 높은 복잡도를 가지며 `h(S)`가 클 때, `q_mv`의 값은 작다. 한편, 콘텐츠가 낮은 복잡도를 가지며 `h(S)`가 작을 때, `q_mv`의 값은 크다. 이 실시예에서, `q_mv`는 인코더에 의해 전송될 수 있거나(명시적 시그널링), 디코더에서 추론될 수 있다(암시적 시그널링).

[0073] 도 9를 참조하면, 명시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법이 전체적으로 참조 번호(900)로 나타내어져 있다. 방법(900)은 제어를 기능 블록(910)으로 넘기는 시작 블록(905)을 포함한다. 기능 블록(910)은 `motion_accuracy_adaptive_flag = 1`로 설정하고, `mv_accuracy_adaptation_mode = 3`으로 설정하며, `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 비트스트림에 기입하고, 제어를 루프 한계 블록(912)으로 넘긴다. 루프 한계 블록(912)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(915)으로 넘긴다. 기능 블록(915)은 움직임 추정을 수행하고, 제어를 기능 블록(920)으로 넘긴다. 기능 블록(920)은 국소 화상 영역에 관한 통계(예컨대, 픽셀 분산, 에지 배향, 강도 등)에 기초하여 움직임 벡터 정확도를 선택하고, 제어를 기능 블록(925)으로 넘긴다. 기능 블록(925)은 선택된 정확도에 기초하여 움직임 벡터를 양자화하고, 제어를 기능 블록(930)으로 넘긴다. 기능 블록(930)은 움직임 벡터 정확도를 (예컨대, 대응하는 디코더로) 전송하고, 제어를 기능 블록(935)으로 넘긴다. 기능 블록(935)은 움직임 보상을 수행하고, 제어를 기능 블록(940)으로 넘긴다. 기능 블록(940)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(942)으로 넘긴다. 루프 한계 블록(942)은 루프를 종료하고, 제어를 종료 블록(999)으로 넘긴다.

[0074] 도 10을 참조하면, 명시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(1000)로 나타내어져 있다. 방법(1000)은 제어를 기능 블록(1010)으로 넘기는 시작 블록(1005)을 포함한다. 기능 블록(1010)은 `motion_accuracy_adaptive_flag` 및 `mv_accuracy_adaptation_mode`를 파싱하고, 제어를 루프 한계 블록(1012)으로 넘긴다. 루프 한계 블록(1012)은 1 내지 블록의 수(#)의 범위를 가지는 변수 `I`를 사용하여 루프를 시작하고, 제어를 기능 블록(1013)으로 넘긴다. 기능 블록(1013)은 움직임 벡터(MV) 구문을 파싱하고, 제어를

결정 블록(1015)으로 넘긴다. 결정 블록(1015)은 $\text{motion_accuracy_adaptive_flag} == 1$ 이고 $\text{mv_accuracy_adaptation_mode} == 3$ 인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(1020)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(1017)으로 넘어간다. 기능 블록(1020)은 움직임 벡터 정확도를 파싱하고, 제어를 기능 블록(1025)으로 넘긴다. 기능 블록(1025)은 수신된 움직임 벡터 인덱스로부터 움직임 벡터를 재구성하고, 제어를 기능 블록(1030)으로 넘긴다. 기능 블록(1030)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(1032)으로 넘긴다. 루프 한계 블록(1032)은 루프를 종료하고, 제어를 종료 블록(1099)으로 넘긴다. 기능 블록(1017)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(1030)으로 넘긴다.

[0075] 도 11을 참조하면, 암시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법이 전체적으로 참조 번호(1100)로 나타내어져 있다. 방법(1100)은 제어를 기능 블록(1110)으로 넘기는 시작 블록(1105)을 포함한다. 기능 블록(1110)은 $\text{motion_accuracy_adaptive_flag} = 1$ 로 설정하고, $\text{mv_accuracy_adaptation_mode} = 4$ 로 설정하며, $\text{motion_accuracy_adaptive_flag}$ 및 $\text{mv_accuracy_adaptation_mode}$ 를 비트스트림에 기입하고, 제어를 루프 한계 블록(1112)으로 넘긴다. 루프 한계 블록(1112)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(1115)으로 넘긴다. 기능 블록(1115)은 움직임 추정을 수행하고, 제어를 기능 블록(1120)으로 넘긴다. 기능 블록(1120)은 국소 화상 영역의 통계(예컨대, 재구성된 화상의 분산 등)에 기초하여 움직임 벡터 정확도를 선택하고, 제어를 기능 블록(1125)으로 넘긴다. 기능 블록(1125)은 선택된 정확도에 기초하여 움직임 벡터 성분을 양자화하고, 제어를 기능 블록(1130)으로 넘긴다. 기능 블록(1130)은 움직임 보상을 수행하고, 제어를 기능 블록(1135)으로 넘긴다. 기능 블록(1135)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(1137)으로 넘긴다. 루프 한계 블록(1137)은 루프를 종료하고, 제어를 종료 블록(1199)으로 넘긴다.

[0076] 도 12를 참조하면, 암시적 시그널링을 갖는 비디오 콘텐츠에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(1200)로 나타내어져 있다. 방법(1200)은 제어를 기능 블록(1210)으로 넘기는 시작 블록(1205)을 포함한다. 기능 블록(1210)은 $\text{motion_accuracy_adaptive_flag}$ 및 $\text{mv_accuracy_adaptation_mode}$ 를 파싱하고, 제어를 루프 한계 블록(1212)으로 넘긴다. 루프 한계 블록(1212)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(1213)으로 넘긴다. 기능 블록(1213)은 움직임 벡터(MV) 구문을 파싱하고, 제어를 결정 블록(1215)으로 넘긴다. 결정 블록(1215)은 $\text{motion_accuracy_adaptive_flag} == 1$ 이고 $\text{mv_accuracy_adaptation_mode} == 4$ 인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(1220)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(1217)으로 넘어간다. 기능 블록(1220)은 국소 화상 영역의 통계로부터 움직임 벡터 정확도를 획득하고, 제어를 기능 블록(1225)으로 넘긴다. 기능 블록(1225)은 수신된 움직임 벡터 인덱스로부터 움직임 벡터를 재구성하고, 제어를 기능 블록(1230)으로 넘긴다. 기능 블록(1230)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(1232)으로 넘긴다. 루프 한계 블록(1232)은 루프를 종료하고, 제어를 종료 블록(1299)으로 넘긴다. 기능 블록(1217)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(1230)으로 넘긴다.

[0077] 움직임 벡터 진폭 적응

[0078] 비디오 코딩에서, 앞서 정의한 바와 같이, 검색된 움직임 벡터와 예측된 움직임 벡터 사이의 차이(mvd)가 인코딩된다. 효과적인 움직임 추정이 비디오 인코더에 의해 이용되는 경우, 대부분은 mvd의 값이 아주 작다. 그렇지만, 비디오 블록이 움직임을 검색을 위한 양호한 특징을 갖고 있지 않을 때, 검색된 mv는 신뢰할 수 없으며, 어떤 랜덤성을 나타낸다. 이 경우에, mvd의 진폭은 꽤 클 수 있다. 검색된 mv가 신뢰할 수 없기 때문에, mvd 코딩을 위한 너무 많은 비트는 필요하지 않다. mvd가 큰 진폭을 가질 때, 비트를 절감하기 위해 그것이 대략적으로 양자화되는 것(낮은 움직임 벡터 정확도)을 선호한다.

[0079] 일 실시예에서, 움직임 벡터 정확도 q_{mv} 는 움직임 벡터 차이 mvd의 진폭의 함수 $q_{mv}=f(|mvd|)$ 이다. 한 예시적인 정확도 함수 f는 $x \leq T$ 일 때 $f(x)=0.25$ 일 수 있고, $x > T$ 일 때 0.5(단, T는 임계값임)일 수 있다.

[0080] 이 예에서, mvd의 양자화는 다음과 같다:

$$Idx_mvd = \begin{cases} sign(mvd) * rounding(|mvd|/0.25) & (|mvd| \leq T \text{ 인 경우}) \\ sign(mvd) * rounding(|mvd - T|/0.5 + T/0.25) & (|mvd| > T \text{ 인 경우}) \end{cases}$$

[0081]

[0082] 여기서 Idx_mvd는 mvd의 양자화 인덱스이다.

[0083] mvd의 재구성, mvq는 다음과 같다:

$$mvq = \begin{cases} Idx_mvd * 0.25 & (|Idx_mvd| \leq T/0.25 \text{ 인 경우}) \\ (Idx_mvd - T/0.25) * 0.5 + T/0.25 & (|Idx_mvd| > T/0.25 \text{ 인 경우}) \end{cases}$$

[0084]

[0085] 도 13을 참조하면, 움직임 벡터 진폭에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 인코딩하는 예시적인 방법이 전체적으로 참조 번호(1300)로 나타내어져 있다. 방법(1300)은 제어를 기능 블록(1310)으로 넘기는 시작 블록(1305)을 포함한다. 기능 블록(1310)은 motion_accuracy_adaptive_flag = 1로 설정하고, mv_accuracy_adaptation_mode = 5로 설정하며, 제어를 루프 한계 블록(1312)으로 넘긴다. 루프 한계 블록(1312)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(1315)으로 넘긴다. 기능 블록(1315)은 움직임 추정을 수행하고, 제어를 기능 블록(1320)으로 넘긴다. 기능 블록(1320)은 움직임 벡터의 진폭에 기초하여 정확도 함수를 선택하고, 제어를 기능 블록(1325)으로 넘긴다. 기능 블록(1325)은 선택된 함수에 기초하여 움직임 벡터 성분을 양자화하고, 제어를 기능 블록(1330)으로 넘긴다. 기능 블록(1330)은 움직임 보상을 수행하고, 제어를 기능 블록(1335)으로 넘긴다. 기능 블록(1335)은 엔트로피 인코딩을 수행하고, 제어를 루프 한계 블록(1337)으로 넘긴다. 루프 한계 블록(1337)은 루프를 종료하고, 제어를 종료 블록(1399)으로 넘긴다.

[0086]

도 14를 참조하면, 암시적 시그널링을 갖는 움직임 정보 진폭에 기초한 움직임 정보의 적응적 코딩을 사용하여 화상 데이터를 디코딩하는 예시적인 방법이 전체적으로 참조 번호(1400)로 나타내어져 있다. 방법(1400)은 제어를 기능 블록(1410)으로 넘기는 시작 블록(1405)을 포함한다. 기능 블록(1410)은 motion_accuracy_adaptive_flag 및 mv_accuracy_adaptation_mode를 파싱하고, 제어를 루프 한계 블록(1412)으로 넘긴다. 루프 한계 블록(1412)은 1 내지 블록의 수(#)의 범위를 가지는 변수 I를 사용하여 루프를 시작하고, 제어를 기능 블록(1413)으로 넘긴다. 기능 블록(1413)은 움직임 벡터(MV) 구문을 파싱하고, 제어를 결정 블록(1415)으로 넘긴다. 결정 블록(1415)은 motion_accuracy_adaptive_flag == 1이고 mv_accuracy_adaptation_mode == 5인지 여부를 판정한다. 그러한 경우, 제어가 기능 블록(1420)으로 넘어간다. 그렇지 않은 경우, 제어가 기능 블록(1417)으로 넘어간다. 기능 블록(1420)은 수신된 움직임 벡터 인덱스의 값으로부터 움직임 벡터 정확도 함수를 획득하고, 제어를 기능 블록(1425)으로 넘긴다. 기능 블록(1425)은 (수신된 움직임 벡터 인덱스로부터) 움직임 벡터를 재구성하고, 제어를 기능 블록(1430)으로 넘긴다. 기능 블록(1430)은 움직임 보상을 수행하고, 제어를 루프 한계 블록(1432)으로 넘긴다. 루프 한계 블록(1432)은 루프를 종료하고, 제어를 종료 블록(1499)으로 넘긴다. 기능 블록(1417)은 균일한 정확도 또는 다른 적응적 정확도 방법을 사용하여 움직임 벡터를 재구성하고, 제어를 기능 블록(1430)으로 넘긴다.

[0087] 구문

[0088] 표 2는 본 발명의 일 실시예에 따른 예시적인 화상 및 슬라이스 헤더 구문을 보여주고 있다.

표 2

picture_header() {	기술자
...	
motion_accuracy_adaptive_flag	u(1)
...	
}	
slice_header() {	
...	
if (motion_accuracy_adaptive_flag){	
mv_accuracy_adaptation_mode	u(3)
if (mv_accuracy_adaptation_mode != 1){	
q_mv_signaling	u(1)
if(q_mv_signaling)	
q_mv	u(v)
}	
if (mv_accuracy_adaptation_mode == 1){	
res_mv_signaling	u(1)
if (res_mv_signaling){	
res_x	u(2)
res_y	u(2)
}	
}	
}	
....	
}	

[0089]

[0090]

표 2에서 구문 요소들 중 일부의 의미는 다음과 같다:

[0091]

motion_accuracy_adaptive_flag는 화상에 대해 움직임 벡터 정확도 적응이 사용되는지 여부를 지정한다. motion_accuracy_adaptive_flag = 1은 화상에서 움직임 정확도 적응 방식이 사용된다는 것을 나타내고, motion_accuracy_adaptive_flag = 0은 화상에서 움직임 정확도 적응 방식이 사용되지 않는다는 것을 나타낸다.

[0092]

mv_accuracy_adaptation_mode는 슬라이스에 대해 사용되는 움직임 벡터 정확도 적응 방식을 지정한다. mv_accuracy_adaptation_mode = 0은 파티션 크기 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다. mv_accuracy_adaptation_mode = 1은 방향 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다. mv_accuracy_adaptation_mode = 2는 QP 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다. mv_accuracy_adaptation_mode = 3은 명시적 시그널링을 갖는 콘텐츠 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다. mv_accuracy_adaptation_mode = 4는 암시적 시그널링을 갖는 콘텐츠 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다. mv_accuracy_adaptation_mode = 5는 진폭 기반 움직임 벡터 정확도 적응이 인에이블되어 있다는 것을 나타낸다.

[0093]

q_mv는 기본 양자화 스텝 크기에 부가하여 움직임 벡터를 양자화하는 데 사용되는 양자화 스텝을 지정한다.

[0094]

res_x는 움직임 벡터의 수평 성분의 정확도를 지정한다.

[0095]

res_y는 움직임 벡터의 수직 성분의 정확도를 지정한다.

[0096]

q_mv_signaling은 명시적 또는 암시적 시그널링을 지정한다. q_mv_signaling = 1은 q_mv가 명시적으로 신호될 것임을 나타낸다. q_mv_signaling = 0은 q_mv가 명시적으로 신호되지 않을 것임을 나타낸다.

[0097]

res_mv_signaling은 res_x 및 res_y의 명시적 또는 암시적 시그널링을 지정한다. res_mv_signaling = 1은

res_x 및 res_y가 명시적으로 신호될 것임을 나타낸다. res_mv_signaling = 0은 res_x 및 res_y가 명시적으로 신호되지 않을 것임을 나타낸다.

- [0098] 이제부터 본 발명의 많은 부수적인 이점들/특징들 중 몇몇에 대한 설명이 제공될 것이며, 이들 중 몇몇은 이상에서 언급하였다. 예를 들어, 한 가지 이점/특징은 움직임 벡터를 사용하여 적어도 화상에서의 블록을 인코딩하는 인코더를 갖는 장치이다. 적응적 움직임 벡터 정확도 방식은 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도를 선택하는 데 사용된다. 움직임 벡터에 대한 정확도를 선택하는 선택 기준은 비레이트-왜곡 기반 기준을 포함한다.
- [0099] 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 선택 기준은 움직임 보상 파티션 크기를 포함한다.
- [0100] 또 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 선택 기준은 움직임 벡터 성분 방향을 포함하고, 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도는 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하도록 선택되며, 수직 성분 및 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 우세 성분으로서 선택된다.
- [0101] 또 다른 이점/특징은 인코더를 갖는 장치이며, 선택 기준은 움직임 벡터 성분 방향을 포함하고, 블록을 인코딩하는 데 사용되는 움직임 벡터의 정확도는 움직임 벡터의 수평 성분과 비교하여 수직 성분에서 상이하도록 선택되며, 수직 성분 및 수평 성분 중에서 가장 큰 정확도를 갖는 성분이 상기한 바와 같이 우세 성분으로서 선택되고, 우세 성분은,
- [0102] (i) 움직임 벡터가 정수 움직임 벡터일 때, 움직임 벡터의 진폭,
- [0103] (ii) 블록에 대한 움직임 보상 파티션의 형상,
- [0104] (iii) 블록에 대한 예측된 움직임 벡터,
- [0105] (iv) 블록과 관련한 이웃 블록의 움직임 벡터, 및
- [0106] (v) 화상 및 하나 이상의 다른 화상 중 적어도 하나의 화상에 관한 전역 움직임 정보 - 화상 및 하나 이상의 다른 화상은 동일한 비디오 시퀀스에 포함되어 있음 - 중 적어도 하나에 따라 결정된다.
- [0107] 또 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 선택 기준은 블록의 인코딩 양자화 파라미터를 포함한다.
- [0108] 게다가, 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 선택 기준은 국소 화상 영역의 통계를 포함하고, 국소 화상 영역은 화상의 일부분, 화상, 및 하나 이상의 다른 화상 중 적어도 하나에 대응하며, 화상 및 하나 이상의 다른 화상은 동일한 비디오 시퀀스에 포함되어 있다.
- [0109] 또한, 다른 이점/특징은 인코더를 갖는 장치이며, 선택 기준은 국소 화상 영역의 통계를 포함하고, 국소 화상 영역은 화상의 일부분, 화상, 및 하나 이상의 다른 화상 중 적어도 하나에 대응하며, 화상 및 하나 이상의 다른 화상은 상기한 바와 같이 동일한 비디오 시퀀스에 포함되어 있고, 국소 화상 영역의 통계는,
- [0110] (i) 국소 영역에서의 픽셀 분산,
- [0111] (ii) 국소 영역에서의 디코딩된 잔차 계수의 분산,
- [0112] (iii) 국소 영역에서의 에지 배향의 분산, 및
- [0113] (iv) 국소 영역에서의 에지 강도의 분산 중 적어도 하나로부터 선택된다.
- [0114] 그에 부가하여, 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 선택 기준은 검색된 움직임 벡터의 진폭을 포함한다.
- [0115] 게다가, 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 상기 블록을 인코딩하는 데 사용된 상기 움직임 벡터의 정확도가 인코딩된 비트스트림으로 명시적으로 신호된다.
- [0116] 게다가, 다른 이점/특징은 이상에서 기술한 인코더를 갖는 장치이며, 블록을 인코딩하는 데 사용된 움직임 벡터의 정확도가 화상에서의 또는 화상을 포함하는 시퀀스에서의 이전에 디코딩된 비디오로부터 추론된다.
- [0117] 본 발명의 이들 및 기타 특징들 및 이점들은 본 명세서에서의 개시 내용에 기초하여 당업자에 의해 용이하게 확 인될 수 있다. 본 발명의 개시 내용이 다양한 형태의 하드웨어, 소프트웨어, 펌웨어, 특수 목적 프로세서, 또

는 이들의 조합으로 구현될 수 있다는 것을 잘 알 것이다.

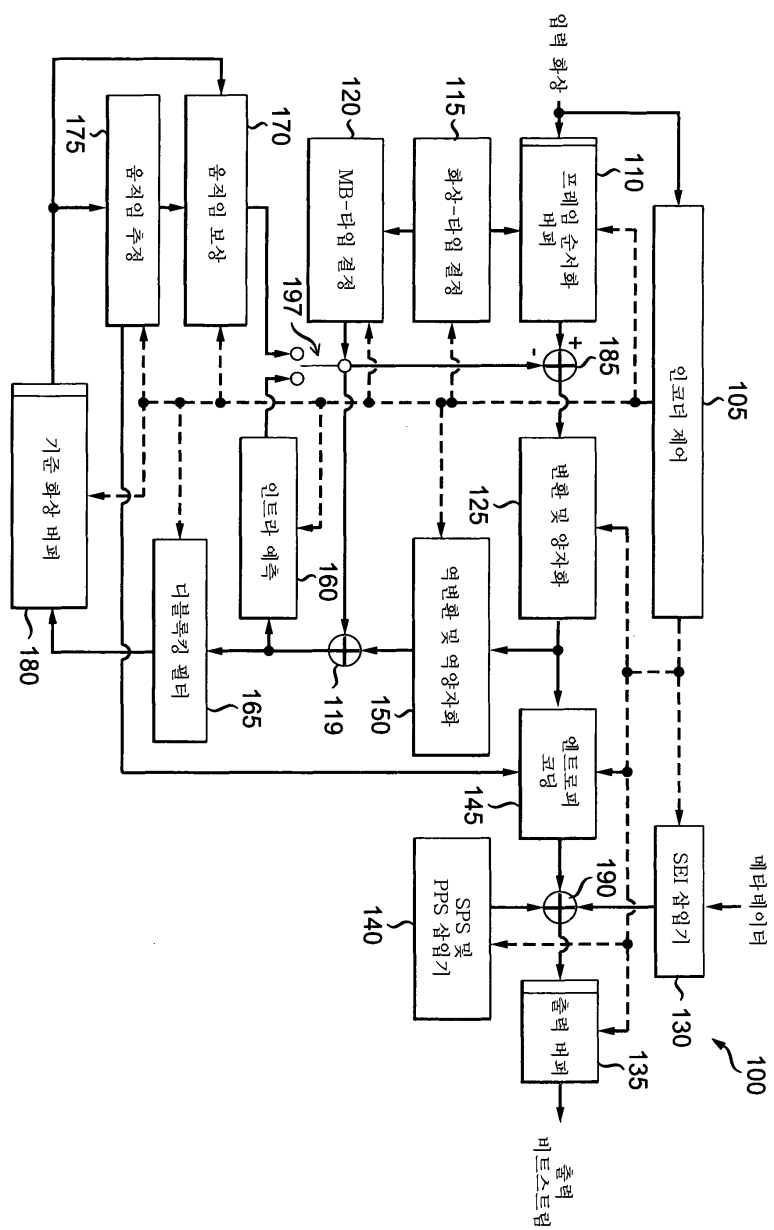
[0118] 가장 양호하게는, 본 발명의 개시 내용이 하드웨어와 소프트웨어의 조합으로서 구현된다. 게다가, 소프트웨어는 프로그램 저장 장치에 유형으로 구현되는 애플리케이션 프로그램으로서 구현될 수 있다. 이 애플리케이션 프로그램은 임의의 적당한 아키텍처를 포함하는 기계에 업로드되어 이 기계에 의해 실행될 수 있다. 양호하게는, 이 기계는 하나 이상의 중앙 처리 장치(CPU), 랜덤 액세스 메모리(RAM), 및 입/출력(I/O) 인터페이스와 같은 하드웨어를 갖는 컴퓨터 플랫폼 상에 구현된다. 이 컴퓨터 플랫폼은 또한 운영 체제 및 마이크로명령어 코드(microinstruction code)도 포함할 수 있다. 본 명세서에 기술된 다양한 프로세스 및 기능이, CPU에 의해 실행될 수 있는, 마이크로명령어 코드의 일부이거나 애플리케이션 프로그램의 일부이거나, 이들의 임의의 조합일 수 있다. 그에 부가하여, 부가의 데이터 저장 장치 및 인쇄 장치와 같은 다양한 다른 주변 장치들이 컴퓨터 플랫폼에 연결되어 있을 수 있다.

[0119] 또한, 첨부 도면에 도시된 시스템 구성요소들 및 방법들 중 일부가 양호하게는 소프트웨어로 구현되기 때문에, 시스템 구성요소들 또는 프로세스 기능 블록들 간의 실제 연결이 본 발명이 프로그램되는 방식에 따라 다를 수 있다는 것을 잘 알 것이다. 본 명세서의 개시 내용을 살펴보고, 당업자라면 본 발명의 이들 및 유사한 구현들 또는 구성들을 생각할 수 있을 것이다.

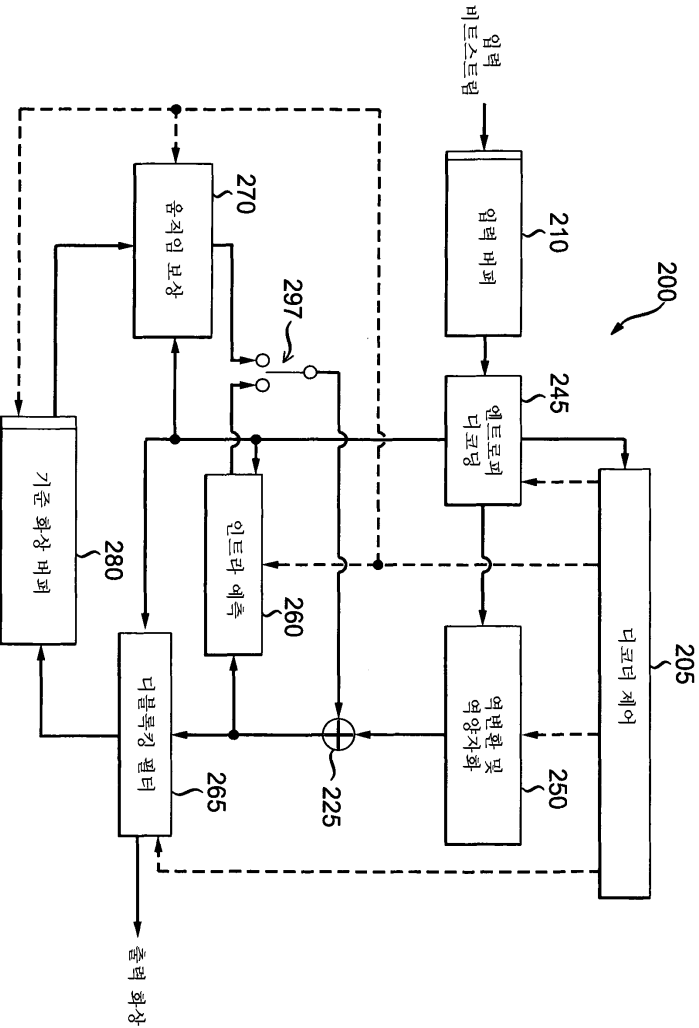
[0120] 예시적인 실시예들이 첨부 도면을 참조하여 본 명세서에 기술되어 있지만, 본 발명이 이들 정확한 실시예로 제한되지 않고 또 다양한 변경 및 수정이 본 발명의 범위 또는 사상을 벗어나지 않고 당업자에 의해 행해질 수 있다는 것을 잘 알 것이다. 모든 이러한 변경 및 수정이 첨부된 청구항들에 기술된 본 발명의 범위 내에 포함되는 것으로 보아야 한다.

도면

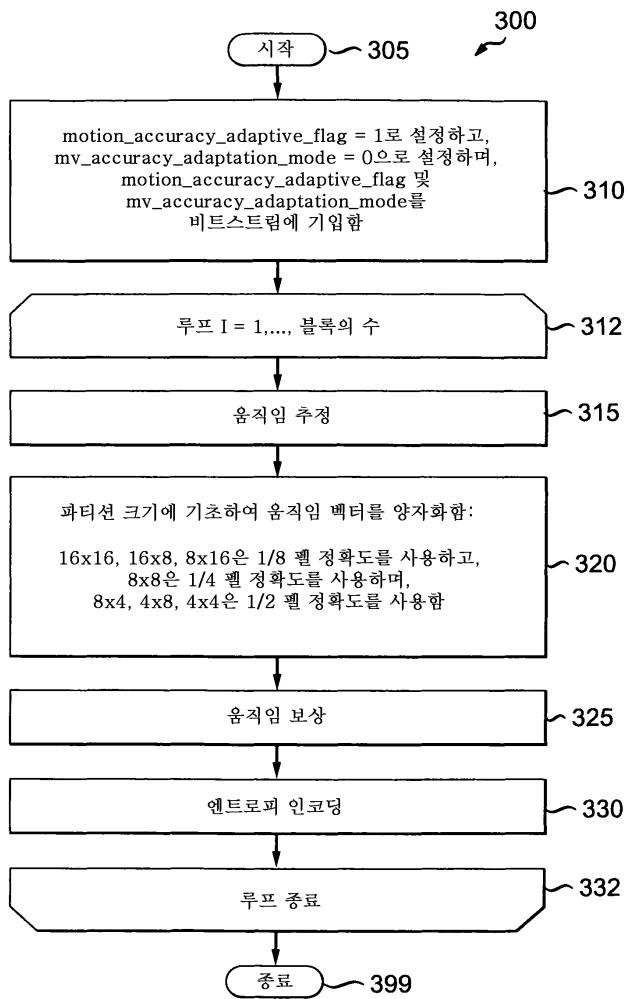
도면1



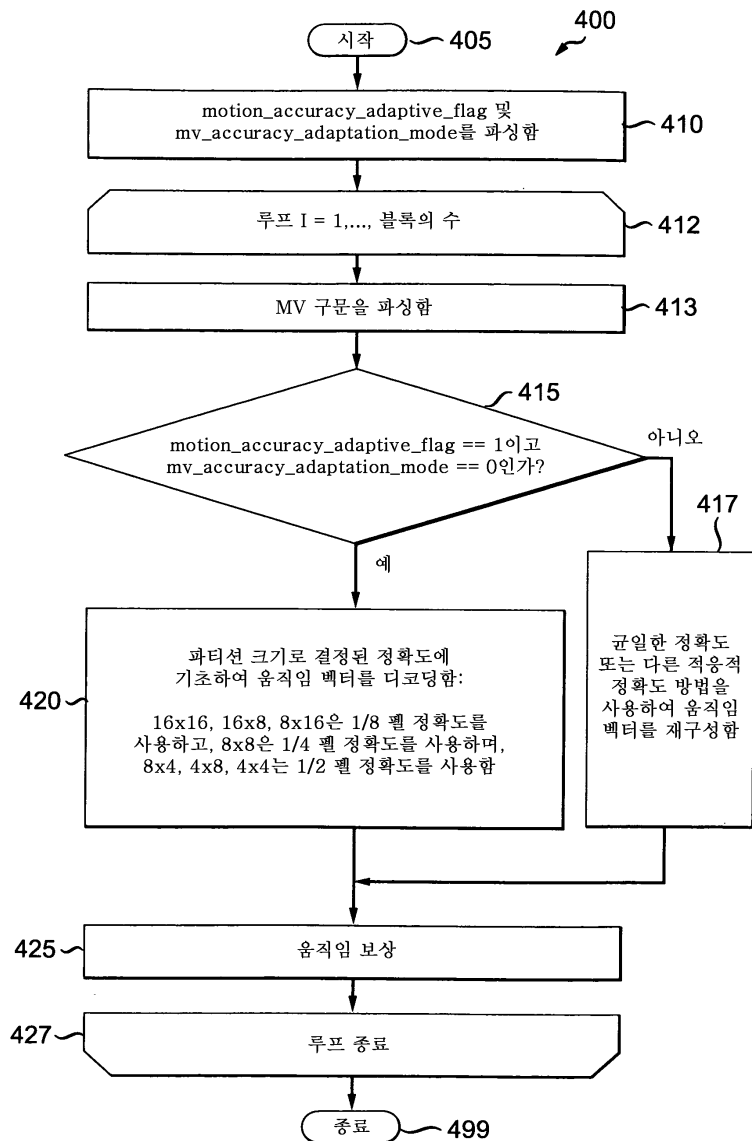
도면2



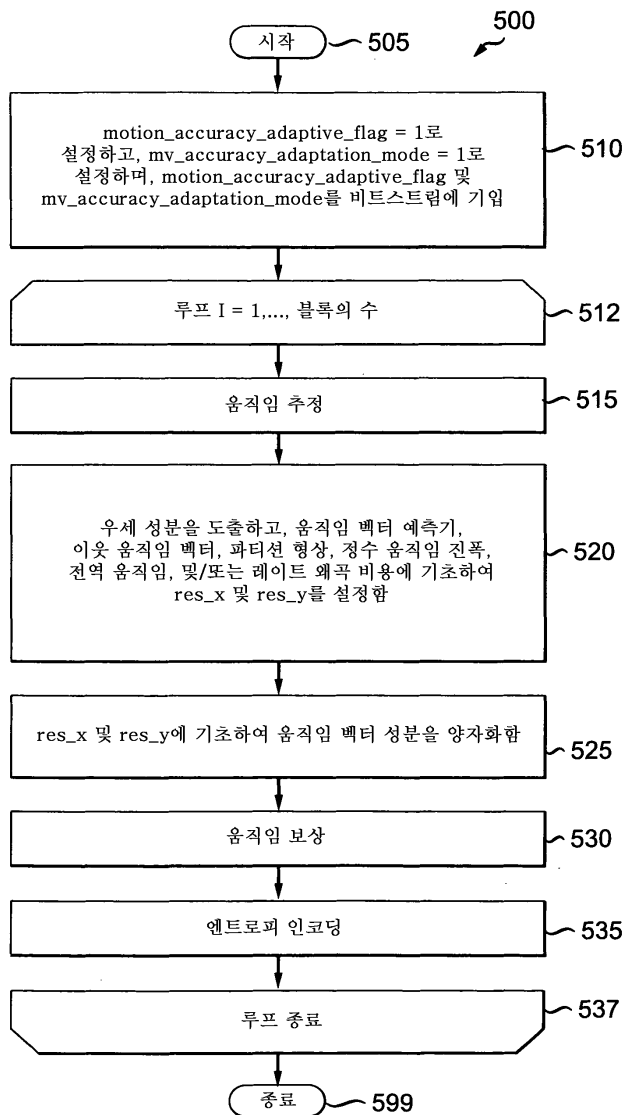
도면3



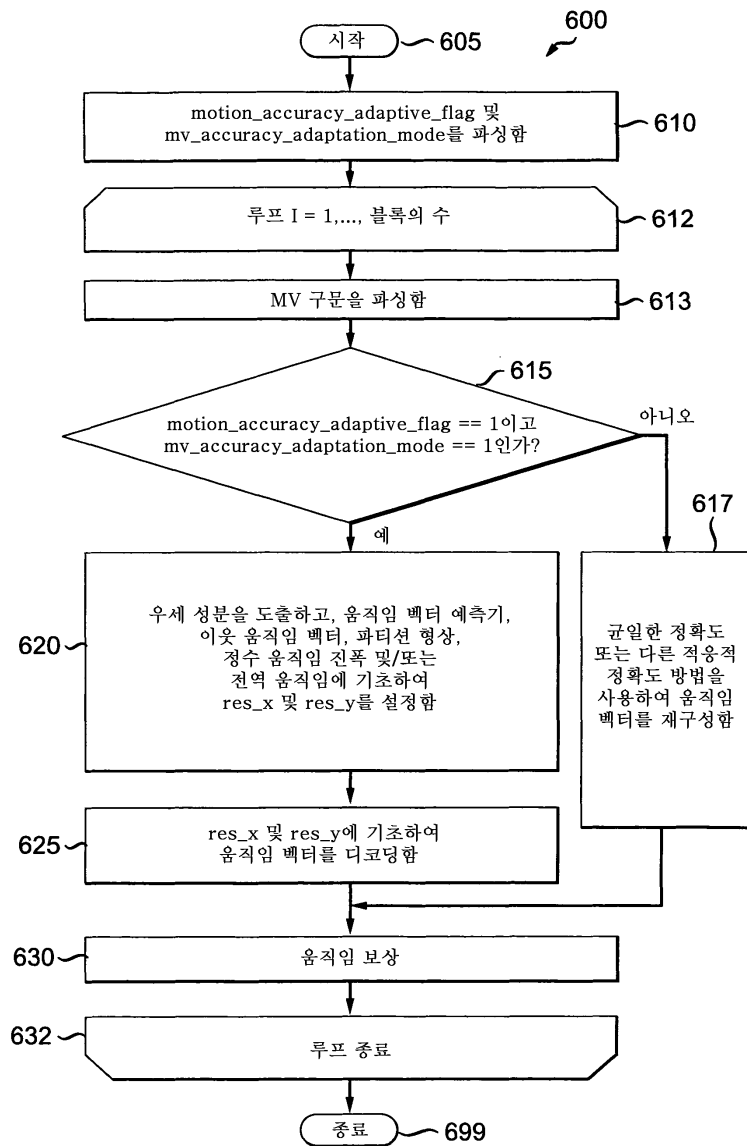
도면4



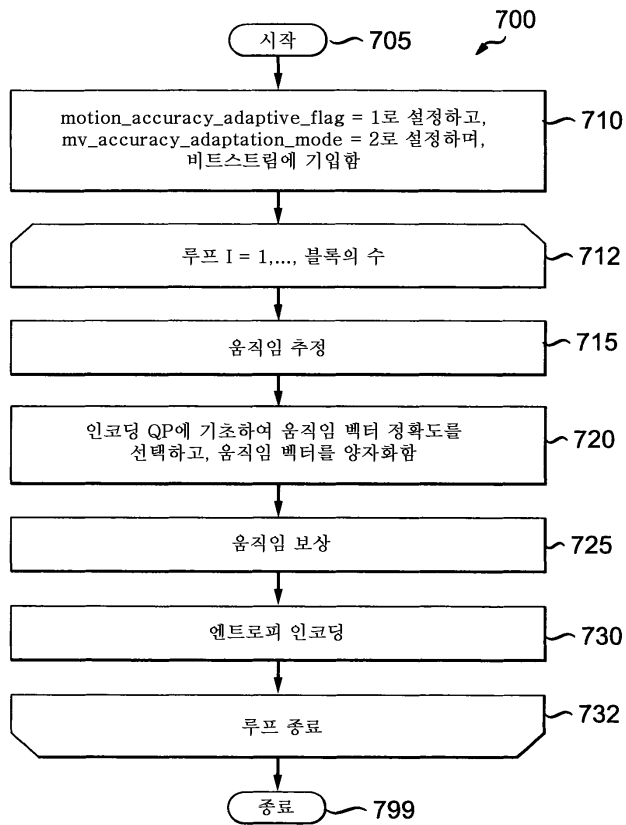
도면5



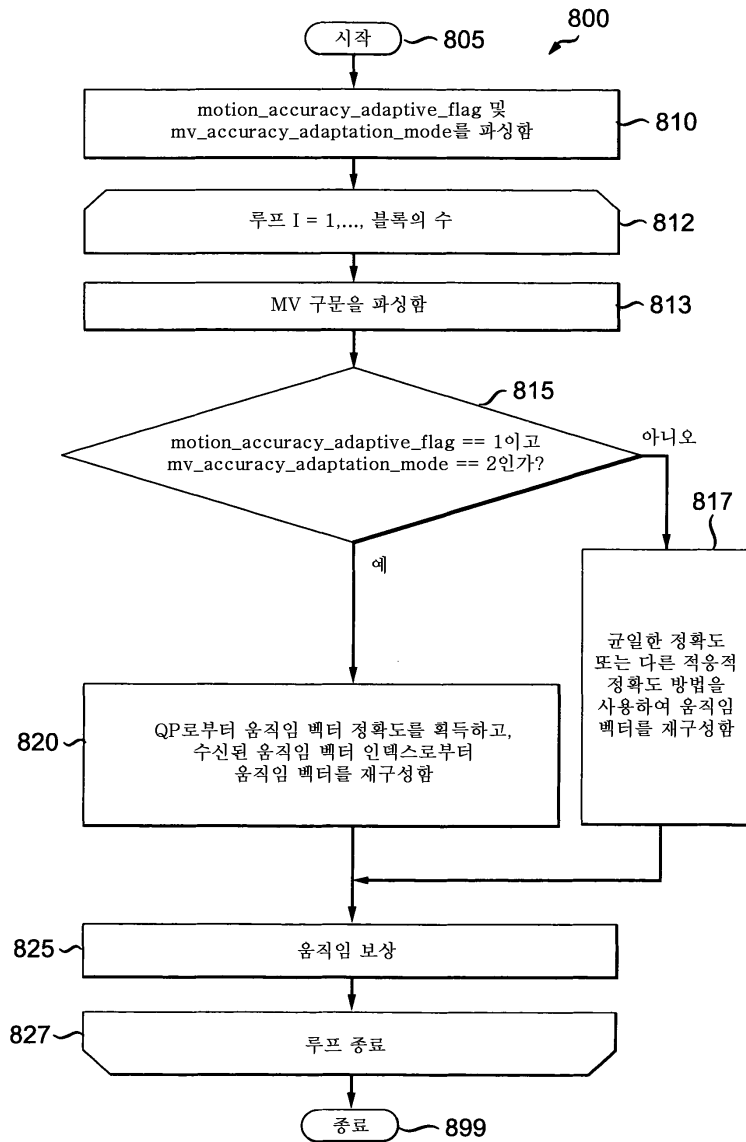
도면6



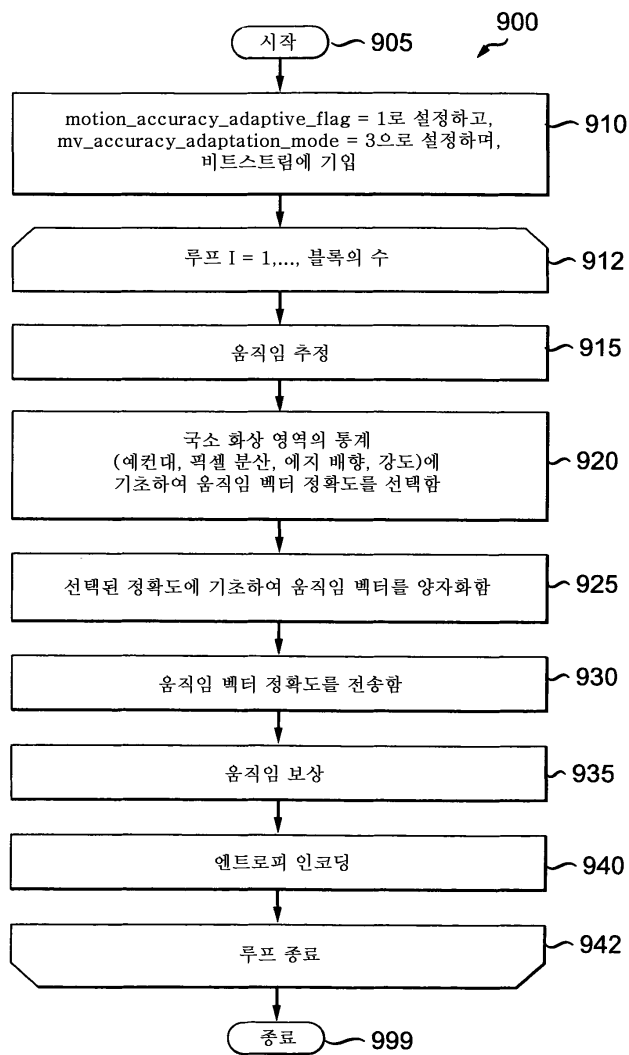
도면7



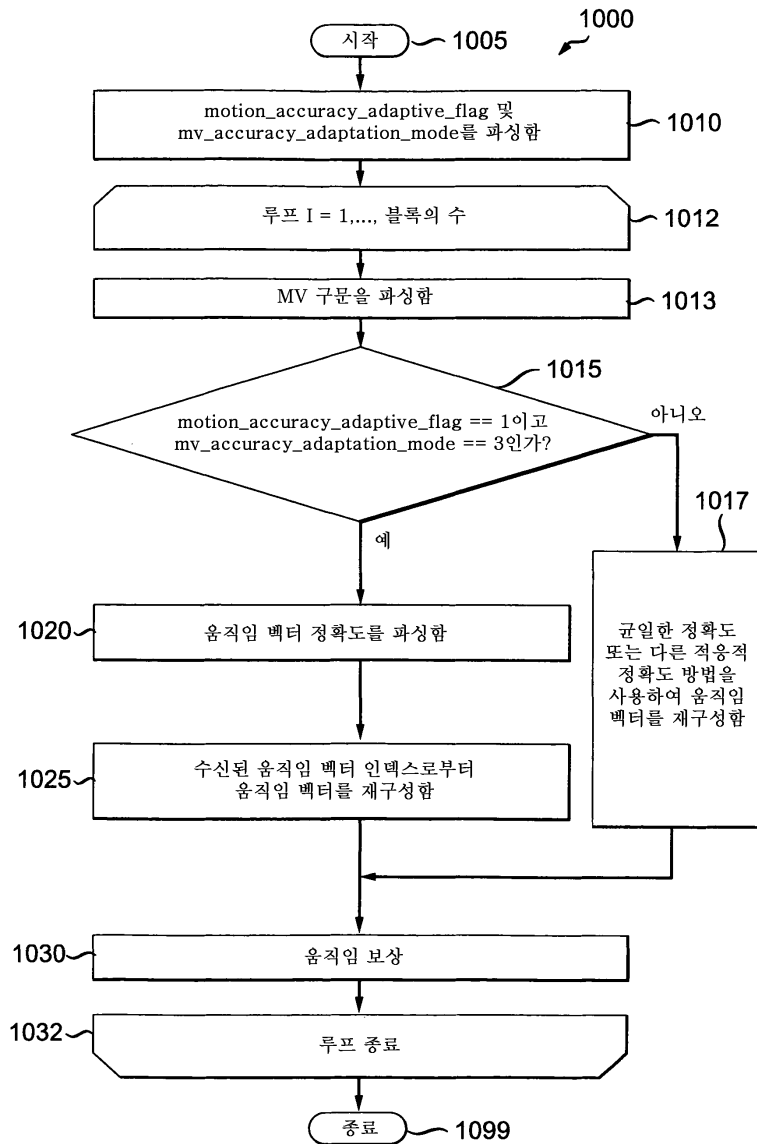
도면8



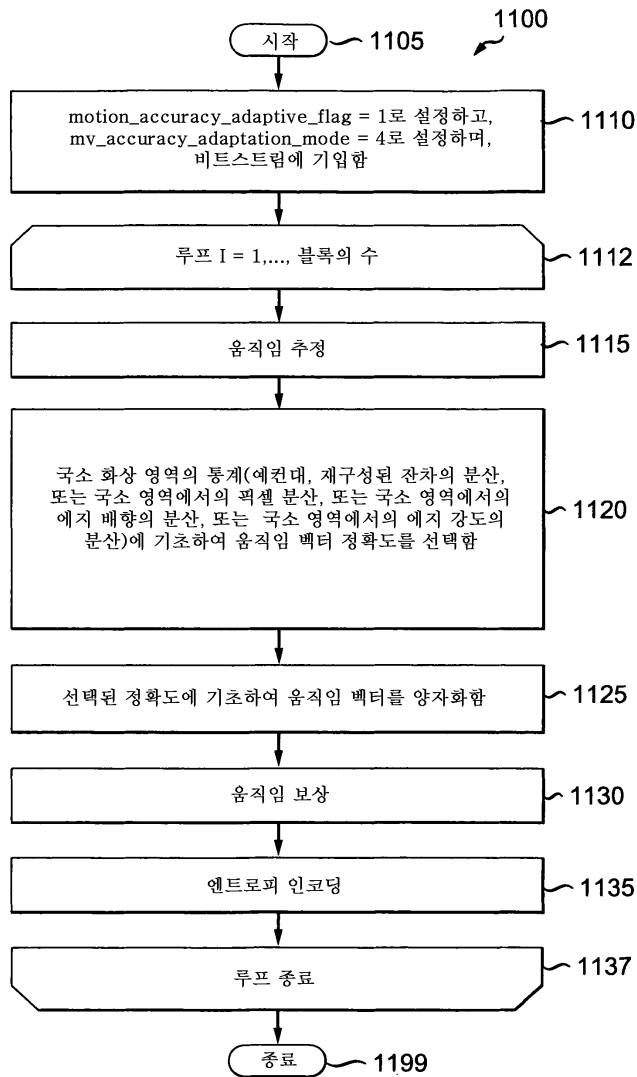
도면9



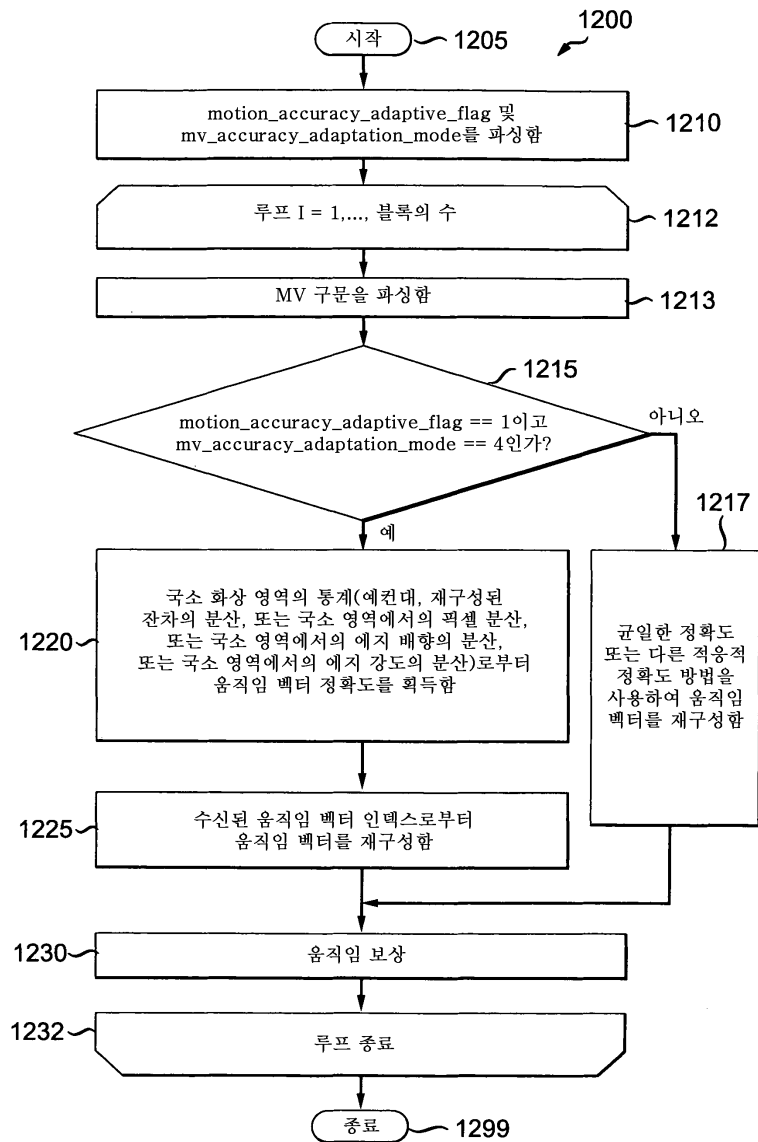
도면10



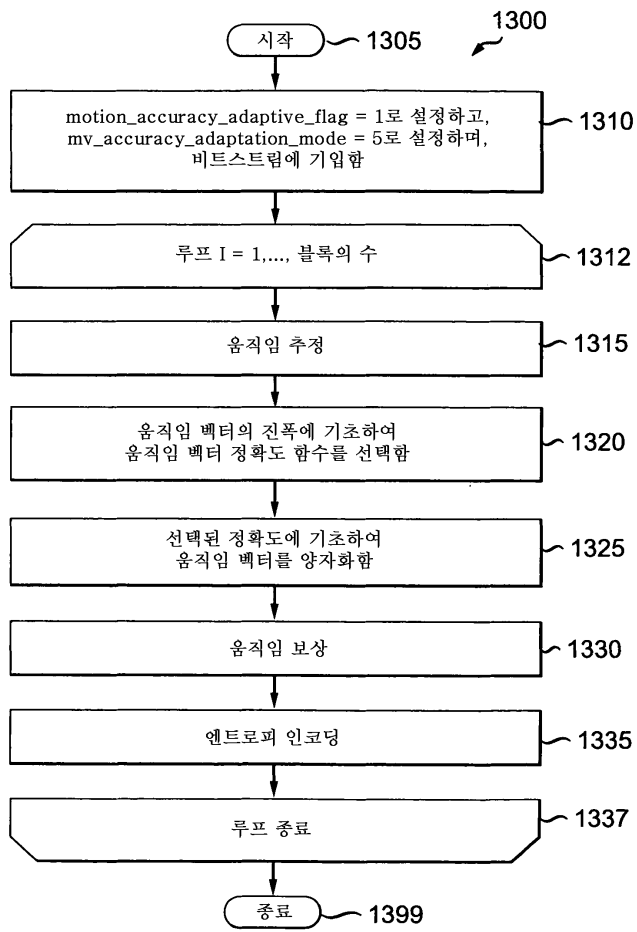
도면11



도면12



도면13



도면14

