



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0107160
(43) 공개일자 2024년07월08일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) H04N 19/176 (2014.01)
H04N 19/184 (2014.01) H04N 19/60 (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/176 (2015.01)
(21) 출원번호 10-2024-7018912
(22) 출원일자(국제) 2022년11월17일
심사청구일자 2024년06월05일
(85) 번역문제출일자 2024년06월05일
(86) 국제출원번호 PCT/US2022/080063
(87) 국제공개번호 WO 2023/092019
국제공개일자 2023년05월25일
(30) 우선권주장
63/281,381 2021년11월19일 미국(US)

(71) 출원인
바이트댄스 아이엔씨
미국, 90066 캘리포니아 로스앤젤레스 스위트룸
137호 6층 제퍼슨 블러바드 웨스트 12655
(72) 발명자
왕, 예-쿠이
미국 90066 캘리포니아 로스 앤젤레스 웨스트 제
퍼슨 블러바드 12655 6층 스위트 넘버 137
(74) 대리인
양영준, 임규빈, 백만기

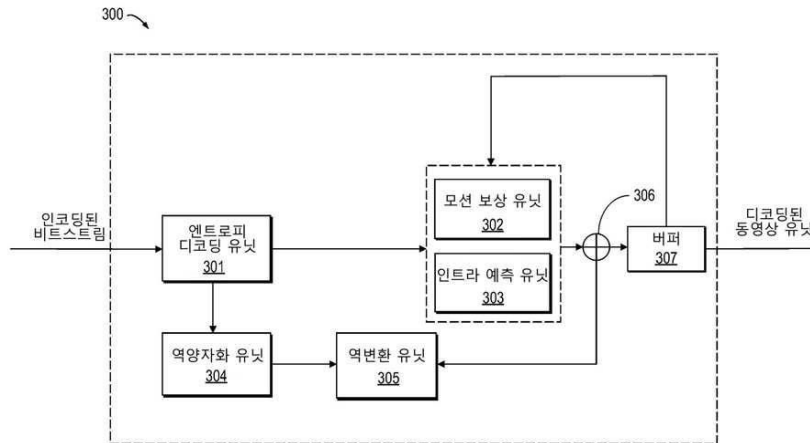
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 동영상 처리 방법, 장치 및 매체

(57) 요약

본 개시의 실시예는, 동영상 처리를 위한 해결책을 제공한다. 동영상 처리 방법을 제안한다. 방법은, 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 동영상의 현재 동영상 블록과 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함하고, GCI는 하나 이상의 구문 요소를 포함하며, 여기서, 하나 이상의 구문 요소 각각은 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다. 제안된 방법은, 종래의 해결책에 비해 코딩 효율성을 향상시킬 수 있는 장점이 있다.

대표도



도3

(52) CPC특허분류

H04N 19/184 (2015.01)

H04N 19/60 (2015.01)

명세서

청구범위

청구항 1

동영상 처리 방법으로서,

상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함하고,

상기 GCI는,

하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs remaining` 및 `dec abs level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual ts coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice header`에 지시되는지 여부를 지시하는 제3 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs remaining` 및 `dec abs level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs remaining` 및 `dec abs level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는

하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh reverse last sig coeff flag`가 상기 SPS를 참조하는 구문 구조 `slice header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함하며,

상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는, 방법.

청구항 2

제1항에 있어서,

상기 제1 미리 결정된 값은 0인, 방법.

청구항 3

제1항 내지 제2항 중 어느 한 항에 있어서,

상기 제1 구문 요소는 구문 요소 `gci all rap pictures constraint flag`이고,

상기 제2 구문 요소는 구문 요소 `gci no extended precision processing constraint flag`이며,

상기 제3 구문 요소는 구문 요소 `gci no ts residual coding rice constraint flag`이고,

상기 제4 구문 요소는 구문 요소 `gci no rrc rice extension constraint flag`이며,

상기 제5 구문 요소는 구문 요소 `gci no persistent rice adaptation constraint flag`이고, 및

상기 제6 구문 요소는 구문 요소 `gci no reverse last sig coeff constraint flag`인, 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 GCI는 제7 구문 요소와 제8 구문 요소를 추가로 포함하고, 상기 제7 구문 요소는 GCI에 대한 구문 구조에

상기 GCI 구문 요소가 존재하는지 여부를 지시하며, 상기 제8 구문 요소는 구문 요소 gci alignment zero bit 이외의 구문 구조에서 추가 GCI 비트 수를 지시하고, 상기 제7 구문 요소는 제2 미리 결정된 값과 동일한 값을 가지며, 상기 제8 구문 요소는 제3 미리 결정된 값과 동일한 값을 갖는, 방법.

청구항 5

제4항에 있어서,

상기 제7 구문 요소는 구문 요소 gci present flag이고, 상기 제8 구문 요소는 구문 요소 gci num additional bits이며, 상기 제2 미리 결정된 값은 1이고, 상기 제3 미리 결정된 값은 0인, 방법.

청구항 6

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 GCI은 상기 제1 구문 요소, 제2 구문 요소, 제3 구문 요소, 제4 구문 요소, 제5 구문 요소 및 제6 구문 요소와 다른 제9 구문 요소를 추가로 포함하며, 상기 제9 구문 요소는 상기 비트스트림에 없고, 상기 제1 미리 결정된 값과 동일한 값을 갖는, 방법.

청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 변환은 상기 현재 동영상 블록을 상기 비트스트림으로 인코딩하는 단계를 포함하는, 방법.

청구항 8

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 변환은 상기 비트스트림으로부터 상기 현재 동영상 블록을 디코딩하는 단계를 포함하는, 방법.

청구항 9

프로세서 및 명령을 포함하는 비일시적 메모리를 포함하는 동영상 데이터를 처리하는 장치로서,

상기 명령은, 상기 프로세서에 의해 실행될 때 상기 프로세서로 하여금 제1항 내지 제8항 중 어느 한 항에 따른 방법을 수행하도록 하는, 장치.

청구항 10

프로세서로 하여금 제1항 내지 제8항 중 어느 한 항에 따른 방법을 수행하도록 하는 명령을 저장하는, 비일시적 컴퓨터 판독 가능 저장 매체.

청구항 11

동영상 처리 장치에 의해 수행되는 방법을 통해 생성된 동영상의 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능 기록 매체로서,

상기 방법은

상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함하고,

상기 GCI는,

하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 abs remaining 및 dec abs level의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 residual ts coding에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 slice header에 표시되는지 여부를 지시하는 제3 구문

요소,

하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 abs remaining 및 dec abs level의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여 각 TU의 시작에서 구문 요소 abs remaining 및 dec abs level의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는

하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 sh reverse last sig coeff flag가 상기 SPS를 참조하는 구문 구조 slice header에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함하며,

하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는, 비일시적 컴퓨터 판독 가능 기록매체.

청구항 12

동영상의 비트스트림을 저장하는 방법으로서,

상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계로서,

상기 GCI는,

하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 abs remaining 및 dec abs level의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 residual ts coding에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 slice header에 표시되는지 여부를 지시하는 제3 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 abs remaining 및 dec abs level의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소,

하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여 각 TU의 시작에서 구문 요소 abs remaining 및 dec abs level의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소 또는

하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 sh reverse last sig coeff flag가 상기 SPS를 참조하는 구문 구조 slice header에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함하는 단계; 및

상기 비트스트림을 비일시적 컴퓨터 판독 가능 매체에 저장하는 단계를 포함하며,

상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는, 방법.

발명의 설명

기술 분야

[0001] [관련 출원의 교차 참조]

[0002] 본 출원은 2021년 11월 19일자로 미국에 출원된 가출원 번호 제 63/281,381호에 대한 우선권을 주장한다. 선행 출원의 개시 내용 그 전체가 본 명세서에 참조로 포함된다.

[0003] [기술분야]

[0004] 본 개시의 실시예는, 일반적으로 동영상 코딩 기술에 관한 것이며, 특히 VVC 범위 확장 일반 제약 정보 플래그

의 시그널링에 관한 것이다.

배경 기술

[0005] 동영상 코딩 표준은 주로 잘 알려진 ITU-T 및 ISO/IEC 표준의 개발을 통해 발전해 왔다. ITU-T는 H.261과 H.263을 제작하였고, ISO/IEC는 MPEG-1과 MPEG-4 Visual을 제작하였으며, 두 조직은 공동으로 H.262/MPEG-2 Video, H.264/MPEG-4 고도 동영상 부호화(AVC) 및 H.265/HEVC 표준을 제작하였다. H.262 이후, 동영상 코딩 표준은 시간적 예측과 변환 코딩이 활용되는 하이브리드 동영상 코딩 구조를 기반으로 하고 있다. HEVC를 초과하는 미래의 동영상 코딩 기술을 탐구하기 위해, 2015년 VCEG와 MPEG이 공동으로 합동 동영상 탐험대(Joint Video Exploration Team, JVET)를 설립하였다. 그 이후로, JVET는 많은 새로운 방법을 채택하여 공통 탐사 모델(Joint Exploration Model, JEM)이라는 참조 소프트웨어에 추가하였다. 그 후, VVC 프로젝트가 공식적으로 시작되었을 때, JVET는 JVET로 이름이 변경되었다. VVC는 HEVC에 비해 비트율을 50% 감소시키는 것을 목표로 하는 새로운 코딩 표준이다.

[0006] VVC 표준 및 관련 코딩된 동영상 비트스트림을 위한 다양한 추가 향상 정보(Versatile Supplemental Enhancement Information for coded video bitstreams, VSEI) 표준은, 텔레비전 방송, 화상 회의 또는 저장 매체에서의 재생과 같은 전통적인 용도, 적응형 비트 전송률 스트리밍, 동영상 영역 추출, 여러 코딩된 동영상 비트스트림의 콘텐츠 구성 및 병합, 다중 뷰 동영상, 확장 가능한 레이어 코딩 및 뷰포트 적응형 360° 몰입형 미디어와 같은 더욱 새롭고 고급 사용 사례를 포함하는 최대한 광범위한 응용 분야에서 사용하도록 설계되었다. VVC 표준 개정안의 최신 초안에는 범위 확장 프로파일 사양 및 기타 측면이 포함되어 있다.

발명의 내용

해결하려는 과제

[0007] 본 개시 내용의 실시에는, 동영상을 처리하는 해결책을 제공한다.

[0008] 제1 측면에서, 동영상 처리 방법이 제안된다. 상기 방법은, 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함한다. 상기 GCI는, 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 지시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.

[0009] 본 개시의 제1 측면에 따른 방법에 기초하여, 6개의 GCI 구문 요소 각각의 값은 비트스트림에 존재하지 않는 경우, 미리 결정된 값(예를 들어, 0)과 동일한 것으로 추론된다. 이러한 GCI 구문 요소의 값이 추론되지 않는 종래의 해결책에 비해, 제안된 방법은, 이러한 GCI 구문 요소가 비트스트림에 없을 때 그 값을 얻을 수 있으므로, 동영상 처리를 위한 균일한 프로세스를 보장하고, 코딩 효율성을 향상시킬 수 있다는 이점이 있다.

[0010] 제2 측면에서, 동영상 데이터를 처리하는 장치가 제안된다. 동영상 데이터를 처리하는 장치는, 프로세서 및 명령이 포함된 비일시적 메모리를 포함한다. 명령은 프로세서에 의해 실행될 때, 프로세서로 하여금 본 개시의 제1 측면에 따른 방법을 수행하도록 한다.

[0011] 제3 측면에서는, 비일시적 컴퓨터 판독 가능 저장 매체가 제안된다. 비일시적 컴퓨터 판독 가능 저장 매체에는, 프로세서로 하여금 본 개시의 제1 측면에 따른 방법을 수행하도록 하는 명령이 저장된다.

[0012] 제4 측면에서는, 다른 비일시적 컴퓨터 판독 가능 기록 매체가 제안된다. 비일시적 컴퓨터 판독 가능 기록 매체

에는 동영상 처리 장치에 의해 수행되는 방법을 통해 생성된 동영상의 비트스트림이 저장된다. 상기 방법은, 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함한다. 상기 GCI는, 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 residual_ts_coding에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 slice_header에 지시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 abs_remaining 및 dec_abs_level의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 abs_remaining 및 dec_abs_level의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 sh_reverse_last_sig_coeff_flag가 상기 SPS를 참조하는 구문 구조 slice_header에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 여기서, 하나 이상의 구문 요소 각각은 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.

[0013] 제5 측면에서는, 동영상의 비트스트림을 저장하는 방법이 제안된다. 방법은, 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계로서, 상기 GCI는 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 residual_ts_coding에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 slice_header에 표시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 abs_remaining 및 dec_abs_level의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여 각 TU의 시작에서 구문 요소 abs_remaining 및 dec_abs_level의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 sh_reverse_last_sig_coeff_flag가 상기 SPS를 참조하는 구문 구조 slice_header에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함하는 단계 및 상기 비트스트림을 비일시적 컴퓨터 판독 기록 매체에 저장하는 단계를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.

[0014] 본 요약은, 아래의 상세한 설명에서 자세히 설명되는 개념의 선택을 간소화된 형태로 설명하기 위해 제공된다. 본 요약은, 청구된 주제의 주요 특징 또는 필수적인 특징을 식별하기 위한 것이 아니며, 또한, 청구된 주제의 범위를 제한하기 위해 사용하려는 의도가 없다.

도면의 간단한 설명

[0015] 첨부된 도면을 참조한 이하의 상세한 설명을 통해, 본 개시의 예시적인 실시예의 상기 목적과 기타 목적, 특징 및 이점이 더욱 명백해진다. 본 개시의 예시적인 실시예에서, 동일한 참조번호는 일반적으로 동일한 구성요소를 지칭한다.

도 1은 본 개시 내용의 일부 실시예에 따른 예시적인 동영상 코딩 시스템을 나타내는 블록도이다.

도 2는 본 개시 내용의 일부 실시예에 따른 제1 예시적인 동영상 인코더를 나타내는 블록도이다.

도 3은 본 개시의 일부 실시예에 따른 예시적인 동영상 디코더를 나타내는 블록도이다.

도 4는 본 개시 내용의 일부 실시예에 따른 동영상 처리 방법을 나타내는 흐름도이다.

도 5는 본 개시 내용의 다양한 실시예를 구현할 수 있는 컴퓨팅 디바이스를 나타내는 블록도이다.

도면 전체에서, 동일하거나 유사한 참조 부호는 일반적으로 동일하거나 유사한 요소를 지칭한다.

발명을 실시하기 위한 구체적인 내용

- [0016] 이하, 일부 실시예를 참조하여 본 개시의 원리에 대해 설명한다. 이러한 실시예는 설명의 목적으로만 기재되며, 본 개시의 범위에 대한 어떠한 제한도 제시하지 않고, 당업자가 본 개시를 이해하고 구현하는 데 도움을 주는 것으로 이해되어야 한다. 본 명세서에 설명되는 개시는 아래에서 설명되는 방법 외에도 다양한 방법으로 구현될 수 있다.
- [0017] 달리 정의되지 않는 한, 아래의 설명 및 청구범위에서 본 명세서에 사용된 모든 기술 용어 및 과학 용어는 본 개시가 속하는 기술분야에서 당업자에 의해 일반적으로 이해되는 것과 동일한 의미를 갖는다.
- [0018] 본 개시에 있어서, "일 실시예", "실시예", "예시적인 실시예" 등의 언급은 설명된 실시예가 특정 기능, 구조 또는 특성을 포함할 수 있다는 것을 나타내지만, 모든 실시예가 반드시 특정 기능, 구조 또는 특성을 포함해야 하는 것은 아니다. 또한, 이러한 문구는 반드시 동일한 실시예를 언급하는 것은 아니다. 또한, 특정 기능, 구조 또는 특성이 예시적인 실시예와 관련하여 설명되는 경우, 명시적으로 설명되었는지 여부에 관계없이 다른 실시예와 관련하여 이러한 기능, 구조 또는 특성에 영향을 미치는 것은 당업자의 지식 범위 내에 있는 것으로 제출된다.
- [0019] "제1" 및 "제2" 등의 용어는 본 명세서에서 다양한 요소를 설명하기 위해 사용될 수 있지만, 이러한 요소는 이러한 용어에 의해 제한되어서는 안 된다는 것을 이해해야 한다. 이러한 용어는 한 요소를 다른 요소와 구별하기 위해서만 사용된다. 예를 들어, 예시적인 실시예의 범위를 벗어나지 않으면서 제1 요소는 제2 요소로 명명될 수 있고, 유사하게 제2 요소가 제1 요소로 명명될 수 있다. 본 명세서에 사용되는 바와 같이, 용어 "및/또는"은 나열된 용어 중 하나 이상의 모든 조합을 포함한다.
- [0020] 본 명세서에 사용되는 용어는 특정한 실시예만을 설명하기 위해 사용되는 것으로, 예시적인 실시예를 한정하려는 의도가 없다. 본 명세서에서 사용되는 단수 형태는 문맥상 명백하게 달리 지시하지 않는 한 복수 형태도 포함하도록 의도된다. 본 명세서에 사용되는 용어 "구성하다", "구성되는", "갖는", "가지는", "포함하는" 및/또는 "포함되는"은 언급된 특징, 요소 및/또는 구성요소 등의 존재를 지정하지만, 하나 이상의 다른 특징, 요소, 구성요소 및/또는 이들의 조합이 존재할 수 있거나 추가되는 것을 배제하지 않는다.
- [0021] 예제 환경
- [0022] 도 1은, 본 개시의 기술을 이용할 수 있는 예시적인 동영상 코딩 시스템(100)을 나타내는 블록도이다. 도시된 바와 같이, 동영상 코딩 시스템(100)은 소스 디바이스(110) 및 대상 디바이스(120)를 포함할 수 있다. 소스 디바이스(110)는 동영상 인코딩 디바이스라고도 지칭될 수 있으며, 대상 디바이스(120)는 동영상 디코딩 디바이스라고도 지칭될 수 있다. 작동 시, 소스 디바이스(110)는 인코딩된 동영상 데이터를 생성하도록 구성될 수 있고, 대상 디바이스(120)는 소스 디바이스(110)에 의해 생성된 인코딩된 동영상 데이터를 디코딩하도록 구성될 수 있다. 소스 디바이스(110)는 동영상 소스(112), 동영상 인코더(114) 및 입력/출력(I/O) 인터페이스(116)를 포함할 수 있다.
- [0023] 동영상 소스(112)는 동영상 캡처 디바이스와 같은 소스를 포함할 수 있다. 동영상 캡처 디바이스의 예에는 동영상 콘텐츠 제공자로부터 동영상 데이터를 수신하기 위한 인터페이스, 동영상 데이터를 생성하기 위한 컴퓨터 그래픽 시스템 및/또는 이들의 조합이 포함되지만 이에 제한되지는 않는다.
- [0024] 동영상 데이터는 하나 이상의 픽처를 포함할 수 있다. 동영상 인코더(114)는 동영상 소스(112)로부터의 동영상 데이터를 인코딩하여 비트스트림을 생성한다. 비트스트림은 동영상 데이터의 코딩된 표현을 형성하는 비트 시퀀스를 포함할 수 있다. 비트스트림은 코딩된 픽처 및 관련 데이터를 포함할 수 있다. 코딩된 픽처는 픽처의 코딩된 표현이다. 관련 데이터에는 시퀀스 매개변수 세트, 픽처 매개변수 세트 및 기타 구문 구조가 포함될 수 있다. I/O 인터페이스(116)는 변조기/복조기 및/또는 송신기를 포함할 수 있다. 인코딩된 동영상 데이터는 네트워크(130A)를 통해 I/O 인터페이스(116)에 의해 대상 디바이스(120)에 직접 전송될 수 있다. 인코딩된 동영상 데이터는 또한 대상 디바이스(120)에 의한 액세스를 위해 저장 매체/서버(130B)에 저장될 수 있다.
- [0025] 대상 디바이스(120)는 I/O 인터페이스(126), 동영상 디코더(124) 및 디스플레이 디바이스(122)를 포함할 수 있다. I/O 인터페이스(126)는 수신기 및/또는 모뎀을 포함할 수 있다. I/O 인터페이스(126)는 소스 디바이스(110) 또는 저장 매체/서버(130B)로부터 인코딩된 동영상 데이터를 획득할 수 있다. 동영상 디코더(124)는 인코딩된 동영상 데이터를 디코딩할 수 있다. 디스플레이 디바이스(122)는 디코딩된 동영상 데이터를 사용자에게 디스플레이할 수 있다. 디스플레이 디바이스(122)는 대상 디바이스(120)에 통합되거나, 외부 디스플레이 디바이스와 인터페이스하도록 구성된 대상 디바이스(120)의 외부에 있을 수 있다.

- [0026] 동영상 인코더(114) 및 동영상 디코더(124)는 고효율 동영상 코딩(High Efficiency Video Coding, HEVC) 표준, 다용도 동영상 코딩(Versatile Video Coding, VVC) 표준 및 기타 현재 표준 및/또는 추가 표준과 같은 동영상 압축 표준에 따라 동작할 수 있다.
- [0027] 도 2는 본 개시의 일부 실시예에 따른, 도 1에 예시된 시스템(100)의 동영상 인코더(114)의 예일 수 있는 동영상 인코더(200)의 예를 나타내는 블록도이다.
- [0028] 동영상 인코더(200)는 본 개시의 기술 중 일부 또는 전부를 구현하도록 구성될 수 있다. 도 2의 예에서, 동영상 인코더(200)는 복수의 기능적 구성요소를 포함한다. 본 개시에서 설명되는 기술은 동영상 인코더(200)의 다양한 구성요소 사이에 공유될 수도 있다. 일부 예에서, 프로세서는 본 개시에 설명된 기술 중 일부 또는 전부를 수행하도록 구성될 수 있다.
- [0029] 일부 실시예에서, 동영상 인코더(200)는 분할 유닛(201), 모드 선택 유닛(203), 모션 추정 유닛(204), 모션 보상 유닛(205) 및 인트라 예측 유닛(206)을 포함할 수 있는 예측 유닛(202), 잔차 생성 유닛(207), 변환 유닛(208), 양자화 유닛(209), 역양자화 유닛(210), 역변환 유닛(211), 재구성 유닛(212), 버퍼(213) 및 엔트로피 인코딩 유닛(214)을 포함할 수 있다.
- [0030] 다른 예에서, 동영상 인코더(200)는 더 많거나, 더 적거나, 다른 기능적 구성요소를 포함할 수 있다. 일 예에서, 예측 유닛(202)은 블록 내 복사(IBC) 유닛을 포함할 수 있다. IBC 유닛은 적어도 하나의 참조 픽처가 현재 동영상 블록이 위치하는 픽처인 IBC 모드에서 예측을 수행할 수 있다.
- [0031] 또한, 모션 추정 유닛(204) 및 모션 보상 유닛(205)과 같은 일부 구성요소는 통합될 수 있지만, 도 2의 예에서는 설명을 위해 별도로 표시된다.
- [0032] 분할 유닛(201)은 픽처를 하나 이상의 동영상 블록으로 분할할 수 있다. 동영상 인코더(200) 및 동영상 디코더(300)는 다양한 동영상 블록 크기를 지원할 수 있다.
- [0033] 모드 선택 유닛(203)은, 예를 들어 예러 결과에 기초하여 인트라 또는 인터 코딩 모드 중 하나를 선택하고, 그 결과로부터 얻어진 인트라 코딩 또는 인터 코딩된 블록을, 잔차 블록 데이터를 생성하기 위한 잔차 생성 유닛(207)에 제공하고, 인코딩된 블록을 재구성하여 참조 픽처로서 사용하기 위해 재구성 유닛(212)에 제공할 수 있다. 일부 예에서, 모드 선택 유닛(203)은 예측이 인트라 예측 신호 및 인트라 예측 신호에 기초하는 인트라 및 인터 예측(Intra and Inter Predication, CIIP) 모드의 조합을 선택할 수 있다. 모드 선택 유닛(203)은 인터 예측의 경우 블록에 대한 모션 벡터의 해상도(예를 들어, 서브 픽셀 또는 정수 픽셀 정밀도)를 선택할 수도 있다.
- [0034] 현재 동영상 블록에 대해 인터 예측을 수행하기 위해, 모션 추정 유닛(204)은 버퍼(213)로부터의 하나 이상의 참조 프레임을 현재 동영상 블록과 비교함으로써 현재 동영상 블록에 대한 모션 정보를 생성할 수 있다. 모션 보상 유닛(205)은 버퍼(213)로부터 현재 동영상 블록과 관련된 픽처를 제외한 픽처의 모션 정보 및 디코딩된 샘플에 기초하여 현재 동영상 블록에 대한 예측 동영상 블록을 결정할 수 있다.
- [0035] 모션 추정 유닛(204) 및 모션 보상 유닛(205)은 예를 들어, 현재 동영상 블록이 I-슬라이스, P-슬라이스 또는 B-슬라이스에 있는지 여부에 따라 현재 동영상 블록에 대해 서로 다른 연산을 수행할 수 있다. 본 명세서에서 사용된 바와 같이, "I-슬라이스"는 매크로블록으로 구성된 픽처의 일부를 의미할 수 있으며, 이는 모두 동일한 픽처의 매크로블록에 기초한다. 또한, 본 명세서에서 사용되는 바와 같이, 일부 측면에서 "P-슬라이스" 및 "B-슬라이스"는 동일한 픽처의 매크로블록에 의존하지 않는 매크로블록으로 구성된 픽처의 부분을 지칭할 수 있다.
- [0036] 일부 예에서, 모션 추정 유닛(204)은 현재 동영상 블록에 대한 단방향 예측을 수행할 수 있고, 모션 추정 유닛(204)은 현재 동영상 블록의 참조 동영상 블록에 대해 리스트 0 또는 리스트 1의 참조 픽처를 검색할 수 있다. 그 후, 모션 추정 유닛(204)은 참조 동영상 블록을 포함하는 리스트 0 또는 리스트 1의 참조 픽처를 지시하는 참조 인덱스와 현재 동영상 블록과 참조 동영상 블록 사이의 공간적 변위를 지시하는 모션 벡터를 생성할 수 있다. 모션 추정 유닛(204)은 참조 인덱스, 예측 방향 지시자 및 모션 벡터를 현재 동영상 블록의 모션 정보로 출력할 수 있다. 모션 보상 유닛(205)은 현재 동영상 블록의 모션 정보에 의해 지시되는 참조 동영상 블록을 기초로 현재 동영상 블록의 예측 동영상 블록을 생성할 수 있다.
- [0037] 대안적으로, 다른 예에서, 모션 추정 유닛(204)은 현재 동영상 블록에 대해 양방향 예측을 수행할 수 있다. 모션 추정 유닛(204)은 현재 동영상 블록의 참조 동영상 블록에 대해 리스트 0의 참조 픽처를 검색할 수 있고, 현재 동영상 블록의 다른 참조 동영상 블록에 대해 리스트 1의 참조 픽처를 검색할 수도 있다. 그 다음, 모션 추정 유닛(204)은 참조 동영상 블록과 현재 동영상 블록 사이의 공간적 변위를 나타내는 모션 벡터 및 참조 동영상

상 블록을 포함하는 리스트 0 및 리스트 1의 참조 픽처를 나타내는 참조 인덱스를 생성할 수 있다. 모션 추정 유닛(204)은 현재 동영상 블록의 참조 인덱스 및 모션 벡터를 현재 동영상 블록의 모션 정보로서 출력할 수 있다. 모션 보상 유닛(205)은 현재 동영상 블록의 모션 정보에 의해 표시된 참조 동영상 블록을 기반으로 현재 동영상 블록의 예측 동영상 블록을 생성할 수 있다.

[0038] 일부 예에서, 모션 추정 유닛(204)은 디코더의 디코딩 처리를 위한 모션 정보의 전체 세트를 출력할 수 있다. 대안적으로, 일부 실시예에서, 모션 추정 유닛(204)은 다른 동영상 블록의 모션 정보를 참조하여 현재 동영상 블록의 모션 정보를 시그널링할 수 있다. 예를 들어, 모션 추정 유닛(204)은 현재 동영상 블록의 모션 정보가 인접한 동영상 블록의 모션 정보와 충분히 유사하다고 결정할 수 있다.

[0039] 일 예에서, 모션 추정 유닛(204)은 현재 동영상 블록과 연관된 구문 구조에서, 현재 동영상 블록이 다른 동영상 블록과 동일한 모션 정보를 갖는다는 것을 동영상 디코더(300)에 지시하는 값을 표시할 수 있다.

[0040] 다른 예에서, 모션 추정 유닛(204)은 현재 동영상 블록과 연관된 구문 구조에서, 다른 동영상 블록 및 모션 벡터 차이(MVD)를 식별할 수 있다. 모션 벡터 차이는 현재 동영상 블록의 모션 벡터와 지시된 동영상 블록의 모션 벡터 간의 차이를 지시한다. 동영상 디코더(300)는 지시된 동영상 블록의 모션 벡터와 모션 벡터 차이를 사용하여 현재 동영상 블록의 모션 벡터를 결정할 수 있다.

[0041] 위에서 논의된 바와 같이, 동영상 인코더(200)는 모션 벡터를 예측적으로 시그널링할 수도 있다. 동영상 인코더(200)에 의해 구현될 수 있는 예측 시그널링 기술의 두 가지 예는 고급 모션 벡터 예측(Advanced Motion Vector Predication, AMVP) 및 병합 모드 시그널링이 포함된다.

[0042] 인트라 예측 유닛(206)은 현재 동영상 블록에 대해 인트라 예측을 수행할 수 있다. 인트라 예측 유닛(206)이 현재 동영상 블록에 대해 인트라 예측을 수행하는 경우, 인트라 예측 유닛(206)은 동일 픽처 내 다른 동영상 블록의 디코딩된 샘플을 기반으로, 현재 동영상 블록에 대한 예측 데이터를 생성할 수 있다. 현재 동영상 블록에 대한 예측 데이터에는 예측된 동영상 블록 및 다양한 구문 요소가 포함될 수 있다.

[0043] 잔차 생성 유닛(207)은 현재 동영상 블록에서 현재 동영상 블록의 예측 동영상 블록을 빼는 것(예를 들어, 마이너스 부호로 지시됨)을 통해 현재 동영상 블록에 대한 잔차 데이터를 생성할 수 있다. 현재 동영상 블록의 잔차 데이터는 현재 동영상 블록의 샘플의 서로 다른 샘플 구성요소에 대응하는 잔차 동영상 블록을 포함할 수 있다.

[0044] 다른 예에서, 예를 들어 스킵 모드에서 현재 동영상 블록에 대한 현재 동영상 블록의 잔차 데이터가 없을 수도 있고, 잔차 생성 유닛(207)은 감산 연산을 수행하지 않을 수도 있다.

[0045] 변환 처리 유닛(208)은 현재 동영상 블록과 연관된 잔차 동영상 블록에 하나 이상의 변환을 적용함으로써 현재 동영상 블록에 대한 하나 이상의 변환 계수 동영상 블록을 생성할 수 있다.

[0046] 변환 처리 유닛(208)이 현재 동영상 블록과 연관된 변환 계수 동영상 블록을 생성한 후, 양자화 유닛(209)은 현재 동영상 블록과 연관된 하나 이상의 양자화 매개변수(QP) 값을 기반으로 현재 동영상 블록과 연관된 변환 계수 동영상 블록을 양자화할 수 있다.

[0047] 역양자화부(210) 및 역변환부(211)는 변환계수 동영상 블록에 각각 역양자화 및 역변환을 적용하여 변환 계수 동영상 블록으로부터 잔차 동영상 블록을 재구성할 수 있다. 재구성 유닛(212)은 버퍼(213)에 저장할 현재 동영상 블록과 연관된 재구성된 동영상 블록을 생성하기 위해, 재구성된 잔차 동영상 블록을 예측 유닛(202)에 의해 생성된 하나 이상의 예측 동영상 블록으로부터의 대응 샘플에 추가할 수 있다.

[0048] 재구성 유닛(212)이 동영상 블록을 재구성한 후, 루프 필터링 동작이 수행되어 동영상 블록 내의 동영상 블로킹 아티팩트를 감소시킬 수 있다.

[0049] 엔트로피 인코딩 유닛(214)은 동영상 인코더(200)의 다른 기능적 구성요소로부터 데이터를 수신할 수 있다. 엔트로피 인코딩 유닛(214)이 데이터를 수신하면, 엔트로피 인코딩 유닛(214)은 하나 이상의 엔트로피 인코딩 동작을 수행하여 엔트로피 인코딩 데이터를 생성하고, 엔트로피 인코딩 데이터를 포함하는 비트스트림을 출력할 수 있다.

[0050] 도 3은 본 개시의 일부 실시예에 따른, 도 1에 예시된 시스템(100)의 동영상 디코더(124)의 예일 수 있는 동영상 디코더(300)의 예를 나타내는 블록도이다.

[0051] 동영상 디코더(300)는 본 개시의 기술 중 일부 또는 전부를 수행하도록 구성될 수 있다. 도 3의 예에서, 동영상 디코더(300)는 복수의 기능 구성요소를 포함한다. 본 개시에 설명된 기술은 동영상 디코더(300)의 다양한 구성

요소 사이에 공유될 수도 있다. 일부 예에서, 프로세서는 본 개시에 설명된 기술 중 일부 또는 전부를 수행하도록 구성될 수 있다.

[0052] 도 3의 예에서, 동영상 디코더(300)는 엔트로피 디코딩 유닛(301), 모션 보상 유닛(302), 인트라 예측 유닛(303), 역양자화 유닛(304), 역변환 유닛(305), 재구성 유닛(306) 및 버퍼(307)를 포함한다. 동영상 디코더(300)는 일부 예에서 동영상 인코더(200)에 관해 설명된 인코딩 패스와 일반적으로 상호 대응하는 디코딩 패스를 수행할 수도 있다.

[0053] 엔트로피 디코딩 유닛(301)은 인코딩된 비트스트림을 검색할 수 있다. 인코딩된 비트스트림은 엔트로피 코딩된 동영상 데이터(예를 들어, 동영상 데이터의 인코딩된 블록)를 포함할 수 있다. 엔트로피 디코딩 유닛(301)은 엔트로피 코딩된 동영상 데이터를 디코딩할 수 있고, 엔트로피 디코딩된 동영상 데이터로부터 모션 보상 유닛(302)은 모션 벡터, 모션 벡터 정밀도, 참조 픽처 리스트 인덱스 및 기타 모션 정보를 포함하는 모션 정보를 결정할 수 있다. 예를 들어, 모션 보상 유닛(302)은 AMVP 및 병합 모드를 수행함으로써 이러한 정보를 결정할 수 있다. 인접한 PB 및 참조 화면의 데이터를 기반으로 가장 가능성이 높은 몇 가지 후보를 도출하는 것을 포함하여 AMVP가 사용된다. 모션 정보에는 일반적으로 수평 및 수직 모션 벡터 변위 값, 하나 또는 두 개의 참조 픽처 인덱스 및 B 슬라이스의 예측 영역의 경우 참조 픽처 리스트가 각 인덱스와 연관되어 있다는 식별이 포함된다. 본 명세서에서 사용된 바와 같이, 일부 측면에서, "병합 모드"는 공간적으로 또는 시간적으로 인접한 블록으로부터 모션 정보를 도출하는 것을 의미할 수 있다.

[0054] 모션 보상 유닛(302)은 모션 보상된 블록을 생성할 수 있고, 아마도 보간 필터에 기초하여 보간을 수행할 수 있다. 하위 픽셀 정밀도로 사용되는 보간 필터의 식별자가 구문 요소에 포함될 수 있다.

[0055] 모션 보상 유닛(302)은 동영상 블록의 인코딩 중에 동영상 인코더(200)에 의해 사용되는 보간 필터를 사용하여 참조 블록의 서브 정수 픽셀에 대한 보간된 값을 계산할 수 있다. 모션 보상 유닛(302)은 수신된 구문 정보에 따라 동영상 인코더(200)에 의해 사용되는 보간 필터를 결정하고, 보간 필터를 사용하여 예측 블록을 생성할 수 있다.

[0056] 모션 보상 유닛(302)은 인코딩된 동영상 시퀀스의 프레임 및/또는 슬라이스를 인코딩하는 데 사용되는 블록의 크기, 인코딩된 동영상 시퀀스 픽처의 각 매크로블록이 어떻게 분할되는지 설명하는 분할 정보, 각 분할이 인코딩되는 방식을 지시하는 모드, 각 인터 인코딩된 블록에 대한 하나 이상의 참조 프레임(및 참조 프레임 리스트) 및 인코딩된 동영상 시퀀스를 디코딩하기 위한 기타 정보를 결정하기 위해 구문 정보의 적어도 일부를 사용할 수 있다. 본 명세서에서 사용된 바와 같이, 일부 측면에서, "슬라이스"는 엔트로피 코딩, 신호 예측 및 잔차 신호 재구성의 관점에서, 동일한 픽처의 다른 슬라이스와 독립적으로 디코딩될 수 있는 데이터 구조를 지칭할 수 있다. 슬라이스는 전체 픽처일 수 있고 픽처의 영역일 수도 있다.

[0057] 인트라 예측 유닛(303)은 예를 들어 비트스트림에서 수신된 인트라 예측 모드를 사용하여 공간적으로 인접한 블록으로부터 예측 블록을 형성할 수 있다. 역양자화 유닛(304)은 비트스트림에 제공되고, 엔트로피 디코딩 유닛(301)에 의해 디코딩된 양자화 동영상 블록 계수를 역 양자화, 즉 비양자화 한다. 역변환 유닛(305)은 역변환을 적용한다.

[0058] 재구성 유닛(306)은 예를 들어, 잔차 블록을 모션 보상 유닛(302) 또는 인트라 예측 유닛(303)에 의해 생성된 대응하는 예측 블록과 합산함으로써 디코딩된 블록을 얻을 수 있다. 원하는 경우, 블록화 아티팩트를 제거하기 위해 디블록킹 필터를 적용하여 디코딩된 블록을 필터링할 수도 있다. 그런 다음 디코딩된 동영상 블록은 버퍼(307)에 저장되며, 이는 후속 모션 보상/인트라 예측을 위한 참조 블록을 제공하고 또한 디스플레이 장치에 표시하기 위해 디코딩된 동영상을 생성한다.

[0059] 이하, 본 개시의 일부 예시적인 실시예를 상세히 설명한다. 본 명세서에서는 이해를 쉽게 하기 위해 섹션 표제가 사용되었으며, 섹션에 개시된 실시예가 해당 섹션에만 제한되지 않는다는 점을 이해해야 한다. 또한, 특정 실시예는 다용도 동영상 코딩 또는 다른 특정 동영상 코덱을 참조하여 설명되지만, 개시된 기술은 다른 동영상 코딩 기술에도 적용될 수 있다. 또한, 일부 실시예에서는 동영상 코딩 단계를 자세히 설명하지만, 코딩을 취소하는 대응하는 해당 디코딩 단계는 디코더에 의해 구현되는 것으로 이해될 것이다. 또한 동영상 처리라는 용어는 동영상 코딩 또는 압축, 동영상 디코딩 또는 압축 해제, 동영상 픽셀을 하나의 압축 형식에서 다른 압축 형식으로 또는 다른 압축 비트율로 표현하는 동영상 트랜스코딩을 포함한다.

[0060] 1. 요약

[0061] 본 개시는 이미지/동영상 코딩 기술에 관한 것이다. 구체적으로, 이는 다용도 동영상 코딩(VVC) 범위 확장, 일

반 제약 정보(GCI) 플래그의 시그널링과 관련된다. 이 아이디어는 VVC 표준과 같은 임의의 동영상 코덱으로 코딩된 동영상 비트스트림에 대해 개별적으로 또는 다양한 조합으로 적용될 수 있다.

2. 배경

2.1. 동영상 코딩 표준

동영상 코딩 표준은 주로 잘 알려진 ITU-T 및 ISO/IEC 표준의 개발을 통해 발전해 왔다. ITU-T는 H.261과 H.263을 제작하였고, ISO/IEC는 MPEG-1과 MPEG-4 Visual을 제작하였으며, 두 조직은 공동으로 H.262/MPEG-2 Video, H.264/MPEG-4 고도 동영상 부호화(AVC) 및 H.265/HEVC 표준을 제작하였다. H.262 이후 동영상 코딩 표준은 시간적 예측과 변환 코딩이 활용되는 하이브리드 동영상 코딩 구조를 기반으로 한다. HEVC를 초과하는 미래의 동영상 코딩 기술을 탐구하기 위해 2015년 VCEG와 MPEG이 공동으로 합동 동영상 탐험대(Joint Video Exploration Team, JVET)를 설립하였다. 그 이후로, JVET는 많은 새로운 방법을 채택하여 공통 탐사 모델(Joint Exploration Model, JEM)이라는 참조 소프트웨어에 추가하였다. 그 후, 다용도 동영상 코딩(VVC) 프로젝트가 공식적으로 시작되었을 때, 합동 동영상 탐험대(JVET)는 JVET로 이름이 변경되었다. VVC는 HEVC에 비해 비트율을 50% 감소시키는 것을 목표로 하는 새로운 코딩 표준으로, 2020년 7월 1일에 끝난 제19차 회의에서 JVET에 의해 최종 확정되었다.

다용도 동영상 코딩(VVC) 표준(ITU-T H.266 | ISO/IEC 23090-3) 및 관련 코딩된 동영상 비트스트림(VSEI)을 위한 다용도 추가 향상 정보 표준(ITU-T H.274 | ISO/IEC 23002-7)은, 텔레비전 방송, 화상 회의 또는 저장 매체에서의 재생과 같은 전통적인 용도, 적응형 비트 전송률 스트리밍, 동영상 영역 추출, 여러 코딩된 동영상 비트스트림의 콘텐츠 구성 및 병합, 다중 뷰 동영상, 확장 가능한 레이어 코딩 및 뷰포트 적응형 360° 몰입형 미디어와 같은 더욱 새롭고 고급 사용 사례를 포함하는 최대한 광범위한 응용 분야에서 사용하도록 설계되었다.

필수 동영상 코딩(Essential Video Coding, EVC) 표준(ISO/IEC 23094-1)은 MPEG에서 최근에 개발한 또 다른 동영상 코딩 표준이다.

VVC 표준 개정안의 최신 초안에는 범위 확장 사양과 기타 몇 가지 측면이 포함되어 있다.

2.2. VVC의 프로파일, 계층, 레벨(PTL) 및 서브 프로파일

VVC v1에서는 표 1에 설명된 바같이, 6개의 프로파일 정의된다. 이러한 프로파일은 각각 광범위한 애플리케이션을 다루기 위해 정의되었으며, 서로 논리적인 "중첩" 관계를 가지고 있다.

표 1에서 "4:2:0"이라는 용어는 소비자 애플리케이션에서 카메라로 캡처된 콘텐츠를 인코딩하는 데 사용되는 가장 일반적인 형식인 루마 평면의 너비와 높이가 절반인 채도 색상 평면이 있는 동영상을 의미한다. "4:4:4"는 그래픽, 디스플레이 모니터 및 컴퓨터 데스크탑 렌더링에 일반적으로 사용되는 루마 평면과 너비 및 높이가 동일한 크로마 색상 평면을 사용하는 것을 의미한다. "4:2:2"는, 스튜디오에서 인터레이스 캡처 동영상에 자주 사용되는 것과 같은 루마 너비의 절반이지만 높이가 같은 크로마를 사용하는 덜 일반적인 형식이다. 흑백 동영상에는 루마 평면이라고 하는 단일 색상 평면만 있다(실제로 루마 신호를 나타내지는 않을 수 있지만, 예를 들어 흑백 픽처는 3D 동영상 애플리케이션의 심도 맵이나 배경에 디코딩된 동영상을 오버레이하는 투명도 맵을 나타낼 수 있다).

비트스트림의 제1 픽처는 일반적으로 인트라 코딩된 픽처이기 때문에, 스틸 픽처 프로파일을 따르는 디코더는, 일반적인 동영상 프로파일 비트스트림의 제1 픽처를 디코딩할 수 있어야 한다는 점에 유의해야 한다. 이러한 디코더는 동영상 비트스트림에서 스냅샷으로 추출된 다른 IRAP 픽처도 디코딩할 수 있다. 동영상 프로파일과 스틸 픽처 프로파일 사이에 이러한 서브 세트 관계가 있으면 다양한 애플리케이션에서 사용할 수 있도록 인코더와 디코더 모듈을 공유할 수 있는 기능이 제공된다(그리고 대부분의 최신 동영상 카메라를 스틸 이미지 사진 촬영에도 사용할 수 있고 그 반대의 경우도 가능하다는 점으로 수렴되는 추세이다).

【표 1】 VVC v1 에서 정의된 6 개의 프로필

프로필 이름	비고
메인 10	<ul style="list-style-type: none"> • 흑백 및 4:2:0 크로마 샘플링 전용 • 8 내지 10 비트 비트 심도 • 비트스트림에는 단 하나의 레이어만 있다
메인 10 스틸 픽처	메인 10 프로필과 동일하지만 비트스트림에는 단일 픽처만 포함된다(픽처 간 예측 없음)
메인 10 4:4:4	메인 10 프로필과 동일하지만 4:2:2 및 4:4:4 크로마 샘플링과 팔레트 및 적응형 색상 변환 코딩 도구를 추가로 지원한다
메인 10 4:4:4 스틸 픽처	메인 10 4:4:4 프로필과 동일하지만 비트스트림에는 단일 픽처만 포함된다(픽처 간 예측 없음)
다층 메인 10	메인 10 프로필과 동일하지만 비트스트림은 하나 이상의 레이어를 가질 수 있다
다층 메인 10 4:4:4	메인 10 4:4:4 프로필과 동일하지만 비트스트림은 하나 이상의 레이어를 가질 수 있다

[0072]

[0073]

PTL 정보는 VPS(이 경우 PTL 구조는 하나 이상의 OLS에 적용됨) 또는 SPS(이 경우 PTL 구조는 해당 SPS를 참조하는 레이어만 포함하는 OLS에 적용됨)에 포함될 수 있는 구문 구조를 사용하여 시그널링된다. 여기에는 비트스트림이 따르는 PTL과 시간적 서브 레이어 표현을 위한 추가 PTL이 포함되며, 각각은 비트스트림의 독립된 서브 세트이다.

[0074]

표 1에 표시된 바와 같이, VVC 사양에 정의된 프로필 외에도 VVC를 사용하면 인코더에서 서브 프로필에 대한 비트스트림 적합성을 시그널링할 수 있다. 서브 프로필은 기존에 지시된 프로필에 추가 제한을 부과하는 프로필과 유사한 상호 운용성 서브 세트 표시기이다. 이러한 서브 프로필은 VVC 사양 외부에서 정의되며 Rec에 지정된 대로 등록된 식별자 코드를 사용하여 지시된다. 동일한 서브 프로필 코드 값에 대해 여러 가지 의미가 정의되는 것을 방지하기 위해 ITU-T T.35에 등록된 식별 코드를 사용하여 표시된다. 외부 조직은 특정 애플리케이션의 요구 사항을 충족하는 데 충분하다고 생각되는 자체 서브 프로필을 정의할 수 있다. PTL 구조 내의 서브 프로필 표시기 구문 요소를 사용하면 비트스트림이 외부에서 정의된 제한 사항을 준수한다는 것을 비트스트림 내에서 시그널링할 수 있다.

[0075]

2.3.VVC 일반 제약 정보 (GCI)

[0076]

VVC에서는 프로필, 계층 및 레벨(PTL) 정보 외에도, PTL 구문 구조에는 비트스트림의 특정 제약 속성을 지시하는 제약 플래그 및 비플래그 구문 요소 리스트를 포함하는 일반 제약 정보(GCI) 구문 구조가 선택적으로 포함될 수 있다. 존재하는 경우, GCI 구문 요소 값이 0보다 크면, 비트스트림이 특정 방식으로 제약되어 있음을 지시하며, 일반적으로 특정 코딩 도구가 비트스트림에서 사용되지 않음을 지시하는 반면, 값 0은 관련 제약이 적용되지 않을 수 있음을 지시하며, 관련 코딩 도구가 비트스트림(지시된 프로필에서 사용이 지원되는 경우)에서 사용

이 허용(필수는 아님)될 수 있음을 알린다.

- [0077] GCI 구문 구조에는 다음을 포함한 여러 유형의 제약 조건 구문 요소가 포함되어 있다.
- [0078] - 모든 레이어가 독립적으로 코딩되거나 비트스트림에 하나의 AU만 포함되어 있음을 지시하는 인트라 코딩만 사용되는 것과 같은 일반적인 비트스트림 제한에 대한 플래그;
- [0079] - 코딩된 픽처의 비트 심도와 채도 형식을 제한하는 필드;
- [0080] - 특정 NAL 유닛 유형이 비트스트림 내에 존재하는 것이 허용되지 않음을 지시하는 플래그;
- [0081] - 비트스트림 내에서 픽처를 슬라이스, 계층 및 하위 픽처로 분할할 수 있는 방식을 제한하는 플래그;
- [0082] - CTU의 크기와 분할 트리의 크기 및 유형을 제한하는 플래그;
- [0083] - 특정 인트라 코딩 도구의 사용을 제한하는 플래그;
- [0084] - 특정 인터 코딩 도구의 사용을 제한하는 플래그;
- [0085] - 변환, 양자화 및 잔여 코딩 도구를 제한하는 플래그 및
- [0086] - 루프 내 필터의 측면을 제한하는 플래그.
- [0087] GCI 구문 구조의 목적은, 비트스트림 디코딩하는 데 필요한 기능에 대한 구성 정보를 간단하게 검색하고, 이전 동영상 코딩 표준에서 허용한 것보다 더 세밀하게 PTL에서 지정한 것 이상으로 제한을 부과하는 상호 운용성 포인트의 시그널링을 허용하기 위한 것이다. 서브 프로파일과 유사하게, GCI 구문 구조를 사용하면, VVC 프로파일의 모든 기능을 지원하지는 않지만 특정 애플리케이션의 요구 사항을 해결하는 디코더 구현에 대해 상호 운용성을 정의할 수 있다. 디코더 구현에서는, 디코딩 프로세스를 구성하는 방법을 결정하고, 비트스트림이 디코더에 의해 디코딩될 수 있는지 여부를 식별하기 위해 비트스트림이 특정 기능의 사용을 피하는지 확인하기 위해 GCI 구문 요소를 검사할 수 있다. VVC 프로파일의 모든 기능을 지원하는 디코더 구현은, 이러한 디코더가 지시된 PTL를 준수하는 모든 비트스트림을 디코딩할 수 있기 때문에, GCI 구문 요소 값을 무시할 수 있다.
- [0088] 의미론이 VVC 사양 외부에서 정의된 서브 프로파일 표시기와 달리, GCI 구문 요소의 의미론은 VVC 사양 내에서 정의된다. 서브 프로파일은, GCI 구문 요소의 값에 제약을 가하는 서브 프로파일과 함께 GCI와 결합하여 사용할 수도 있다. 서브 프로파일 표시기와 함께 또는 그 대신 GCI를 사용하면 서브 프로파일 표시기의 의미를 디코더에 의해 인식되지 못할 가능성을 피할 수 있다(서브 프로파일 표시기의 의미를 공개할 필요가 없기 때문이다).

[0089] GCI 구문 구조의 구문은 다음과 같다.

general_constraints_info() {	설명자
gci_present_flag	u(1)
if(gci_present_flag) {	
/* 일반적인 */	
gci_intra_only_constraint_flag	u(1)
gci_all_layers_independent_constraint_flag	u(1)
gci_one_au_only_constraint_flag	u(1)
/* 픽처 형식 */	
gci_sixteen_minus_max_bitdepth_constraint_idc	u(4)
gci_three_minus_max_chroma_format_constraint_idc	u(2)
/* NAL 유닛 유형 관련 */	
gci_no_mixed_nalu_types_in_pic_constraint_flag	u(1)
gci_no_trail_constraint_flag	u(1)
gci_no_stsa_constraint_flag	u(1)
gci_no_rasl_constraint_flag	u(1)
gci_no_rapl_constraint_flag	u(1)
gci_no_idr_constraint_flag	u(1)
gci_no_cra_constraint_flag	u(1)
gci_no_gdr_constraint_flag	u(1)
gci_no_aps_constraint_flag	u(1)
gci_no_idr_rpl_constraint_flag	u(1)
/* 계층, 슬라이스, 서브픽처 분할 */	
gci_one_tile_per_pic_constraint_flag	u(1)
gci_pic_header_in_slice_header_constraint_flag	u(1)
gci_one_slice_per_pic_constraint_flag	u(1)
gci_no_rectangular_slice_constraint_flag	u(1)
gci_one_slice_per_subpic_constraint_flag	u(1)
gci_no_subpic_info_constraint_flag	u(1)

[0090]

/* CTU 및 블록 분할 */	
gci_three_minus_max_log2_ctu_size_constraint_idc	u(2)
gci_no_partition_constraints_override_constraint_flag	u(1)
gci_no_mtt_constraint_flag	u(1)
gci_no_qtbt_dual_tree_intra_constraint_flag	u(1)
/* 인트라 */	
gci_no_palette_constraint_flag	u(1)
gci_no_ibc_constraint_flag	u(1)
gci_no_isp_constraint_flag	u(1)
gci_no_mrl_constraint_flag	u(1)
gci_no_mip_constraint_flag	u(1)
gci_no_cclm_constraint_flag	u(1)
/* 인터 */	
gci_no_ref_pic_resampling_constraint_flag	u(1)
gci_no_res_change_in_clvs_constraint_flag	u(1)
gci_no_weighted_prediction_constraint_flag	u(1)
gci_no_ref_wraparound_constraint_flag	u(1)
gci_no_temporal_mvp_constraint_flag	u(1)
gci_no_sbtmvp_constraint_flag	u(1)
gci_no_amvr_constraint_flag	u(1)
gci_no_bdof_constraint_flag	u(1)
gci_no_smvd_constraint_flag	u(1)
gci_no_dmvr_constraint_flag	u(1)
gci_no_mmvd_constraint_flag	u(1)
gci_no_affine_motion_constraint_flag	u(1)
gci_no_prof_constraint_flag	u(1)
gci_no_bcw_constraint_flag	u(1)
gci_no_ciip_constraint_flag	u(1)

[0091]

gci_no_gpm_constraint_flag	u(1)
/* 변환, 정량화, 잔류 */	
gci_no_luma_transform_size_64_constraint_flag	u(1)
gci_no_transform_skip_constraint_flag	u(1)
gci_no_bdpcm_constraint_flag	u(1)
gci_no_mts_constraint_flag	u(1)
gci_no_lfnst_constraint_flag	u(1)
gci_no_joint_cbr_constraint_flag	u(1)
gci_no_sbt_constraint_flag	u(1)
gci_no_act_constraint_flag	u(1)
gci_no_explicit_scaling_list_constraint_flag	u(1)
gci_no_dep_quant_constraint_flag	u(1)
gci_no_sign_data_hiding_constraint_flag	u(1)
gci_no_cu_qp_delta_constraint_flag	u(1)
gci_no_chroma_qp_offset_constraint_flag	u(1)
/* 루프 필터 */	
gci_no_sao_constraint_flag	u(1)
gci_no_alf_constraint_flag	u(1)
gci_no_ccalf_constraint_flag	u(1)
gci_no_lmcs_constraint_flag	u(1)
gci_no_ladf_constraint_flag	u(1)
gci_no_virtual_boundaries_constraint_flag	u(1)
gci_num_reserved_bits	u(8)
for(i = 0; i < gci_num_reserved_bits; i++)	
gci_reserved_zero_bit[i]	u(1)
}	
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
}	

[0092]

[0093]

GCI 구문 구조의 구문에서, 제1 구문 요소는 모든 GCI 구문 요소의 존재와 향후의 GCI 구문 요소를 위해 예약된 비트의 존재를 제어하는 gci_present_flag이다.

[0094]

단순화를 위해, gci_present_flag, gci_num_reserved_bits, gci_reserved_zero_bit[i] 및 gci_alignment_zero_bit를 제외한 GCI 구문 구조의 모든 구문 요소를 GCI 구문 요소라고 한다.

[0095]

일부 GCI 구문 구조의 의미는 다음과 같다.

[0096]

gci_present_flag가 1이면 GCI 구문 요소가 general_constraints_info() 구문 구조에 있음을 지정한다. gci_present_flag가 0이면 general_constraints_info() 구문 구조에 GCI 필드가 존재하지 않음을 지정한다.

[0097]

이 절에 지정된 GCI 구문 요소의 의미는 gci_present_flag가 1과 동일할 때 적용된다. gci_present_flag가 0이면, general_constraint_info() 구문 구조는 어떠한 제약도 부과하지 않는다.

[0098]

gci_intra_only_constraint_flag가 1이면InScope의 모든 슬라이스에 대한 sh_slice_type가 2와 동일해야 함을 지정한다. gci_intra_only_constraint_flag가 0이면 이러한 제약 조건을 부과하지 않는다.

[0099]

...

[0100]

2.4.VVC 범위 확장의 일반 제한 정보 (GCI) 플래그

[0101]

VVC 범위 확장 개발 중에 다음 6개의 새로운 GCI 플래그가 추가되었다.

[0102]

1) gci_all_rap_pictures_constraint_flag

- [0103] 2) gci_no_extended_precision_processing_constraint_flag
- [0104] 3) gci_no_ts_residual_coding_rice_constraint_flag
- [0105] 4) gci_no_rrc_rice_extension_constraint_flag
- [0106] 5) gci_no_persistent_rice_adaptation_constraint_flag
- [0107] 6) gci_no_reverse_last_sig_coeff_constraint_flag

[0108] 이러한 새로운 범위 확장 GCI 플래그를 포함하기 위해 변경된 GCI 구문 구조의 구문은 다음과 같다(추가 항목에는 밑줄이 그어져 있고, 삭제된 부분에는 취소선이 표시되어 있다).

general_constraints_info() {	설명자
gci_present_flag	u(1)
if(gci_present_flag) {	
<i>/* 일반적인 */</i>	
gci_intra_only_constraint_flag	u(1)
gci_no_virtual_boundaries_constraint_flag	u(1)
..	
gci_no_virtual_boundaries_constraint_flag	u(1)
gci_num_reserved_bits	u(8)
<u>gci_num_additional_bits</u>	<u>u(8)</u>
if(<u>gci_num_additional_bits</u> > 0) {	
<u>gci_all_rap_pictures_constraint_flag</u>	<u>u(1)</u>
<u>gci_no_extended_precision_processing_constraint_flag</u>	<u>u(1)</u>
<u>gci_no_ts_residual_coding_rice_constraint_flag</u>	<u>u(1)</u>
<u>gci_no_rrc_rice_extension_constraint_flag</u>	<u>u(1)</u>
<u>gci_no_persistent_rice_adaptation_constraint_flag</u>	<u>u(1)</u>
<u>gci_no_reverse_last_sig_coeff_constraint_flag</u>	<u>u(1)</u>
<u>numAdditionalBitsUsed = 6</u>	
} else	
<u>numAdditionalBitsUsed = 0</u>	
<u>for(i = 0; i < <u>gci_num_additional_bits</u> - <u>numAdditionalBitsUsed</u>; i++)</u>	
for(i = 0; i < gci_num_reserved_bits; i++)	
gci_reserved_zero_bit[i]	u(1)
}	
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
}	

- [0109]
- [0110] 단순화를 위해, gci_present_flag, gci_num_additional_bits, gci_reserved_zero_bit[i] 및 gci_alignment_zero_bit를 제외한 업데이트된 GCI 구문 구조의 모든 구문 요소를 GCI 구문 요소라고 한다.

[0111] GCI 구문 구조의 의미에 대한 변경 사항은 다음과 같다.

..

gci_num_reserved_bits 는 예약된 GCI 비트 수를 지정한다. 본 사양의 이 버전을 따르는 비트스트림에서 **gci_num_reserved_bits** 의 값은 0 과 동일해야 한다. **gci_num_reserved_bits** 이 다른 값은 향후 ITU-T ISO/IEC 에서 사용할 수 있도록 예약되어 있다. 본 사양의 이 버전에서는 **gci_num_reserved_bits** 의 값이 0 이어야 하지만, 본 사양의 이 버전을 준수하는 디코더는 0 보다 큰 **gci_num_reserved_bits** 의 값이 구문에 나타나는 것을 허용해야 하며, **gci_num_reserved_bit** 가 0 보다 큰 경우 모든 **gci_reserved_zero_bit[i]** 구문 요소의 값을 무시한다.

gci_num_additional_bits 는 **gci_alignment_zero_bit** 구문 요소(존재하는 경우) 이외의 일반 제약 정보 구문 구조에서 추가 GCI 비트 수를 지정한다. **gci_num_additional_bits** 의 값은 이 문서의 이 버전을 따르는 비트스트림에서 0 또는 1 과 같아야 한다. **gci_num_additional_bits** 에 대해 1 보다 큰 값은 향후 ITU-T ISO/IEC 에서 사용할 수 있도록 예약되어 있다. 이 문서의 이 버전에서 **gci_num_additional_bits** 의 값이 0 또는 1 이어야 하지만, 이 문서의 이 버전을 준수하는 디코더는 1 보다 큰 **gci_num_additional_bits** 값이 구문에 나타나는 것을 허용해야 하며, **gci_num_additional_bit** 가 1 보다 클 때 모든 **gci_reserved_zero_bit[i]** 구문 요소의 값을 무시한다.

gci_allRapPicturesConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처가 **ph_recovery_poc_cnt** 가 0 인 GDR 픽처 또는 IRAP 픽처임을 지정한다. **gci_allRapPicturesConstraintFlag** 가 0 이면 이러한 제약이 부과되지 않는다.

gci_noExtendedPrecisionProcessingConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처에 대한 **sps_extended_precision_flag** 가 0 과 같아야 함을 지정한다. **gci_noExtendedPrecisionProcessingConstraintFlag** 가 0 이면 이러한 제약이 부과되지 않는다.

gci_noTsResidualCodingRiceConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처에 대한 **sps_ts_residual_coding_rice_present_in_sh_flag** 가 0 과 같아야 함을 지정한다. **gci_noTsResidualCodingRiceConstraintFlag** 가 0 이면 이러한 제약이 부과되지 않는다.

[0112]

gci_noRrcRiceExtensionConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처에 대한 **sps_rrc_rice_extension_flag** 가 0 과 같아야 함을 지정한다. **gci_noRrcRiceExtensionConstraintFlag** 가 0 이면 이러한 제약이 부과되지 않는다.

gci_noPersistentRiceAdaptationConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처에 대한 **sps_persistent_rice_adaptation_enabled_flag** 가 0 과 같아야 함을 지정한다. **gci_noPersistentRiceAdaptationConstraintFlag** 가 0 이면 이러한 제한이 부과되지 않는다.

gci_noReverseLastSigCoeffConstraintFlag 가 1 이면 **OlsInScope** 의 모든 픽처에 대한 **sps_reverse_last_sig_coeff_enabled_flag** 가 0 과 같아야 함을 지정한다. **gci_noReverseLastSigCoeffConstraintFlag** 가 0 이면 이러한 제약이 부과되지 않는다.

[0113]

..

[0114] 3. 문제점

[0115] **gci_present_flag**가 0이면, 새로운 범위 확장 GCI 플래그를 포함한 GCI 구문 요소가 존재하지 않는다. 이 경우,

general_constraint_info() 구문 구조는 어떠한 제약도 부과하지 않는다. 그러나, gci_present_flag가 1인 경우, VVC 버전 1에 지정된 모든 GCI 구문 요소가 확실히 존재하더라도, 새로운 범위 확장 GCI 플래그는 gci_num_additional_bits(VVC 버전 1의 gci_num_reserved_bits)가 0보다 큰지 여부에 따라 존재할수 있고 존재하지 않을 수도 있다.

gci_present_flag가 1이고 gci_num_additional_bits가 0인 경우 문제가 발생하는데, 이 경우, 새로운 범위 확장 GCI 플래그가 존재하지 않고 해당 값이 추론되지 않으므로 1과 동일한 플래그로 지정된 제약 조건이 적용되는지 여부가 불분명하다.

4. 상세한 해결책

상기 문제를 해결하기 위해, 아래에 요약된 방법이 개시된다. 해결책은 일반적인 개념을 설명하기 위한 예시로 간주되어야 하며 협소하게 해석되어서는 안 된다. 또한 이러한 해결책은 개별적으로 적용되거나 어떤 방식으로든 조합될 수 있다.

1) 일 예에서, gci_present_flag가 1이고 gci_num_additional_bits가 0인 경우, 6개의 새로운 VVC 범위 확장 GCI 플래그(아래에 나열됨) 각각의 값은 0과 같은 것으로 추론된다.

a. gci_all_rap_pictures_constraint_flag

b. gci_no_extended_precision_processing_constraint_flag

c. gci_no_ts_residual_coding_rice_constraint_flag

d. gci_no_rrc_rice_extension_constraint_flag

e. gci_no_persistent_rice_adaptation_constraint_flag

f. gci_no_reverse_last_sig_coeff_constraint_flag

2) 대안적으로, 다른 예에서는 6개의 새로운 VVC 범위 확장 GCI 플래그 각각의 값이 존재하지 않을 때마다 0과 같은 것으로 추론된다.

3) 대안적으로, 또 다른 예에서는 VVC 버전 1 및 6개의 새로운 VVC 범위 확장 GCI 플래그에 지정된 요소를 포함하여 모든 GCI 구문 요소 각각의 값이 존재하지 않을 때마다 0과 같은 것으로 추론된다.

5. 실시예

다음은 섹션 4에 요약된 모든 해결책 측면에 대한 일부 예시적인 실시예이다. 이러한 실시예는 VVC에 적용될 수 있다. 변경 사항은 강조 표시되어 있고, 추가된 부분은 밑줄이 그어져 있으며 삭제된 부분은 취소선이 표시되어 있다.

5.1. 실시예 1

본 실시예는 섹션 4의 항목 1에 해당한다.

GCI 구문 구조의 의미는 다음과 같다.

gci_present_flag, gci_num_additional_bits, gci_reserved_zero_bit[i] 및 gci_alignment_zero_bit를 제외한 GCI 구문 구조의 모든 구문 요소를 GCI 구문 요소라고 한다.

gci_present_flag가 1이면 gci_intra_only_constraint_flag부터 gci_no_virtual_boundaries_constraint_flag까지의 GCI 구문 요소가 General_constraints_info() 구문 구조에 존재함을 지정한다. gci_present_flag가 0이면 GCI 플래그 구문 요소가 General_constraints_info() 구문 구조에 존재하지 않음을 지정한다.

이 절에 지정된 gci_intra_only_constraint_flag에서 gci_no_virtual_boundaries_constraint_flag까지의 GCI 구문 요소의 의미는 gci_present_flag가 1일 때 적용된다. gci_present_flag가 0이면 General_constraint_info() 구문 구조는 어떠한 제약도 부과하지 않는다.

- [0136] **gci_intra_only_constraint_flag**가 1이면 **0lsInScope**의 모든 슬라이스에 대한 **sh_slice_type**이 2와 동일해야 함을 지정한다. **gci_intra_only_constraint_flag**가 0이면 이러한 제약 조건이 부과되지 않는다.
- [0137] ..
- [0138] **gci_sixteen_minus_max_bitlength_constraint_idc**가 0보다 크면 **0lsInScope**의 모든 픽처에 대해 **sps_bitlength_minus8**에 8을 더한 값이 0에서 16 - **gci_sixteen_minus_max_bitepth_constraint_idc** 범위 내에 있어야 함을 지정한다. **gci_sixteen_minus_max_bitlength_constraint_idc**가 0이면 이러한 제약이 부과되지 않는다. **gci_sixteen_minus_max_bitlength_constraint_idc**의 값은 0부터 8까지의 범위에 있어야 한다.
- [0139] **gci_three_minus_max_chroma_format_constraint_idc**가 0보다 크면, **0lsInScope**의 모든 픽처에 대한 **sps_chroma_format_idc**가 0에서 3 - **gci_three_minus_max_chroma_format_constraint_idc** 범위 내에 있어야 함을 지정한다. **gci_three_minus_max_chroma_format_constraint_idc**가 0이면 이러한 제약이 부과되지 않는다.
- [0140] ..
- [0141] **gci_num_additional_bits**는 **gci_alignment_zero_bit** 구문 요소(존재하는 경우) 이외의 일반 제약 정보 구문 구조에서 추가 GCI 비트 수를 지정한다. **gci_num_additional_bits**의 값은 이 문서의 이 버전을 따르는 비트스트림에서 0 또는 1과 같아야 한다. **gci_num_additional_bits**에 대해 1보다 큰 값은 ITU-T | ISO/IEC에서 향후 사용을 위해 예약되어 있다. 이 문서 이 버전에서는 **gci_num_additional_bits** 값이 0 또는 1과 같아야 하지만, 이 문서의 이 버전을 준수하는 디코더는 1보다 큰 **gci_num_additional_bits** 값이 구문에 표시되도록 허용하고, **gci_num_additional_bits**가 1보다 큰 경우, 모든 **gci_reserved_zero_bit[i]** 구문 요소의 값을 무시한다.
- [0142] **gci_all_rap_pictures_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처가 **ph_recovery_poc_cnt**가 0인 GDR 픽처 또는 IRAP 픽처임을 지정한다. **gci_all_rap_pictures_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0이면 **gci_all_rap_pictures_constraint_flag**의 값은 0과 같다고 추론된다.
- [0143] **gci_no_extended_precision_processing_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처에 대한 **sps_extended_precision_flag**가 0과 같아야 함을 지정한다. **gci_no_extended_precision_processing_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0이면 **gci_no_extended_precision_processing_constraint_flag**의 값이 0과 같은 것으로 추론된다.
- [0144] **gci_no_ts_residual_coding_rice_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처에 대한 **sps_ts_residual_coding_rice_present_in_sh_flag**가 0과 같아야 함을 지정한다. **gci_no_ts_residual_coding_rice_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0이면 **gci_no_ts_residual_coding_rice_constraint_flag**의 값이 0과 같은 것으로 추론된다.
- [0145] **gci_no_rrc_rice_extension_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처에 대한 **sps_rrc_rice_extension_flag**가 0과 같아야 함을 지정한다. **gci_no_rrc_rice_extension_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0이면 **gci_no_rrc_rice_extension_constraint_flag**의 값은 0과 같은 것으로 추론된다.
- [0146] **gci_no_persistent_rice_adaptation_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처에 대한 **sps_persistent_rice_adaptation_enabled_flag**가 0과 같아야 함을 지정한다. **gci_no_persistent_rice_adaptation_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0이면 **gci_no_persistent_rice_adaptation_constraint_flag**의 값은 0과 같은 것으로 추론된다.
- [0147] **gci_no_reverse_last_sig_coeff_constraint_flag**가 1이면 **0lsInScope**의 모든 픽처에 대한 **sps_reverse_last_sig_coeff_enabled_flag**가 0과 같아야 함을 지정한다. **gci_no_reverse_last_sig_coeff_constraint_flag**가 0이면 이러한 제약이 부과되지 않는다. **gci_present_flag**가 1이고 **gci_num_additional_bits**가 0과 같으면 **gci_no_reverse_last_sig_coeff_constraint_flag**의 값이 0과 같은 것으로 추론된다.

- [0148] ..
- [0149] **5.2. 실시예 2**
- [0150] 본 실시예는 섹션 4의 항목 2에 해당한다.
- [0151] GCI 구문 구조의 의미는 다음과 같다.
- [0152] gci_present_flag, gci_num_additional_bits, gci_reserved_zero_bit[i] 및 gci_alignment_zero_bit를 제외한 GCI 구문 구조의 모든 구문 요소를 GCI 구문 요소라고 한다.
- gci_present_flag가 1이면 gci_intra_only_constraint_flag부터 gci_no_virtual_boundaries_constraint_flag까지의 GCI 구문 요소가 general_constraints_info() 구문 구조에 존재함을 지정한다. gci_present_flag가 0이면 GCI 필드 구문 요소가 general_constraints_info() 구문 구조에 존재하지 않음을 지정한다.
- [0153]
- [0154] 이 절에 지정된 gci_intra_only_constraint_flag에서 gci_no_virtual_boundaries_constraint_flag까지의 GCI 구문 요소의 의미는 gci_present_flag가 1일 때 적용된다. gci_present_flag가 0이면 general_constraint_info() 구문 구조는 어떠한 제약도 부과하지 않는다.
- [0155] gci_intra_only_constraint_flag가 1이면 0IsInScope의 모든 슬라이스에 대한 sh_slice_type이 2와 동일해야 함을 지정한다. gci_intra_only_constraint_flag가 0이면 이러한 제약이 부과되지 않는다.
- [0156] ..
- [0157] gci_sixteen_minus_max_bitlength_constraint_idc가 0보다 크면 0IsInScope의 모든 픽처에 대해 sps_bitlength_minus8에 8을 더한 값이 0에서 16 - gci_sixteen_minus_max_bitepth_constraint_idc 범위 내에 있어야 함을 지정한다. gci_sixteen_minus_max_bitlength_constraint_idc가 0이면 이러한 제약이 부과되지 않는다. gci_sixteen_minus_max_bitlength_constraint_idc의 값은 0부터 8까지의 범위에 있어야 한다.
- [0158] gci_three_minus_max_chroma_format_constraint_idc가 0보다 크면, 0IsInScope의 모든 픽처에 대해 sps_chroma_format_idc가 0에서 3 - gci_three_minus_max_chroma_format_constraint_idc 범위 내에 있어야 함을 지정한다. gci_three_minus_max_chroma_format_constraint_idc가 0이면 이러한 제약이 부과되지 않는다.
- [0159] ..
- [0160] gci_num_additional_bits는 gci_alignment_zero_bit 구문 요소(존재하는 경우) 이외의 일반 제약 정보 구문 구조에서 추가 GCI 비트 수를 지정한다. gci_num_additional_bits의 값은 이 문서의 이 버전을 따르는 비트스트림에서 0 또는 1과 같아야 한다. gci_num_additional_bits에 대해 1보다 큰 값은 ITU-T | ISO/IEC에서 향후 사용을 위해 예약되어 있다. 이 문서의 이 버전에서는 gci_num_additional_bits 값이 0 또는 1과 같아야 하지만, 이 문서의 이 버전을 준수하는 디코더는 1보다 큰 gci_num_additional_bits 값이 구문에 표시되도록 허용하고 gci_num_additional_bits가 1보다 큰 경우 모든 gci_reserved_zero_bit[i] 구문 요소의 값을 무시한다.
- [0161] gci_all_rap_pictures_constraint_flag가 1이면 0IsInScope의 모든 픽처가 ph_recovery_poc_cnt가 0인 GDR 픽처 또는 IRAP 픽처임을 지정한다. gci_all_rap_pictures_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우 gci_all_rap_pictures_constraint_flag 값이 0과 같은 것으로 추론된다.
- [0162] gci_no_extended_precision_processing_constraint_flag가 1이면 0IsInScope의 모든 픽처에 대해 sps_extended_precision_flag가 0과 같아야 함을 지정한다. gci_no_extended_precision_processing_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우 gci_no_extended_precision_processing_constraint_flag 값이 0과 같은 것으로 추론된다.
- [0163] gci_no_ts_residual_coding_rice_constraint_flag가 1이면 0IsInScope의 모든 픽처에 대해 sps_ts_residual_coding_rice_present_in_sh_flag가 0과 같아야 함을 지정한다. gci_no_ts_residual_coding_rice_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우, gci_no_ts_residual_coding_rice_constraint_flag 값이 0과 같은 것으로 추론된다.
- [0164] gci_no_rrc_rice_extension_constraint_flag가 1이면 0IsInScope의 모든 픽처에 대해

sps_rrc_rice_extension_flag가 0과 같아야 함을 지정한다. gci_no_rrc_rice_extension_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우, gci_no_rrc_rice_extension_constraint_flag 값이 0과 같은 것으로 추론된다.

[0165] gci_no_persistent_rice_adaptation_constraint_flag가 1이면 01sInScope의 모든 픽처에 대한 sps_persistent_rice_adaptation_enabled_flag가 0과 같아야 함을 지정한다. gci_no_persistent_rice_adaptation_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우, gci_no_persistent_rice_adaptation_constraint_flag 값이 0과 같은 것으로 추론된다.

[0166] gci_no_reverse_last_sig_coeff_constraint_flag가 1이면 01sInScope의 모든 픽처에 대한 sps_reverse_last_sig_coeff_enabled_flag가 0과 같아야 함을 지정한다. gci_no_reverse_last_sig_coeff_constraint_flag가 0이면 이러한 제약이 부과되지 않는다. 존재하지 않는 경우 gci_no_reverse_last_sig_coeff_constraint_flag 값이 0과 같은 것으로 추론된다.

[0167] ..

[0168] 5.3. 실시예 3

[0169] 본 실시예는 섹션 4의 항목 3에 해당한다.

[0170] GCI 구문 구조의 의미는 다음과 같다.

[0171] gci_present_flag, gci_num_additional_bits, gci_reserved_zero_bit[i] 및 gci_alignment_zero_bit를 제외한 GCI 구문 구조의 모든 구문 요소를 GCI 구문 요소라고 한다.

[0172] GCI 구문 요소 중 하나라도 존재하지 않으면 해당 GCI 구문 요소의 값이 0과 같은 것으로 추론된다.

[0173] gci_present_flag 가 1 이면 gci_intra_only_constraint_flag 부터 gci_no_virtual_boundaries_constraint_flag 까지의 해당 GCI 구문 요소가 general_constraints_info() 구문 구조에 존재함을 지정한다. gci_present_flag가 0이면 해당 GCI 필드 구문 요소가 General_constraints_info() 구문 구조에 존재하지 않음을 지정한다.

[0174] 이 절에 지정된 GCI 구문 요소의 의미는 gci_present_flag가 1일 때 적용된다. gci_present_flag가 0이면 General_constraint_info() 구문 구조는 어떠한 제약도 부과하지 않는다.

[0175] gci_intra_only_constraint_flag가 1이면 01sInScope의 모든 슬라이스에 대한 sh_slice_type가 2와 동일해야 함을 지정한다. gci_intra_only_constraint_flag가 0이면 이러한 제약이 부과되지 않는다.

[0176] ..

[0177] gci_sixteen_minus_max_bitlength_constraint_idc가 0보다 크면 01sInScope의 모든 픽처에 대해 sps_bitlength_minus8에 8을 더한 값이 0에서 16 - gci_sixteen_minus_max_bitdepth_constraint_idc 범위 내에 있어야 함을 지정한다. gci_sixteen_minus_max_bitlength_constraint_idc가 0이면 이러한 제약이 부과되지 않는다. gci_sixteen_minus_max_bitlength_constraint_idc의 값은 0부터 8까지의 범위에 있어야 한다.

[0178] gci_three_minus_max_chroma_format_constraint_idc가 0보다 크면 01sInScope의 모든 픽처에 대한 sps_chroma_format_idc가 0에서 3 - gci_three_minus_max_chroma_format_constraint_idc 범위 내에 있어야 함을 지정한다. gci_three_minus_max_chroma_format_constraint_idc가 0이면 이러한 제약이 부과되지 않는다.

[0179] ..

[0180] gci_num_additional_bits는 gci_alignment_zero_bit 구문 요소(존재하는 경우) 이외의 일반 제약 정보 구문 구조에서 추가 GCI 비트 수를 지정한다. gci_num_additional_bits의 값은 이 문서의 이 버전을 따르는 비트스트림에서 0 또는 1과 같아야 한다. gci_num_additional_bits에 대해 1보다 큰 값은 ITU-T | ISO/IEC에서 향후 사용을 위해 예약되어 있다. 이 문서의 이 버전에서는 gci_num_additional_bits 값이 0 또는 1과 같아야 하지만, 이 문서의 이 버전을 준수하는 디코더는 1보다 큰 gci_num_additional_bits 값이 구문에 표시되도록 허용하고 gci_num_additional_bits가 1보다 크면 모든 gci_reserved_zero_bit[i] 구문 요소의 값을 무시한다.

[0181] gci_all_rap_pictures_constraint_flag가 1이면 01sInScope의 모든 픽처는 ph_recovery_poc_cnt가 0인 GDR 픽

처 또는 IRAP 픽처임을 지정한다. `gci_all_rap_pictures_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0182] `gci_no_extended_precision_processing_constraint_flag`가 1이면 `OlInScope`의 모든 픽처에 대한 `sps_extended_precision_flag`가 0과 같아야 함을 지정한다. `gci_no_extended_precision_processing_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0183] `gci_no_ts_residual_coding_rice_constraint_flag`가 1이면 `OlInScope`의 모든 픽처에 대한 `sps_ts_residual_coding_rice_present_in_sh_flag`가 0과 같아야 함을 지정한다. `gci_no_ts_residual_coding_rice_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0184] `gci_no_rrc_rice_extension_constraint_flag`가 1이면 `OlInScope`의 모든 픽처에 대한 `sps_rrc_rice_extension_flag`가 0과 같아야 함을 지정한다. `gci_no_rrc_rice_extension_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0185] `gci_no_persistent_rice_adaptation_constraint_flag`가 1이면 `OlInScope`의 모든 픽처에 대한 `sps_persistent_rice_adaptation_enabled_flag`가 0과 같아야 함을 지정한다. `gci_no_persistent_rice_adaptation_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0186] `gci_no_reverse_last_sig_coeff_constraint_flag`가 1이면 `OlInScope`의 모든 픽처에 대한 `sps_reverse_last_sig_coeff_enabled_flag`가 0과 같아야 함을 지정한다. `gci_no_reverse_last_sig_coeff_constraint_flag`가 0이면 이러한 제약이 부과되지 않는다.

[0187] ..

[0188] 본 개시의 실시예는 VVC 범위 확장 일반 제약 정보 플래그의 시그널링에 관한 것이다.

[0189] 도 4는 본 개시의 일부 실시예에 따른 동영상 처리 방법(400)을 나타내는 흐름도이다. 도 4에 도시된 바와 같이, 단계 402에서, 동영상의 현재 동영상 블록과 동영상의 비트스트림 간의 변환은 동영상에 대한 일반 제약 정보(GCI)에 기초하여 수행된다. 본 명세서에서 사용되는 용어 "블록"은 코딩 트리 블록(CTB), 코딩 트리 유닛(CTU), 코딩 블록(CB), 코딩 유닛(CU), 예측 유닛(PU), 변환 유닛(TU), 예측 블록(PB), 변환 블록(TB), 다수의 샘플/픽셀 등을 포함하는 동영상 처리 유닛 등을 나타낼 수 있다. 블록은 직사각형 또는 직사각형이 아닐 수 있다. 일부 실시예에서, 변환은 현재 동영상 블록을 비트스트림으로 인코딩하는 단계를 포함할 수 있다. 추가로 또는 대안적으로, 변환은 비트 스트림으로부터 현재 동영상 블록을 디코딩하는 단계를 포함할 수 있다.

[0190] GCI는 비트스트림의 특정 제약 속성을 지시한다. GCI는, (1) 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, (2) 하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, (3) 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 지시되는지 여부를 지시하는 제3 구문 요소, (4) 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, (5) 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 (6) 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다. 일 예에서, 제1 미리 결정된 값은 0과 같을 수 있다. 제1 미리 결정된 값은 1과 같은 다른 임의의 적절한 값일 수 있음을 이해해야 한다. 본 개시의 범위는 이러한 점에서 제한되지 않는다.

[0191] 일부 실시예에서, 제1 구문 요소는 구문 요소 `gci_all_rap_pictures_constraint_flag`일 수 있고, 제2 구문 요소는 구문 요소 `gci_no_extended_precision_processing_constraint_flag`일 수 있으며, 제3 구문 요소는 구문 요소 `gci_no_ts_residual_coding_rice_constraint_flag`일 수 있고, 제4 구문 요소는 구문 요소 `gci_no_rrc_rice_extension_constraint_flag`일 수 있으며, 제5 구문 요소는 구문 요소 `gci_no_persistent_rice_adaptation_constraint_flag`일 수 있고, 및 제6 구문 요소는 구문 요소 `gci_no_reverse_last_sig_coeff_constraint_flag`일 수 있다. 예를 들어, 구문 요소

`gci_all_rap_pictures_constraint_flag`가 비트스트림에 없으면, 해당 값은 0으로 추론될 수 있다.

[0192] 위의 관점에서, 6개의 GCI 구문 요소 각각의 값은, 비트스트림에 존재하지 않는 경우, 미리 결정된 값(예를 들어, 0)과 동일한 것으로 추론된다. 제안된 방법은, 이러한 GCI 구문 요소의 값을 추론하지 않는 종래의 해결책에 비해, 이러한 GCI 구문 요소가 비트스트림에 없을 때에도 해당 값을 얻을 수 있기 때문에, 동영상 처리를 위한 균일한 프로세스를 확보하고 코딩 효율성을 향상시킬 수 있다는 이점이 있다.

[0193] 일부 실시예에서, GCI는 제7 구문 요소 및 제8 구문 요소를 추가로 포함할 수 있다. 제7 구문 요소는 GCI에 대한 구문 구조에 GCI 구문 요소가 존재하는지 여부를 지시할 수 있다. 제7 구문 요소는 제2 미리 결정된 값과 동일한 값을 가질 수 있다. 제8 구문 요소는 구문 요소 `gci_alignment_zero_bit` 외에 구문 구조에서 추가적인 GCI 비트 수를 지시할 수 있다. 제8 구문 요소는 제3 미리 결정된 값과 동일한 값을 가질 수 있다. 일 예에서, 제7 구문 요소는 구문 요소 `gci_present_flag`일 수 있고, 제8 구문 요소는 구문 요소 `gci_num_additional_bits`일 수 있다. 또한, 제2 미리 결정된 값은 1일 수 있고, 제3 미리 결정된 값은 0일 수 있다. 제2 및 제3 미리 결정된 값은 임의의 다른 적절한 값일 수 있다는 것을 이해해야 한다. 본 개시의 범위는 이러한 점에서 제한되지 않는다.

[0194] 대안적으로 또는 추가적으로, GCI는 제1 구문 요소, 제2 구문 요소, 제3 구문 요소, 제4 구문 요소, 제5 구문 요소 및 제6 구문 요소와는 다른 제9 구문 요소를 추가로 포함할 수 있다. 제9 구문 요소는 비트스트림에 없을 수 있고, 제1 미리 결정된 값과 동일한 값을 가질 수 있다. 예를 들어, 제9 구문 요소는 구문 요소 `gci_intra_only_constraint_flag`, 구문 요소 `gci_no_virtual_boundaries_constraint_flag`와 같이 VVC 버전 1에 지정된 GCI 구문 요소 중 하나일 수 있다. 구문 요소 `gci_intra_only_constraint_flag`가 비트스트림에 없을 경우, 해당 값은 0으로 추론될 수 있다. 위의 예시 및/또는 예는 단지 설명의 목적으로만 기재된 것임을 이해해야 한다. 본 개시의 범위는 이러한 점에서 제한되지 않는다.

[0195] 본 개시의 실시예에 따르면, 비밀시적 컴퓨터 판독 가능 기록 매체가 제안된다. 비밀시적 컴퓨터 판독 가능 기록 매체에는 동영상의 비트스트림이 저장된다. 비트스트림은 동영상 처리 장치에 의해 수행되는 방법을 통해 생성될 수 있다. 이 방법에 따르면, 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 동영상의 현재 동영상 블록과 동영상의 비트스트림 간의 변환이 수행된다. GCI는, 하나 이상의 출력 계층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지를 지시하는 제1 구문 요소, 하나 이상의 OLS에 있는 모든 픽처에 대해, 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS에 있는 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 서열 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 지시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS에 있는 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS에 있는 모든 픽처에 대해, 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 이전 TU에서 누적된 통계를 사용하여 각 변환 단위(TU)의 시작에서 초기화되는지 여부를 지시하는 제5 구문 요소 또는 하나 이상의 OLS에 있는 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소로 구성된다. 여기서, 상기 하나 이상의 구문 요소 각각은 비트스트림에 없으며 제1 미리 결정된 값과 동일한 값을 갖는다.

[0196] 본 발명의 실시예에서는 동영상의 비트스트림을 저장하는 방법을 제안한다. 이 방법은 동영상에 대한 일반 제약 정보(GCI)에 기초하여 동영상의 현재 동영상 블록과 동영상의 비트스트림 간의 변환을 수행하고, 비트스트림을 비밀시적 컴퓨터 판독 가능 기록 매체에 저장한다. GCI는 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 표시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지

시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 비트스트림에 없으며 제1 미리 결정된 값과 동일한 값을 갖는다.

[0197] 본 발명의 구현은 다음 조항을 고려하여 설명될 수 있으며, 그 특징은 임의의 합리적인 방식으로 조합될 수 있다.

[0198] 조항 1. 동영상 처리 방법으로서, 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함한다. 상기 GCI는 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복귀 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 표시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.

[0199] 조항 2. 조항 1의 방법에서, 상기 제1 미리 결정된 값은 0이다.

[0200] 조항 3. 조항 1 내지 조항 2 중 어느 한 조항의 방법에 있어서, 상기 제1 구문 요소는 구문 요소 `gci_all_rap_pictures_constraint_flag`이고, 상기 제2 구문 요소는 구문 요소 `gci_no_extended_precision_processing_constraint_flag`이며, 상기 제3 구문 요소는 구문 요소 `gci_no_ts_residual_coding_rice_constraint_flag`이고, 상기 제4 구문 요소는 구문 요소 `gci_no_rrc_rice_extension_constraint_flag`이며, 상기 제5 구문 요소는 구문 요소 `gci_no_persistent_rice_adaptation_constraint_flag`이고 및 상기 제6 구문 요소는 구문 요소 `gci_no_reverse_last_sig_coeff_constraint_flag`이다.

[0201] 조항 4. 조항 1 내지 조항 3 중 어느 한 조항의 방법에 있어서, 상기 GCI는 제7 구문 요소와 제8 구문 요소를 추가로 포함하고, 상기 제7 구문 요소는 GCI 구문 요소에 대한 구문 구조에 상기 GCI 구문 요소가 존재하는지 여부를 지시하며, 상기 제8 구문 요소는 구문 요소 `gci_alignment_zero_bit` 이외의 구문 구조에서 추가 GCI 비트 수를 지시하고, 상기 제7 구문 요소는 제2 미리 결정된 값과 동일한 값을 가지며, 상기 제8 구문 요소는 제3 미리 결정된 값과 동일한 값을 갖는다.

[0202] 조항 5. 조항 4의 방법에 있어서, 상기 제7 구문 요소는 구문 요소 `gci_present_flag`이고, 상기 제8 구문 요소는 구문 요소 `gci_num_additional_bits`이며, 상기 제2 미리 결정된 값은 1이고, 상기 제3 미리 결정된 값은 0이다.

[0203] 조항 6. 조항 1 내지 조항 3 중 어느 한 조항의 방법에 있어서, 상기 GCI는 제1 구문 요소, 제2 구문 요소, 제3 구문 요소, 제4 구문 요소, 제5 구문 요소 및 제6 구문 요소와 다른 제9 구문 요소를 추가로 포함하며, 상기 제9 구문 요소는 상기 비트스트림에 없고, 상기 제1 미리 결정된 값과 동일한 값을 갖는다.

[0204] 조항 7. 조항 1 내지 조항 6 중 어느 한 조항의 방법에 있어서, 상기 변환은 상기 현재 동영상 블록을 상기 비트스트림으로 인코딩하는 단계를 포함한다.

[0205] 조항 8. 조항 1 내지 조항 6 중 어느 한 조항의 방법에 있어서, 상기 변환은 상기 비트스트림으로부터 상기 현재 동영상 블록을 디코딩하는 단계를 포함한다.

[0206] 조항 9. 프로세서 및 명령을 포함하는 비일시적 메모리를 포함하는 동영상 데이터를 처리하는 장치로서, 상기 명령은 상기 프로세서에 의해 실행될 때 상기 프로세서로 하여금 조항 1 내지 조항 8 중 어느 한 조항의 방법을 수행하도록 한다.

[0207] 조항 10. 비일시적 컴퓨터 판독 가능 저장 매체에는 프로세서로 하여금 조항 1 내지 조항 8 중 어느 한 조항의 방법을 수행하도록 하는 명령이 저장된다.

- [0208] 조항 11. 동영상 처리 장치에 의해 수행되는 방법을 통해 생성된 동영상의 비트스트림을 저장하는 비밀시적 컴퓨터 판독 가능 기록 매체로서, 여기서 상기 방법은 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계를 포함한다. 상기 GCI는 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 표시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.
- [0209] 조항 12. 동영상의 비트스트림을 저장하는 방법은, 상기 동영상에 대한 일반 제약 정보(GCI)에 기초하여, 상기 동영상의 현재 동영상 블록과 상기 동영상의 비트스트림 간의 변환을 수행하는 단계로서, 상기 GCI는 하나 이상의 출력층 세트(OLS)의 모든 픽처가 복구 포인트 픽처인지 인트라 랜덤 액세스 포인트(IRAP) 픽처인지 여부를 지시하는 제1 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해 확장된 동적 범위가 스케일링 및 변환 프로세스에 사용되는지 여부와 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제2 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 현재 슬라이스의 구문 구조 `residual_ts_coding`에 사용되는 Rice 매개변수가 시퀀스 매개변수 세트(SPS)를 참조하는 구문 구조 `slice_header`에 표시되는지 여부를 지시하는 제3 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 대체 Rice 매개변수 유도가 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화에 사용되는지 여부를 지시하는 제4 구문 요소, 하나 이상의 OLS의 모든 픽처에 대해, 이전 변환 유닛(TU)으로부터 누적된 통계를 사용하여, 각 TU의 시작에서 구문 요소 `abs_remaining` 및 `dec_abs_level`의 이진화를 위한 Rice 매개변수 유도가 초기화되는지 여부를 지시하는 제5 구문 요소, 또는 하나 이상의 OLS의 모든 픽처에 대해, 구문 요소 `sh_reverse_last_sig_coeff_flag`가 상기 SPS를 참조하는 구문 구조 `slice_header`에 존재하는지 여부를 지시하는 제6 구문 요소 중 하나 이상의 구문 요소를 포함하는 단계 및 상기 비트스트림을 비밀시적 컴퓨터 판독 가능 매체에 저장하는 단계를 포함한다. 여기서, 상기 하나 이상의 구문 요소 각각은 상기 비트스트림에 존재하지 않고 제1 미리 결정된 값과 동일한 값을 갖는다.
- [0210] 예시 디바이스
- [0211] 도 5는 본 개시의 다양한 실시예가 구현될 수 있는 컴퓨팅 디바이스(500)를 나타내는 블록도이다. 컴퓨팅 디바이스(500)는 소스 디바이스(110)(또는 동영상 인코더(114 또는 200)) 또는 대상 디바이스(120)(또는 동영상 디코더(124 또는 300))로서 구현되거나 그 내에 포함될 수 있다.
- [0212] 도 5에 도시된 컴퓨팅 디바이스(500)는 단지 예시를 위한 것이며, 어떠한 방식으로든 본 개시의 실시예의 기능 및 범위에 대한 제한을 시사하지 않는다는 것이 이해될 것이다.
- [0213] 도 5에 도시된 바와 같이, 컴퓨팅 디바이스(500)는 범용 컴퓨팅 디바이스(500)를 포함한다. 컴퓨팅 디바이스(500)는 적어도 하나 이상의 프로세서 또는 처리 유닛(510), 메모리(520), 저장 유닛(530), 하나 이상의 통신 유닛(540), 하나 이상의 입력 디바이스(550) 및 하나 이상의 출력 디바이스(560)를 포함할 수 있다.
- [0214] 일부 실시예에서, 컴퓨팅 디바이스(500)는 컴퓨팅 기능이 있는 임의의 사용자 단말기 또는 서버 단말기로서 구현될 수 있다. 서버 단말기는 서비스 제공자가 제공하는 서버, 대규모 컴퓨팅 디바이스 등이 될 수 있다. 사용자 단말기에는 휴대 전화, 방송국, 유닛, 장치, 멀티미디어 컴퓨터, 멀티미디어 태블릿, 인터넷 노드, 커뮤니케이터, 데스크톱 컴퓨터, 랩톱 컴퓨터, 노트북 컴퓨터, 넷북 컴퓨터, 태블릿 컴퓨터, 개인 통신 시스템(PCS) 장치, 개인 내비게이션 장치, 개인 정보 단말기(PDA), 오디오/동영상 플레이어, 디지털 카메라/동영상 카메라, 위치 확인 장치, 텔레비전 수신기, 라디오 방송 수신기, 전자책 디바이스, 게임 디바이스 또는 이러한 디바이스의 액세서리 및 주변 장치 또는 이들의 조합을 포함한 임의의 유형의 이동 단말기, 고정 단말기 또는 휴대용 단말기일 수 있다. 컴퓨팅 디바이스(500)는 사용자에게 대한 임의의 유형의 인터페이스(예를 들어, "웨어러블" 회로

등)를 지원할 수 있는 것으로 고려된다.

- [0215] 처리 유닛(510)은 물리적 또는 가상의 프로세서일 수 있으며, 메모리(520)에 저장된 프로그램을 기반으로, 다양한 프로세스를 구현할 수 있다. 다중 프로세서 시스템에서, 컴퓨팅 디바이스(500)의 병렬 처리 기능을 향상시키기 위해, 다중 처리 유닛이 컴퓨터 실행 가능 명령을 병렬로 수행한다. 처리 유닛(510)은 중앙 처리 유닛(CPU), 마이크로프로세서, 컨트롤러 또는 마이크로컨트롤러라고도 지칭될 수 있다.
- [0216] 컴퓨팅 디바이스(500)는 일반적으로 다양한 컴퓨터 저장 매체를 포함한다. 이러한 매체는 휘발성 및 비휘발성 매체, 또는 분리 가능형 및 분리 불가능형 매체를 포함하지만 이에 제한되지 않는 컴퓨팅 디바이스(500)에 의해 액세스 가능한 임의의 매체일 수 있다. 메모리(520)는 휘발성 메모리(예를 들어, 레지스터, 캐시, 랜덤 액세스 메모리 (RAM)), 비휘발성 메모리(예를 들어, 읽기 전용 메모리(ROM)), 전기적으로 소거 가능한 프로그램 가능 읽기 전용 메모리(EEPROM) 또는 플래시 메모리) 또는 이들의 임의의 조합일 수 있다. 저장 유닛(530)은 임의의 분리 가능형 또는 분리 불가능형 매체일 수 있으며, 정보 및/또는 데이터를 저장하는 데 사용될 수 있고, 컴퓨팅 디바이스(500)에서 액세스될 수 있는 메모리, 플래시 메모리 드라이브, 자기 디스크 또는 다른 매체와 같은 기계 판독 가능 매체를 포함할 수 있다.
- [0217] 컴퓨팅 디바이스(500)는 추가적인 분리 가능형/분리 불가능형, 휘발성/비휘발성 메모리 매체를 추가로 포함할 수 있다. 도 5에는 도시되지 않았지만, 분리 가능형 비휘발성 자기 디스크에서 읽거나 쓰기 위한 자기 디스크 드라이브와, 분리 가능형 비휘발성 광 디스크에서 읽거나 쓰기 위한 광 디스크 드라이브를 제공할 수 있다. 이러한 경우, 각 드라이브는 하나 이상의 데이터 매체 인터페이스를 통해 버스(미도시)에 연결될 수 있다.
- [0218] 통신 유닛(540)은 통신 매체를 통해 추가 컴퓨팅 디바이스와 통신한다. 또한, 컴퓨팅 디바이스(500)의 구성요소 기능은 단일 컴퓨팅 클러스터 또는 통신 연결을 통해 통신할 수 있는 복수의 컴퓨팅 기계에 의해 구현될 수 있다. 따라서, 컴퓨팅 디바이스(500)는 하나 이상의 다른 서버, 네트워크에 연결된 개인용 컴퓨터(PC) 또는 추가적인 일반 네트워크 노드와의 논리적 연결을 사용하여 네트워크 환경에서 작동할 수 있다.
- [0219] 입력 디바이스(550)는 마우스, 키보드, 트랙킹 볼, 음성 입력 디바이스 등과 같은 다양한 입력 디바이스 중 하나 이상일 수 있다. 출력 디바이스(560)는 디스플레이, 확성기, 프린터 등과 같은 다양한 출력 디바이스 중 하나 이상일 수 있다. 통신 유닛(540)을 통해, 컴퓨팅 디바이스(500)는 저장 디바이스 및 디스플레이 디바이스와 같은 하나 이상의 외부 디바이스(미도시), 사용자가 컴퓨팅 디바이스(500)와 상호 작용할 수 있도록 하는 하나 이상의 디바이스, 또는 필요한 경우, 컴퓨팅 디바이스(500)가 하나 이상의 다른 컴퓨팅 디바이스와 통신할 수 있도록 하는 임의의 디바이스(네트워크 카드, 모뎀 등과 같은)와 추가로 통신할 수 있다. 이러한 통신은 입력/출력(I/O) 인터페이스(미도시)를 통해 수행될 수 있다.
- [0220] 일부 실시예에서, 단일 디바이스에 통합되는 대신, 컴퓨팅 디바이스(500)의 일부 또는 모든 구성요소는 클라우드 컴퓨팅 아키텍처로 배열될 수도 있다. 클라우드 컴퓨팅 아키텍처에서, 구성요소는 원격으로 제공될 수 있으며 함께 작동하여 본 개시에 설명된 기능을 구현할 수 있다. 일부 실시예에서, 클라우드 컴퓨팅은 컴퓨팅, 소프트웨어, 데이터 액세스 및 저장 서비스를 제공하며, 이러한 서비스를 제공하는 시스템 또는 하드웨어의 물리적 위치 또는 구성을 최종 사용자는 알 필요가 없다. 다양한 실시예에서, 클라우드 컴퓨팅은 적합한 프로토콜을 사용하여 광역 네트워크(인터넷 등)를 통해 서비스를 제공한다. 예를 들어, 클라우드 컴퓨팅 공급자는 웹 브라우저나 기타 컴퓨팅 구성요소를 통해 액세스할 수 있는 광역 네트워크를 통해 애플리케이션을 제공한다. 클라우드 컴퓨팅 아키텍처의 소프트웨어 또는 구성요소와 대응하는 데이터는, 원격 위치에 있는 서버에 저장될 수 있다. 클라우드 컴퓨팅 환경의 컴퓨팅 자원은 원격 데이터 센터의 위치에 병합되거나 분산될 수 있다. 클라우드 컴퓨팅 인프라는 사용자를 위한 단일 액세스 포인트로 작동하지만 공유 데이터 센터를 통해 서비스를 제공할 수 있다. 따라서 클라우드 컴퓨팅 아키텍처는, 본 명세서에 설명된 구성요소 및 기능을 원격 위치에 있는 서비스 제공업체로부터 제공하는 데 사용될 수 있다. 대안적으로, 이들은 종래의 서버로부터 제공될 수 있고, 또한 클라이언트 디바이스에 직접 또는 다른 방식으로 설치될 수도 있다.
- [0221] 컴퓨팅 디바이스(500)는 본 개시의 실시예에 있어서, 동영상 인코딩/디코딩을 구현하는 데 사용될 수 있다. 메모리(520)는 하나 이상의 프로그램 명령을 갖는 하나 이상의 동영상 코딩 모듈(525)을 포함할 수 있다. 이러한 모듈은 본 명세서에 설명된 다양한 실시예의 기능을 수행하기 위해 처리 유닛(510)에 의해 액세스 가능하고 실행 가능하다.
- [0222] 동영상 인코딩을 수행하는 예시적인 실시예에서, 입력 디바이스(550)는 인코딩될 동영상 데이터를 입력(570)으로 수신할 수 있다. 동영상 데이터는 예를 들어, 동영상 코딩 모듈(525)에 의해 처리되어 인코딩된 비트스트림

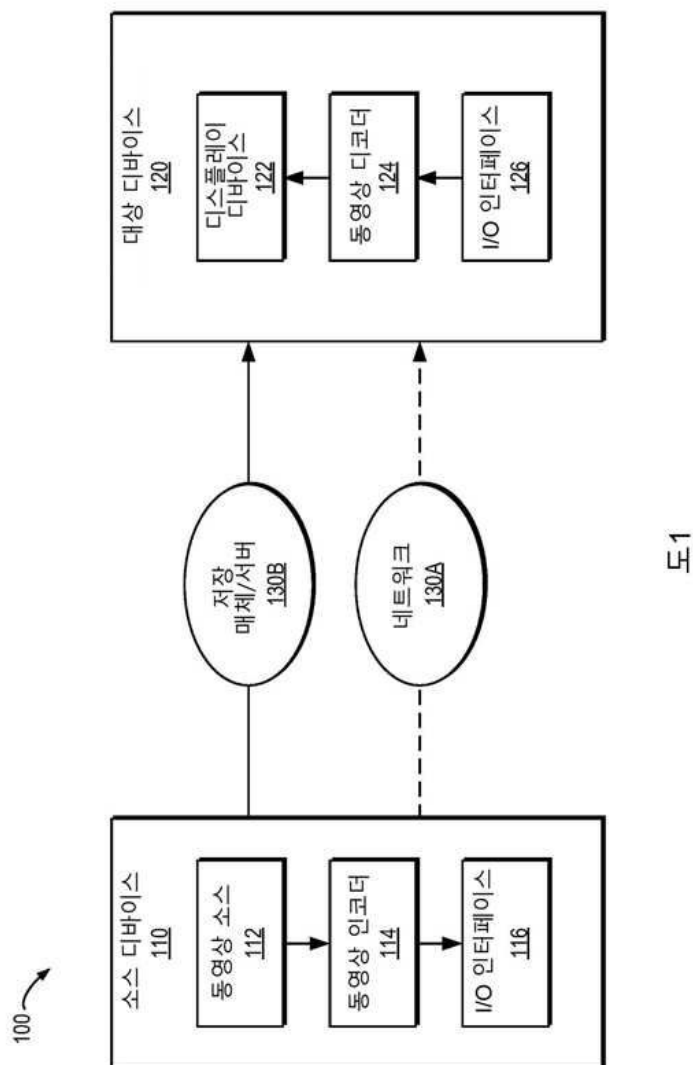
을 생성할 수 있다. 인코딩된 비트스트림은 출력 디바이스(560)를 통해 출력(580)으로 제공될 수 있다.

[0223] 동영상 디코딩을 수행하는 예시적인 실시예에서, 입력 디바이스(550)는 인코딩된 비트스트림을 입력(570)으로 수신할 수 있다. 인코딩된 비트스트림은 예를 들어, 동영상 코딩 모듈(525)에 의해 처리되어 디코딩된 동영상 데이터를 생성할 수 있다. 디코딩된 동영상 데이터는 출력 디바이스(560)를 통해 출력(580)으로 제공될 수 있다.

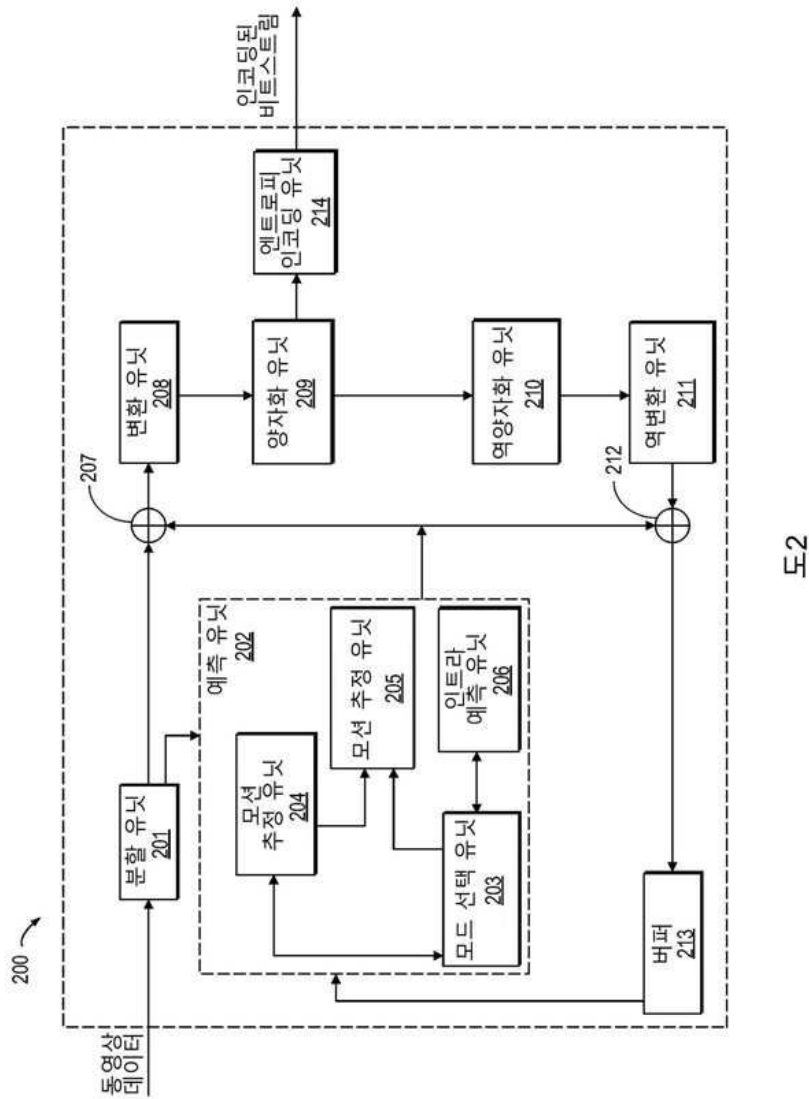
[0224] 본 개시는, 바람직한 실시예를 참조하여 특별히 도시되고 설명되었지만, 당업자는 첨부된 청구범위를 통해 정의된 바와 같은 본 출원의 사상 및 범위를 벗어나지 않으면서 형태 및 세부 사항의 다양한 변경이 이루어질 수 있음을 이해할 것이다. 이러한 변형은 본 출원의 범위에 포함되도록 의도된다. 따라서, 본 출원의 실시예에 대한 전술한 설명은 제한하려고 하는 의도가 없다.

도면

도면1

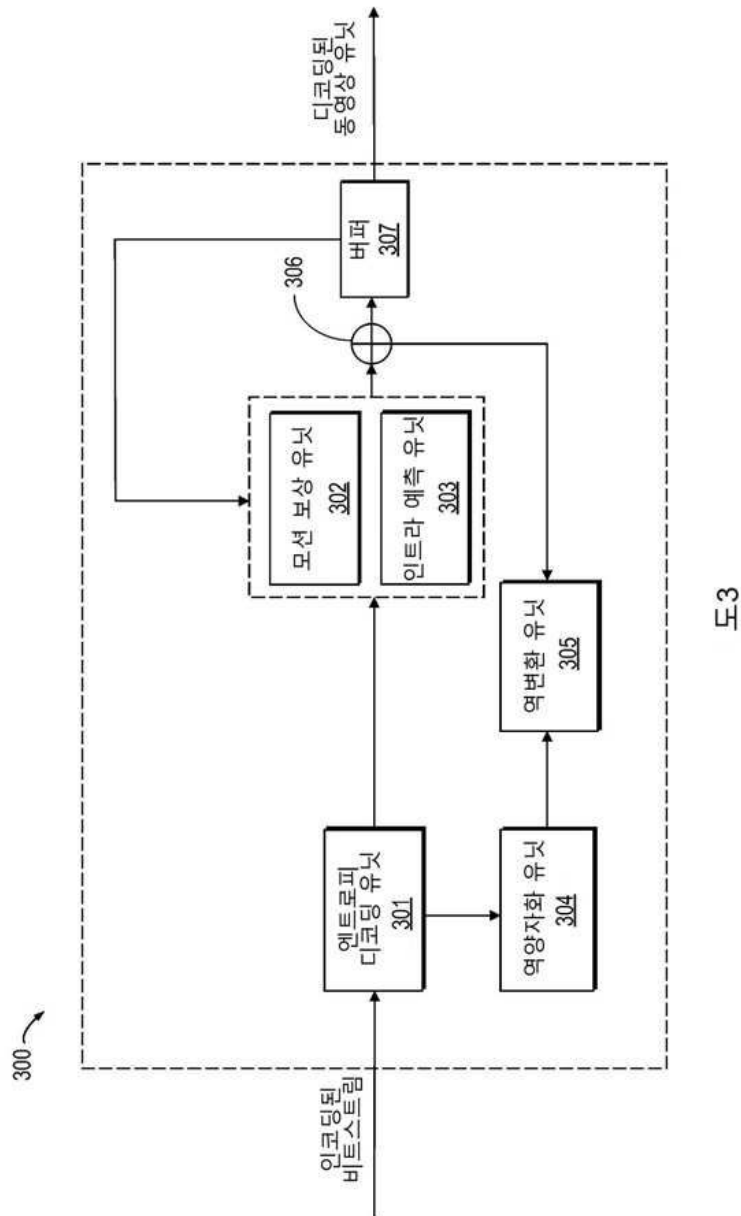


도면2



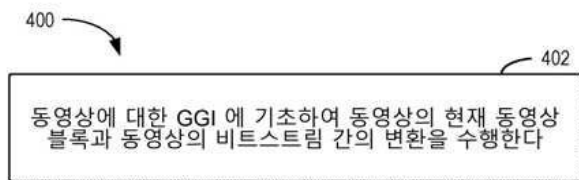
도2

도면3



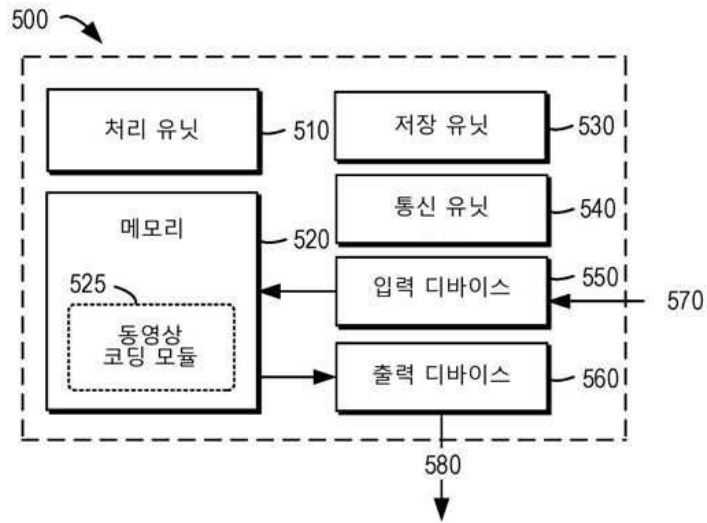
도3

도면4



도4

도면5



도5