

**(19) 대한민국특허청(KR)**
(12) 공개특허공보(A)**(11) 공개번호** 10-2020-0078501
(43) 공개일자 2020년07월01일

- (51) 국제특허분류(Int. Cl.)
H04N 19/129 (2014.01) H04N 19/132 (2014.01)
H04N 19/18 (2014.01) H04N 19/60 (2014.01)
H04N 19/70 (2014.01)
- (52) CPC특허분류
H04N 19/129 (2015.01)
H04N 19/132 (2015.01)
- (21) 출원번호 10-2020-7011528
- (22) 출원일자(국제) 2019년12월13일
심사청구일자 2020년04월21일
- (85) 번역문제출일자 2020년04월21일
- (86) 국제출원번호 PCT/KR2019/017723
- (87) 국제공개번호 WO 2020/130514
국제공개일자 2020년06월25일
- (30) 우선권주장
62/780,925 2018년12월17일 미국(US)
62/792,826 2019년01월15일 미국(US)
- (71) 출원인
엘지전자 주식회사
서울특별시 영등포구 여의대로 128 (여의도동)
- (72) 발명자
최정아
서울시 서초구 양재대로11길 19 LG전자 특허센터
김승환
서울시 서초구 양재대로11길 19 LG전자 특허센터
(뒷면에 계속)
- (74) 대리인
인비전 특허법인

전체 청구항 수 : 총 15 항

(54) 발명의 명칭 고주파 제로잉을 기반으로 변환 계수 스캔 순서를 결정하는 방법 및 장치**(57) 요약**

본 개시에 따른 디코딩 장치에 의하여 수행되는 영상 디코딩 방법은, 레지듀얼 정보를 포함하는 비트스트림을 수신하는 단계, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 단계, 역양자화 과정에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 단계, 상기 도출된 변환 계수들에 역변환을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계 및 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 단계를 포함하는 것을 특징으로 한다.

(52) CPC특허분류

H04N 19/18 (2015.01)

H04N 19/60 (2015.01)

H04N 19/70 (2015.01)

(72) 발명자

허진

서울시 서초구 양재대로11길 19 LG전자 특허센터

유선미

서울시 서초구 양재대로11길 19 LG전자 특허센터

이령

서울시 서초구 양재대로11길 19 LG전자 특허센터

최장원

서울시 서초구 양재대로11길 19 LG전자 특허센터

명세서

청구범위

청구항 1

디코딩 장치에 의하여 수행되는 영상 디코딩 방법에 있어서,

레지듀얼 정보를 포함하는 비트스트림을 수신하는 단계;

상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 단계;

역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 단계;

상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계; 및

상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 단계를 포함하되,

상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고,

상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝(transform coefficient scanning)이 수행되는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 2

제1항에 있어서,

상기 레지듀얼 정보는 상기 현재 블록 내 서브 블록에 대한 변환 계수들의 변환 계수 레벨이 모두 0인지 여부를 나타내는 코디드 서브 블록 플래그(coded subblock flag)를 포함하고,

상기 서브 블록은 상기 저주파 변환 계수 영역과 관련되는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 3

제1항에 있어서,

상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대해서는 상기 변환 계수 스캐닝이 수행되지 않는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 4

제1항에 있어서,

상기 저주파 변환 계수 영역의 폭 또는 상기 저주파 변환 계수 영역의 높이는 32 이하인 것을 특징으로 하는, 영상 디코딩 방법.

청구항 5

제4항에 있어서,

상기 현재 블록의 폭이 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 상기 현재 블록의 상기 폭과 동일한 것으로 결정되고,

상기 현재 블록의 높이가 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 상기 현재 블록의 상기 높이와 동일한 것으로 결정되고,

상기 현재 블록의 상기 폭이 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 32로 결정되고,

상기 현재 블록의 상기 높이가 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 32

로 결정되는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 6

제5항에 있어서,

상기 저주파 변환 계수 영역의 상기 폭 및 상기 저주파 변환 계수 영역의 상기 높이는 아래의 수학적식을 기반으로 결정되고,

$$\log_2(\text{ZoTbWidth}) = \text{Min}(\log_2(\text{TbWidth}), 5)$$

$$\log_2(\text{ZoTbHeight}) = \text{Min}(\log_2(\text{TbHeight}), 5)$$

상기 제5항의 수학적식에서 상기 ZoTbWidth는 상기 저주파 변환 계수 영역의 상기 폭을 나타내고, 상기 ZoTbHeight는 상기 저주파 변환 계수 영역의 상기 높이를 나타내고, 상기 TbWidth는 상기 현재 블록의 상기 폭을 나타내고, 상기 TbHeight는 상기 현재 블록의 상기 높이를 나타내는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 7

제4항에 있어서,

상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대하여 제로잉(zeroing)이 적용되고, 상기 제로잉이 적용된 변환 계수들의 개수는 상기 저주파 변환 계수 영역의 상기 폭 또는 상기 저주파 변환 계수 영역의 상기 높이에 기반하여 결정되는 것을 특징으로 하는, 영상 디코딩 방법.

청구항 8

인코딩 장치에 의하여 수행되는 영상 인코딩 방법에 있어서,

현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계;

상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출하는 단계;

양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출하는 단계; 및

상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩하는 단계를 포함하되,

상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고,

상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝이 수행되는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 9

제8항에 있어서,

상기 레지듀얼 정보는 상기 현재 블록 내 서브 블록에 대한 변환 계수들의 변환 계수 레벨이 모두 0인지 여부를 나타내는 코디드 서브 블록 플래그(coded subblock flag)를 포함하고,

상기 서브 블록은 상기 저주파 변환 계수 영역과 관련되는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 10

제8항에 있어서,

상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대해서는 상기 변환 계수 스캐닝이 수행되지 않는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 11

제8항에 있어서,

상기 저주파 변환 계수 영역의 폭 또는 상기 저주파 변환 계수 영역의 높이는 32 이하인 것을 특징으로 하는,

영상 인코딩 방법.

청구항 12

제11항에 있어서,

상기 현재 블록의 폭이 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 상기 현재 블록의 상기 폭과 동일한 것으로 결정되고,

상기 현재 블록의 높이가 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 상기 현재 블록의 상기 높이와 동일한 것으로 결정되고,

상기 현재 블록의 상기 폭이 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 32로 결정되고,

상기 현재 블록의 상기 높이가 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 32로 결정되는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 13

제12항에 있어서,

상기 저주파 변환 계수 영역의 폭 및 상기 저주파 변환 계수 영역의 상기 높이는 아래의 수학적식을 기반으로 결정되고,

$$\log_2(\text{ZoTbWidth}) = \text{Min}(\log_2(\text{TbWidth}), 5)$$

$$\log_2(\text{ZoTbHeight}) = \text{Min}(\log_2(\text{TbHeight}), 5)$$

상기 제5항의 수학적식에서 상기 ZoTbWidth는 상기 저주파 변환 계수 영역의 상기 폭을 나타내고, 상기 ZoTbHeight는 상기 저주파 변환 계수 영역의 상기 높이를 나타내고, 상기 TbWidth는 상기 현재 블록의 상기 폭을 나타내고, 상기 TbHeight는 상기 현재 블록의 상기 높이를 나타내는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 14

제11항에 있어서,

상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대하여 제로잉이 적용되고, 상기 제로잉이 적용된 변환 계수들의 개수는 상기 저주파 변환 계수 영역의 상기 폭 또는 상기 저주파 변환 계수 영역의 상기 높이에 기반하여 결정되는 것을 특징으로 하는, 영상 인코딩 방법.

청구항 15

영상 디코딩을 수행하는 디코딩 장치에 있어서,

레지듀얼 정보를 포함하는 비트스트림을 수신하고, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 엔트로피 디코딩부;

역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 역양자화부;

상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 역변환부; 및

상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 가산부를 포함하되,

상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고,

상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝이 수행되는 것을 특징으로 하는, 디코딩 장치.

발명의 설명

기술 분야

[0001] 본 개시는 영상 코딩 기술에 관한 것으로서 보다 상세하게는 영상 코딩 시스템에서 고주파 제로잉(high frequency zeroing)을 기반으로 변환 계수 스캔 순서(transform coefficient scan order)를 결정하는 방법 및 장치에 관한 것이다.

배경 기술

[0002] 최근 4K 또는 8K 이상의 UHD(Ultra High Definition) 영상/비디오와 같은 고해상도, 고품질의 영상/비디오에 대한 수요가 다양한 분야에서 증가하고 있다. 영상/비디오 데이터가 고해상도, 고품질이 될수록 기존의 영상/비디오 데이터에 비해 상대적으로 전송되는 정보량 또는 비트량이 증가하기 때문에 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 데이터를 전송하거나 기존의 저장 매체를 이용해 영상/비디오 데이터를 저장하는 경우, 전송 비용과 저장 비용이 증가된다.

[0003] 또한, 최근 VR(Virtual Reality), AR(Artificial Reality) 콘텐츠나 홀로그램 등의 실감 미디어(Immersive Media)에 대한 관심 및 수요가 증가하고 있으며, 게임 영상과 같이 현실 영상과 다른 영상 특성을 갖는 영상/비디오에 대한 방송이 증가하고 있다.

[0004] 이에 따라, 상기와 같은 다양한 특성을 갖는 고해상도 고품질의 영상/비디오의 정보를 효과적으로 압축하여 전송하거나 저장하고, 재생하기 위해 고효율의 영상/비디오 압축 기술이 요구된다.

발명의 내용

해결하려는 과제

[0005] 본 개시의 기술적 과제는 영상 코딩 효율을 높이는 방법 및 장치를 제공함에 있다.

[0006] 본 개시의 다른 기술적 과제는 레지듀얼 코딩의 효율을 높이는 방법 및 장치를 제공함에 있다.

[0007] 본 개시의 또 다른 기술적 과제는 변환 계수 레벨 코딩의 효율을 높이는 방법 및 장치를 제공함에 있다.

[0008] 본 개시의 또 다른 기술적 과제는 고주파 제로잉을 기반으로 변환 계수를 코딩하여 레지듀얼 코딩 효율을 높이는 방법 및 장치를 제공함에 있다.

[0009] 본 개시의 또 다른 기술적 과제는 고주파 제로잉을 기반으로 현재 블록(또는 현재 변환 블록)에 대한 변환 계수들을 코딩할 때, 상기 현재 블록에서 상기 고주파 제로잉이 적용되지 않는 영역의 정보를 기반으로 변환 계수 그룹을 스캔하는 방법을 변경하는 방법 및 장치를 제공함에 있다.

[0010] 본 개시의 또 다른 기술적 과제는 고주파 제로잉이 적용되는 영역과 관련된 변환 계수 그룹에 대한 변환 계수 스캔을 생략하고, 상기 변환 계수 그룹에 대한 코디드 서브 블록 플래그(coded subblock flag)의 시그널링을 생략하는 방법 및 장치를 제공함에 있다.

과제의 해결 수단

[0011] 본 개시의 일 실시예에 따르면, 디코딩 장치에 의하여 수행되는 영상 디코딩 방법이 제공된다. 상기 방법은, 레지듀얼 정보를 포함하는 비트스트림을 수신하는 단계, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 단계, 역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 단계, 상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계 및 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 단계를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝이 수행되는 것을 특징으로 한다.

[0012] 본 개시의 다른 일 실시예에 따르면, 영상 디코딩을 수행하는 디코딩 장치가 제공된다. 상기 디코딩 장치는, 레지듀얼 정보를 포함하는 비트스트림을 수신하고, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 엔트로피 디코딩부, 역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 역양자

화부, 상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 역변환부 및 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 가산부를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스케닝이 수행되는 것을 특징으로 한다.

[0013] 본 개시의 또 다른 일 실시예에 따르면, 인코딩 장치에 의하여 수행되는 영상 인코딩 방법이 제공된다. 상기 방법은, 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계, 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출하는 단계, 양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출하는 단계 및 상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩하는 단계를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스케닝이 수행되는 것을 특징으로 한다.

[0014] 본 개시의 또 다른 일 실시예에 따르면, 영상 인코딩을 수행하는 인코딩 장치가 제공된다. 상기 인코딩 장치는, 현재 블록에 대한 레지듀얼 샘플들을 도출하는 감산부, 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출하는 변환부, 양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출하는 양자화부 및 상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩하는 엔트로피 인코딩부를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스케닝이 수행되는 것을 특징으로 한다.

[0015] 본 개시의 또 다른 일 실시예에 따르면, 비디오 디코딩 장치가 일부 실시예들에 따른 디코딩 방법들을 수행하도록 야기하는 지시들(instructions)에 대한 정보를 저장하는, 디코더로 판독 가능한 저장 매체가 제공된다.

[0016] 본 개시의 또 다른 일 실시예에 따르면, 비디오 디코딩 장치가 일 실시예에 따른 디코딩 방법을 수행하도록 야기하는 지시들에 대한 정보를 저장하는, 디코더로 판독 가능한 저장 매체가 제공된다. 상기 일 실시예에 따른 디코딩 방법은, 레지듀얼 정보를 포함하는 비트스트림을 수신하는 단계, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 단계, 역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 단계, 상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계 및 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 단계를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스케닝이 수행되는 것을 특징으로 한다.

발명의 효과

[0017] 본 개시에 따르면 전반적인 영상/비디오 압축 효율을 높일 수 있다.

[0018] 본 개시에 따르면 레지듀얼 코딩의 효율을 높일 수 있다.

[0019] 본 개시에 따르면 변환 계수 레벨 코딩의 효율을 높일 수 있다.

[0020] 본 개시에 따르면 고주파 제로잉을 기반으로 변환 계수를 코딩하여 레지듀얼 코딩 효율을 높일 수 있다.

[0021] 본 개시에 따르면 고주파 제로잉을 기반으로 현재 블록(또는 현재 변환 블록) 내 라스트 유효 변환 계수의 위치 정보를 코딩하여 영상 코딩 효율을 높일 수 있다.

[0022] 본 개시에 따르면 고주파 제로잉을 적용할 시 고주파 제로잉 영역(보다 정확하게는, 고주파 제로잉이 적용되지 않는 영역)의 사이즈를 기반으로 선택 요소에 대한 이진화를 수행함으로써 보다 효율적인 부호화를 수행하고, 문맥 부호화 빈(context-coded bin)의 개수를 절감함으로써 CABAC의 처리량(throughput)을 향상시킬 수 있다.

[0023] 본 개시에 따르면 고주파 제로잉을 기반으로 현재 블록(또는 현재 변환 블록)에 대한 변환 계수들을 코딩할 때, 상기 현재 블록에서 상기 고주파 제로잉이 적용되지 않는 영역의 정보를 기반으로 변환 계수 그룹을 스캔함으로써 영상 코딩 효율을 높일 수 있다.

[0024] 본 개시에 따르면 고주파 제로잉이 적용되는 영역과 관련된 변환 계수 그룹에 대한 변환 계수 스캔을 생략하고, 상기 변환 계수 그룹에 대한 코디드 서브 블록 플래그(coded subblock flag)의 시그널링을 생략함으로써 영상 코딩 효율을 높일 수 있다.

도면의 간단한 설명

- [0025] 도 1은 본 개시가 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.
- 도 2는 본 개시가 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다.
- 도 3은 본 개시가 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다.
- 도 4는 일 실시예에 따른 고주파 제로잉을 설명하기 위한 도면이다.
- 도 5는 고주파 제로잉이 적용된 현재 블록에 대한 변환 계수들을 스캔하는 방법의 일 예시를 설명하기 위한 도면이다.
- 도 6은 고주파 제로잉이 적용된 현재 블록에 대한 변환 계수들을 스캔하는 방법의 다른 일 예시를 설명하기 위한 도면이다.
- 도 7a 및 도 7b는 일 실시예에 따른 엔트로피 인코딩부의 구성 및 동작을 설명하기 위한 도면이다.
- 도 8a 및 도 8b는 일 실시예에 따른 엔트로피 디코딩부의 구성 및 동작 방법을 설명하기 위한 도면이다.
- 도 9는 일 실시예에 따른 인코딩 장치의 동작을 도시하는 흐름도이다.
- 도 10은 일 실시예에 따른 인코딩 장치의 구성을 도시하는 블록도이다.
- 도 11은 일 실시예에 따른 디코딩 장치의 동작을 도시하는 흐름도이다.
- 도 12는 일 실시예에 따른 디코딩 장치의 구성을 도시하는 블록도이다.
- 도 13은 본 문서의 개시가 적용될 수 있는 콘텐츠 스트리밍 시스템의 예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0026] **발명의 실시를 위한 최선의 형태**

[0027] 본 개시의 일 실시예에 따르면, 디코딩 장치에 의하여 수행되는 영상 디코딩 방법이 제공된다. 상기 방법은, 레지듀얼 정보를 포함하는 비트스트림을 수신하는 단계, 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하는 단계, 역양자화 과정(inverse quantization process)에 기반하여 상기 양자화된 변환 계수들로부터 상기 현재 블록에 대한 변환 계수들을 도출하는 단계, 상기 도출된 변환 계수들에 역변환(inverse transform)을 적용하여 상기 현재 블록에 대한 레지듀얼 샘플들을 도출하는 단계 및 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 기반으로 복원 픽처를 생성하는 단계를 포함하되, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역과 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝이 수행되는 것을 특징으로 한다.

[0028] **발명의 실시를 위한 형태**

[0029] 본 개시는 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 개시를 특정 실시예에 한정하려고 하는 것이 아니다. 본 명세서에서 상용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 개시의 기술적 사상을 한정하려는 의도로 사용되는 것은 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서 "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는

것으로 이해되어야 한다.

- [0030] 한편, 본 개시에서 설명되는 도면상의 각 구성들은 서로 다른 특징적인 기능들에 관한 설명의 편의를 위해 독립적으로 도시된 것으로서, 각 구성들이 서로 별개의 하드웨어나 별개의 소프트웨어로 구현된다는 것을 의미하지는 않는다. 예컨대, 각 구성 중 두 개 이상의 구성이 합쳐져 하나의 구성을 이룰 수도 있고, 하나의 구성이 복수의 구성으로 나뉘어질 수도 있다. 각 구성이 통합 및/또는 분리된 실시예도 본 개시의 본질에서 벗어나지 않는 한 본 개시의 권리범위에 포함된다.
- [0031] 이하, 첨부한 도면들을 참조하여, 본 개시의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 이하, 도면상의 동일한 구성 요소에 대해서는 동일한 참조 부호를 사용하고 동일한 구성 요소에 대해서 중복된 설명은 생략될 수 있다.
- [0032] 도 1은 본 개시를 적용될 수 있는 비디오/영상 코딩 시스템의 예를 개략적으로 나타낸다.
- [0033] 이 문서는 비디오/영상 코딩에 관한 것이다. 예를 들어 이 문서에서 개시된 방법/실시예는 VVC (versatile video coding) 표준, EVC (essential video coding) 표준, AV1 (AOMedia Video 1) 표준, AVS2 (2nd generation of audio video coding standard) 또는 차세대 비디오/영상 코딩 표준(ex. H.267 or H.268 등)에 개시되는 방법에 적용될 수 있다.
- [0034] 이 문서에서는 비디오/영상 코딩에 관한 다양한 실시예들을 제시하며, 다른 언급이 없는 한 상기 실시예들은 서로 조합되어 수행될 수도 있다.
- [0035] 이 문서에서 비디오(video)는 시간의 흐름에 따른 일련의 영상(image)들의 집합을 의미할 수 있다. 픽처(picture)는 일반적으로 특정 시간대의 하나의 영상을 나타내는 단위를 의미하며, 슬라이스(slice)/타일(tile)은 코딩에 있어서 픽처의 일부를 구성하는 단위이다. 슬라이스/타일은 하나 이상의 CTU(coding tree unit)을 포함할 수 있다. 하나의 픽처는 하나 이상의 슬라이스/타일로 구성될 수 있다. 하나의 픽처는 하나 이상의 타일 그룹으로 구성될 수 있다. 하나의 타일 그룹은 하나 이상의 타일들을 포함할 수 있다. 브릭은 픽처 내 타일 이내의 CTU 행들의 사각 영역을 나타낼 수 있다(a brick may represent a rectangular region of CTU rows within a tile in a picture). 타일은 다수의 브릭들로 파티셔닝될 수 있고, 각 브릭은 상기 타일 내 하나 이상의 CTU 행들로 구성될 수 있다(A tile may be partitioned into multiple bricks, each of which consisting of one or more CTU rows within the tile). 다수의 브릭들로 파티셔닝되지 않은 타일은 또한 브릭으로 불릴 수 있다(A tile that is not partitioned into multiple bricks may be also referred to as a brick). 브릭 스캔은 픽처를 파티셔닝하는 CTU들의 특정한 순차적 오더링을 나타낼 수 있으며, 상기 CTU들은 브릭 내에서 CTU 래스터 스캔으로 정렬될 수 있고, 타일 내 브릭들은 상기 타일의 상기 브릭들의 래스터 스캔으로 연속적으로 정렬될 수 있고, 그리고 픽처 내 타일들은 상기 픽처의 상기 타일들의 래스터 스캔으로 연속적으로 정렬될 수 있다(A brick scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a brick, bricks within a tile are ordered consecutively in a raster scan of the bricks of the tile, and tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture). 타일은 특정 타일 열 및 특정 타일 열 이내의 CTU들의 사각 영역이다(A tile is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture). 상기 타일 열은 CTU들의 사각 영역이고, 상기 사각 영역은 상기 픽처의 높이와 동일한 높이를 갖고, 너비는 픽처 파라미터 세트 내의 선택스 요소들에 의하여 명시될 수 있다(The tile column is a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set). 상기 타일 행은 CTU들의 사각 영역이고, 상기 사각 영역은 픽처 파라미터 세트 내의 선택스 요소들에 의하여 명시되는 너비를 갖고, 높이는 상기 픽처의 높이와 동일할 수 있다(The tile row is a rectangular region of CTUs having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture). 타일 스캔은 픽처를 파티셔닝하는 CTU들의 특정 순차적 오더링을 나타낼 수 있고, 상기 CTU들은 타일 내 CTU 래스터 스캔으로 연속적으로 정렬될 수 있고, 픽처 내 타일들은 상기 픽처의 상기 타일들의 래스터 스캔으로 연속적으로 정렬될 수 있다(A tile scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture). 슬라이스는 픽처의 정수개의 브릭들을 포함할 수 있고, 상기 정수개의 브릭들은 하나의 NAL 유닛에 포함될 수 있다(A slice includes an integer number of bricks of a picture that may be exclusively contained in a single NAL unit). 슬라이스는 다수의 완전한 타일들로 구성

될 수 있고, 또는 하나의 타일의 완전한 브릭들의 연속적인 시퀀스일 수도 있다(A slice may consists of either a number of complete tiles or only a consecutive sequence of complete bricks of one tile). 이 문서에서 타일 그룹과 슬라이스는 혼용될 수 있다. 예를 들어 본 문서에서 tile group/tile group header는 slice/slice header로 불리 수 있다.

- [0036] 픽셀(pixel) 또는 펠(pe)은 하나의 픽처(또는 영상)을 구성하는 최소의 단위를 의미할 수 있다. 또한, 픽셀에 대응하는 용어로서 '샘플(sample)'이 사용될 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 루마(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 크로마(chroma) 성분의 픽셀/픽셀 값을 나타낼 수도 있다.
- [0037] 유닛(unit)은 영상 처리의 기본 단위를 나타낼 수 있다. 유닛은 픽처의 특정 영역 및 해당 영역에 관련된 정보 중 적어도 하나를 포함할 수 있다. 하나의 유닛은 하나의 루마 블록 및 두개의 크로마(ex. cb, cr) 블록을 포함할 수 있다. 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, MxN 블록은 M개의 열과 N개의 행으로 이루어진 샘플들(또는 샘플 어레이) 또는 변환 계수(transform coefficient)들의 집합(또는 어레이)을 포함할 수 있다.
- [0038] 이 문서에서 "/"와 ","는 "및/또는"으로 해석된다. 예를 들어, "A/B"는 "A 및/또는 B"로 해석되고, "A, B"는 "A 및/또는 B"로 해석된다. 추가적으로, "A/B/C"는 "A, B 및/또는 C 중 적어도 하나"를 의미한다. 또한, "A, B, C"도 "A, B 및/또는 C 중 적어도 하나"를 의미한다. (In this document, the term "/" and "," should be interpreted to indicate "and/or." For instance, the expression "A/B" may mean "A and/or B." Further, "A, B" may mean "A and/or B." Further, "A/B/C" may mean "at least one of A, B, and/or C." Also, "A/B/C" may mean "at least one of A, B, and/or C.")
- [0039] 추가적으로, 본 문서에서 "또는"은 "및/또는"으로 해석된다. 예를 들어, "A 또는 B"은, 1) "A" 만을 의미하고, 2) "B" 만을 의미하거나, 3) "A 및 B"를 의미할 수 있다. 달리 표현하면, 본 문서의 "또는"은 "추가적으로 또는 대체적으로(additionally or alternatively)"를 의미할 수 있다. (Further, in the document, the term "or" should be interpreted to indicate "and/or." For instance, the expression "A or B" may comprise 1) only A, 2) only B, and/or 3) both A and B. In other words, the term "or" in this document should be interpreted to indicate "additionally or alternatively.")
- [0040] 도 1을 참조하면, 비디오/영상 코딩 시스템은 제1 장치(소스 디바이스) 및 제2 장치(수신 디바이스)를 포함할 수 있다. 소스 디바이스는 인코딩된 비디오(video)/영상(image) 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 수신 디바이스로 전달할 수 있다.
- [0041] 상기 소스 디바이스는 비디오 소스, 인코딩 장치, 전송부를 포함할 수 있다. 상기 수신 디바이스는 수신부, 디코딩 장치 및 렌더러를 포함할 수 있다. 상기 인코딩 장치는 비디오/영상 인코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 비디오/영상 디코딩 장치라고 불릴 수 있다. 송신기는 인코딩 장치에 포함될 수 있다. 수신기는 디코딩 장치에 포함될 수 있다. 렌더러는 디스플레이부를 포함할 수도 있고, 디스플레이부는 별개의 디바이스 또는 외부 컴포넌트로 구성될 수도 있다.
- [0042] 비디오 소스는 비디오/영상의 캡처, 합성 또는 생성 과정 등을 통하여 비디오/영상을 획득할 수 있다. 비디오 소스는 비디오/영상 캡처 디바이스 및/또는 비디오/영상 생성 디바이스를 포함할 수 있다. 비디오/영상 캡처 디바이스는 예를 들어, 하나 이상의 카메라, 이전에 캡처된 비디오/영상을 포함하는 비디오/영상 아카이브 등을 포함할 수 있다. 비디오/영상 생성 디바이스는 예를 들어 컴퓨터, 태블릿 및 스마트폰 등을 포함할 수 있으며 (전자적으로) 비디오/영상을 생성할 수 있다. 예를 들어, 컴퓨터 등을 통하여 가상의 비디오/영상이 생성될 수 있으며, 이 경우 관련 데이터가 생성되는 과정으로 비디오/영상 캡처 과정이 같음될 수 있다.
- [0043] 인코딩 장치는 입력 비디오/영상을 인코딩할 수 있다. 인코딩 장치는 압축 및 코딩 효율을 위하여 예측, 변환, 양자화 등 일련의 절차를 수행할 수 있다. 인코딩된 데이터(인코딩된 비디오/영상 정보)는 비트스트림(bitstream) 형태로 출력될 수 있다.
- [0044] 전송부는 비트스트림 형태로 출력된 인코딩된 비디오/영상 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 수신 디바이스의 수신부로 전달할 수 있다. 디지털 저장 매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장 매체를 포함할 수 있다. 전송부는 미리 정해진 파일 포맷을 통하여 미디어 파일을 생성하기 위한 엘리먼트를 포함할 수 있고, 방송/통신 네트워크를 통한 전송을 위한 엘리먼트를 포함할 수 있다. 수신부는 상기 비트스트림을 수신/추출하여 디코딩 장치로 전달할 수 있다.

- [0045] 디코딩 장치는 인코딩 장치의 동작에 대응하는 역양자화, 역변환, 예측 등 일련의 절차를 수행하여 비디오/영상을 디코딩할 수 있다.
- [0046] 렌더러는 디코딩된 비디오/영상을 렌더링할 수 있다. 렌더링된 비디오/영상은 디스플레이부를 통하여 디스플레이될 수 있다.
- [0047] 도 2는 본 개시가 적용될 수 있는 비디오/영상 인코딩 장치의 구성을 개략적으로 설명하는 도면이다. 이하 비디오 인코딩 장치라 함은 영상 인코딩 장치를 포함할 수 있다.
- [0048] 도 2를 참조하면, 인코딩 장치(200)는 영상 분할부(image partitioner, 210), 예측부(predictor, 220), 레지듀얼 처리부(residual processor, 230), 엔트로피 인코딩부(entropy encoder, 240), 가산부(adder, 250), 필터링부(filter, 260) 및 메모리(memory, 270)를 포함하여 구성될 수 있다. 예측부(220)는 인터 예측부(221) 및 인트라 예측부(222)를 포함할 수 있다. 레지듀얼 처리부(230)는 변환부(transformer, 232), 양자화부(quantizer 233), 역양자화부(dequantizer 234), 역변환부(inverse transformer, 235)를 포함할 수 있다. 레지듀얼 처리부(230)는 감산부(subtractor, 231)를 더 포함할 수 있다. 가산부(250)는 복원부(reconstructor) 또는 복원 블록 생성부(reconstructged block generator)로 불릴 수 있다. 상술한 영상 분할부(210), 예측부(220), 레지듀얼 처리부(230), 엔트로피 인코딩부(240), 가산부(250) 및 필터링부(260)는 실시예에 따라 하나 이상의 하드웨어 컴포넌트(예를 들어 인코더 칩셋 또는 프로세서)에 의하여 구성될 수 있다. 또한 메모리(270)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(270)을 내/외부 컴포넌트로 더 포함할 수도 있다.
- [0049] 영상 분할부(210)는 인코딩 장치(200)에 입력된 입력 영상(또는, 픽처, 프레임)를 하나 이상의 처리 유닛(processing unit)으로 분할할 수 있다. 일 예로, 상기 처리 유닛은 코딩 유닛(coding unit, CU)이라고 불릴 수 있다. 이 경우 코딩 유닛은 코딩 트리 유닛(coding tree unit, CTU) 또는 최대 코딩 유닛(largest coding unit, LCU)으로부터 QTBT(T Quad-tree binary-tree ternary-tree) 구조에 따라 재귀적으로(recursively) 분할될 수 있다. 예를 들어, 하나의 코딩 유닛은 쿼드 트리 구조, 바이너리 트리 구조, 및/또는 터너리 구조를 기반으로 하위(deeper) 텡스의 복수의 코딩 유닛들로 분할될 수 있다. 이 경우 예를 들어 쿼드 트리 구조가 먼저 적용되고 바이너리 트리 구조 및/또는 터너리 구조가 나중에 적용될 수 있다. 또는 바이너리 트리 구조가 먼저 적용될 수도 있다. 더 이상 분할되지 않는 최종 코딩 유닛을 기반으로 본 개시에 따른 코딩 절차가 수행될 수 있다. 이 경우 영상 특성에 따른 코딩 효율 등을 기반으로, 최대 코딩 유닛이 바로 최종 코딩 유닛으로 사용될 수 있고, 또는 필요에 따라 코딩 유닛은 재귀적으로(recursively) 보다 하위 텡스의 코딩 유닛들로 분할되어 최적의 사이즈의 코딩 유닛이 최종 코딩 유닛으로 사용될 수 있다. 여기서 코딩 절차라 함은 후술하는 예측, 변환, 및 복원 등의 절차를 포함할 수 있다. 다른 예로, 상기 처리 유닛은 예측 유닛(PU: Prediction Unit) 또는 변환 유닛(TU: Transform Unit)을 더 포함할 수 있다. 이 경우 상기 예측 유닛 및 상기 변환 유닛은 각각 상술한 최종 코딩 유닛으로부터 분할 또는 파티셔닝될 수 있다. 상기 예측 유닛은 샘플 예측의 단위일 수 있고, 상기 변환 유닛은 변환 계수를 유도하는 단위 및/또는 변환 계수로부터 레지듀얼 신호(residual signal)를 유도하는 단위일 수 있다.
- [0050] 유닛은 경우에 따라서 블록(block) 또는 영역(area) 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, MxN 블록은 M개의 열과 N개의 행으로 이루어진 샘플들 또는 변환 계수(transform coefficient)들의 집합을 나타낼 수 있다. 샘플은 일반적으로 픽셀 또는 픽셀의 값을 나타낼 수 있으며, 휘도(luma) 성분의 픽셀/픽셀값만을 나타낼 수도 있고, 채도(chroma) 성분의 픽셀/픽셀 값을 나타낼 수도 있다. 샘플은 하나의 픽처(또는 영상)을 픽셀(pixel) 또는 펠(pel)에 대응하는 용어로서 사용될 수 있다.
- [0051] 인코딩 장치(200)는 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)를 감산하여 레지듀얼 신호(residual signal, 잔여 블록, 잔여 샘플 어레이)를 생성할 수 있고, 생성된 레지듀얼 신호는 변환부(232)로 전송된다. 이 경우 도시된 바와 같이 인코더(200) 내에서 입력 영상 신호(원본 블록, 원본 샘플 어레이)에서 예측 신호(예측 블록, 예측 샘플 어레이)를 감산하는 유닛은 감산부(231)라고 불릴 수 있다. 예측부는 처리 대상 블록(이하, 현재 블록이라 함)에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 현재 블록 또는 CU 단위로 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있다. 예측부는 각 예측모드에 대한 설명에서 후술하는 바와 같이 예측 모드 정보 등 예측에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 예측에 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

- [0052] 인트라 예측부(222)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 비방향성 모드는 예를 들어 DC 모드 및 플래너 모드(Planar 모드)를 포함할 수 있다. 방향성 모드는 예측 방향의 세밀한 정도에 따라 예를 들어 33개의 방향성 예측 모드 또는 65개의 방향성 예측 모드를 포함할 수 있다. 다만, 이는 예시로서 설정에 따라 그 이상 또는 그 이하의 개수의 방향성 예측 모드들이 사용될 수 있다. 인트라 예측부(222)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.
- [0053] 인터 예측부(221)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring block)을 포함할 수 있다. 상기 참조 블록을 포함하는 참조 픽처와 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일할 수도 있고, 다를 수도 있다. 상기 시간적 주변 블록은 동일 위치 참조 블록(collocated reference block), 동일 위치 CU(coICU) 등의 이름으로 불릴 수 있으며, 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일 위치 픽처(collocated picture, colPic)라고 불릴 수도 있다. 예를 들어, 인터 예측부(221)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출하기 위하여 어떤 후보가 사용되는지를 지시하는 정보를 생성할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 예를 들어 스킵 모드와 머지 모드의 경우에, 인터 예측부(221)는 주변 블록의 움직임 정보를 현재 블록의 움직임 정보로 이용할 수 있다. 스킵 모드의 경우, 머지 모드와 달리 레지듀얼 신호가 전송되지 않을 수 있다. 움직임 정보 예측(motion vector prediction, MVP) 모드의 경우, 주변 블록의 움직임 벡터를 움직임 벡터 예측자(motion vector predictor)로 이용하고, 움직임 벡터 차분(motion vector difference)을 시그널링함으로써 현재 블록의 움직임 벡터를 지시할 수 있다.
- [0054] 예측부(220)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 인트라 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고 또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 콘텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔레트 모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보를 기반으로 픽처 내 샘플 값을 시그널링할 수 있다.
- [0055] 상기 예측부 (인터 예측부(221) 및/또는 상기 인트라 예측부(222) 포함)를 통해 생성된 예측 신호는 복원 신호를 생성하기 위해 이용되거나 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 변환부(232)는 레지듀얼 신호에 변환 기법을 적용하여 변환 계수들(transform coefficients)를 생성할 수 있다. 예를 들어, 변환 기법은 DCT(Discrete Cosine Transform), DST(Discrete Sine Transform), KLT(Karhunen-Loeve Transform), GBT(Graph-Based Transform), 또는 CNT(Conditionally Non-linear Transform) 중 적어도 하나를 포함할 수 있다. 여기서, GBT는 픽셀 간의 관계 정보를 그래프로 표현한다고 할 때 이 그래프로부터 얻어진 변환을 의미한다. CNT는 이전에 복원된 모든 픽셀(all previously reconstructed pixel)를 이용하여 예측 신호를 생성하고 그에 기초하여 획득되는 변환을 의미한다. 또한, 변환 과정은 정사각형의 동일한 크기를 갖는 픽셀 블록에 적용될 수도 있고, 정사각형이 아닌 가변 크기의 블록에도 적용될 수 있다.
- [0056] 양자화부(233)는 변환 계수들을 양자화하여 엔트로피 인코딩부(240)로 전송되고, 엔트로피 인코딩부(240)는 양자화된 신호(양자화된 변환 계수들에 관한 정보)를 인코딩하여 비트스트림으로 출력할 수 있다. 상기 양자화된 변환 계수들에 관한 정보는 레지듀얼 정보라고 불릴 수 있다. 양자화부(233)는 계수 스캔 순서(scan order)를 기반으로 블록 형태의 양자화된 변환 계수들을 1차원 벡터 형태로 재정렬할 수 있고, 상기 1차원 벡터 형태의 양자화된 변환 계수들을 기반으로 상기 양자화된 변환 계수들에 관한 정보를 생성할 수도 있다. 엔트로피 인코

딩부(240)는 예를 들어 지수 골롬(exponential Golomb), CAVLC(context-adaptive variable length coding), CABAC(context-adaptive binary arithmetic coding) 등과 같은 다양한 인코딩 방법을 수행할 수 있다. 엔트로피 인코딩부(240)는 양자화된 변환 계수들 외 비디오/이미지 복원에 필요한 정보들(예컨대 선택 요소들(syntax elements)의 값 등)을 함께 또는 별도로 인코딩할 수도 있다. 인코딩된 정보(ex. 인코딩된 비디오/영상 정보)는 비트스트림 형태로 NAL(network abstraction layer) 유닛 단위로 전송 또는 저장될 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)을 더 포함할 수 있다. 본 문서에서 인코딩 장치에서 디코딩 장치로 전달/시그널링되는 정보 및/또는 선택 요소들은 비디오/영상 정보에 포함될 수 있다. 상기 비디오/영상 정보는 상술한 인코딩 절차를 통하여 인코딩되어 상기 비트스트림에 포함될 수 있다. 상기 비트스트림은 네트워크를 통하여 전송될 수 있고, 또는 디지털 저장매체에 저장될 수 있다. 여기서 네트워크는 방송망 및/또는 통신망 등을 포함할 수 있고, 디지털 저장매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장매체를 포함할 수 있다. 엔트로피 인코딩부(240)로부터 출력된 신호는 전송하는 전송부(미도시) 및/또는 저장하는 저장부(미도시)가 인코딩 장치(200)의 내/외부 엘리먼트로서 구성될 수 있고, 또는 전송부는 엔트로피 인코딩부(240)에 포함될 수도 있다.

[0057] 양자화부(233)로부터 출력된 양자화된 변환 계수들은 예측 신호를 생성하기 위해 이용될 수 있다. 예를 들어, 양자화된 변환 계수들에 역양자화부(234) 및 역변환부(235)를 통해 역양자화 및 역변환을 적용함으로써 레지듀얼 신호(레지듀얼 블록 or 레지듀얼 샘플들)를 복원할 수 있다. 가산부(155)는 복원된 레지듀얼 신호를 인터 예측부(221) 또는 인트라 예측부(222)로부터 출력된 예측 신호에 더함으로써 복원(reconstructed) 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)가 생성될 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다. 가산부(250)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.

[0058] 한편 픽처 인코딩 및/또는 복원 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.

[0059] 필터링부(260)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(260)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(270), 구체적으로 메모리(270)의 DPB에 저장할 수 있다. 상기 다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다. 필터링부(260)는 각 필터링 방법에 대한 설명에서 후술하는 바와 같이 필터링에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(240)로 전달할 수 있다. 필터링 관련 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

[0060] 메모리(270)에 전송된 수정된 복원 픽처는 인터 예측부(221)에서 참조 픽처로 사용될 수 있다. 인코딩 장치는 이를 통하여 인터 예측이 적용되는 경우, 인코딩 장치(100)와 디코딩 장치에서의 예측 미스매치를 피할 수 있고, 부호화 효율도 향상시킬 수 있다.

[0061] 메모리(270) DPB는 수정된 복원 픽처를 인터 예측부(221)에서의 참조 픽처로 사용하기 위해 저장할 수 있다. 메모리(270)는 현재 픽처 내 움직임 정보가 도출된(또는 인코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(221)에 전달할 수 있다. 메모리(270)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(222)에 전달할 수 있다.

[0062] 도 3은 본 개시가 적용될 수 있는 비디오/영상 디코딩 장치의 구성을 개략적으로 설명하는 도면이다.

[0063] 도 3을 참조하면, 디코딩 장치(300)는 엔트로피 디코딩부(entropy decoder, 310), 레지듀얼 처리부(residual processor, 320), 예측부(predictor, 330), 가산부(adder, 340), 필터링부(filter, 350) 및 메모리(memoery, 360)를 포함하여 구성될 수 있다. 예측부(330)는 인터 예측부(331) 및 인트라 예측부(332)를 포함할 수 있다. 레지듀얼 처리부(320)는 역양자화부(dequantizer, 321) 및 역변환부(inverse transformer, 321)를 포함할 수 있다. 상술한 엔트로피 디코딩부(310), 레지듀얼 처리부(320), 예측부(330), 가산부(340) 및 필터링부(350)는 실시예에 따라 하나의 하드웨어 컴포넌트(예를 들어 디코더 칩셋 또는 프로세서)에 의하여 구성될 수 있다. 또한 메모리(360)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(360)을 내/외부 컴포넌트로 더 포함할 수도 있다.

- [0064] 비디오/영상 정보를 포함하는 비트스트림이 입력되면, 디코딩 장치(300)는 도 2의 인코딩 장치에서 비디오/영상 정보가 처리된 프로세스에 대응하여 영상을 복원할 수 있다. 예를 들어, 디코딩 장치(300)는 상기 비트스트림으로부터 획득한 블록 분할 관련 정보를 기반으로 유닛들/블록들을 도출할 수 있다. 디코딩 장치(300)는 인코딩 장치에서 적용된 처리 유닛을 이용하여 디코딩을 수행할 수 있다. 따라서 디코딩의 처리 유닛은 예를 들어 코딩 유닛일 수 있고, 코딩 유닛은 코딩 트리 유닛 또는 최대 코딩 유닛으로부터 쿼드 트리 구조, 바이너리 트리 구조 및/또는 터너리 트리 구조를 따라서 분할될 수 있다. 코딩 유닛으로부터 하나 이상의 변환 유닛이 도출될 수 있다. 그리고, 디코딩 장치(300)를 통해 디코딩 및 출력된 복원 영상 신호는 재생 장치를 통해 재생될 수 있다.
- [0065] 디코딩 장치(300)는 도 2의 인코딩 장치로부터 출력된 신호를 비트스트림 형태로 수신할 수 있고, 수신된 신호는 엔트로피 디코딩부(310)를 통해 디코딩될 수 있다. 예를 들어, 엔트로피 디코딩부(310)는 상기 비트스트림을 파싱하여 영상 복원(또는 픽처 복원)에 필요한 정보(ex. 비디오/영상 정보)를 도출할 수 있다. 상기 비디오/영상 정보는 어댑테이션 파라미터 세트(APS), 픽처 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 또는 비디오 파라미터 세트(VPS) 등 다양한 파라미터 세트에 관한 정보를 더 포함할 수 있다. 또한 상기 비디오/영상 정보는 일반 제한 정보(general constraint information)을 더 포함할 수 있다. 디코딩 장치는 상기 파라미터 세트에 관한 정보 및/또는 상기 일반 제한 정보를 더 기반으로 픽처를 디코딩할 수 있다. 본 문서에서 후술되는 시그널링/수신되는 정보 및/또는 신택스 요소들은 상기 디코딩 절차를 통하여 디코딩되어 상기 비트스트림으로부터 획득될 수 있다. 예컨대, 엔트로피 디코딩부(310)는 지수 곱셈 부호화, CAVLC 또는 CABAC 등의 코딩 방법을 기초로 비트스트림 내 정보를 디코딩하고, 영상 복원에 필요한 신택스 엘리먼트의 값, 레지듀얼에 관한 변환 계수의 양자화된 값 등을 출력할 수 있다. 보다 상세하게, CABAC 엔트로피 디코딩 방법은, 비트스트림에서 각 신택스 요소에 해당하는 빈을 수신하고, 디코딩 대상 신택스 요소 정보와 주변 및 디코딩 대상 블록의 디코딩 정보 혹은 이전 단계에서 디코딩된 심볼/빈의 정보를 이용하여 문맥(context) 모델을 결정하고, 결정된 문맥 모델에 따라 빈(bin)의 발생 확률을 예측하여 빈의 산술 디코딩(arithmetic decoding)을 수행하여 각 신택스 요소의 값에 해당하는 심볼을 생성할 수 있다. 이때, CABAC 엔트로피 디코딩 방법은 문맥 모델 결정 후 다음 심볼/빈의 문맥 모델을 위해 디코딩된 심볼/빈의 정보를 이용하여 문맥 모델을 업데이트할 수 있다. 엔트로피 디코딩부(310)에서 디코딩된 정보 중 예측에 관한 정보는 예측부(인tra 예측부(332) 및 inter 예측부(331))로 제공되고, 엔트로피 디코딩부(310)에서 엔트로피 디코딩이 수행된 레지듀얼 값, 즉 양자화된 변환 계수들 및 관련 파라미터 정보는 레지듀얼 처리부(320)로 입력될 수 있다. 레지듀얼 처리부(320)는 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플들, 레지듀얼 샘플 어레이)를 도출할 수 있다. 또한, 엔트로피 디코딩부(310)에서 디코딩된 정보 중 필터링에 관한 정보는 필터링부(350)로 제공될 수 있다. 한편, 인코딩 장치로부터 출력된 신호를 수신하는 수신부(미도시)가 디코딩 장치(300)의 내/외부 엘리먼트로서 더 구성될 수 있고, 또는 수신부는 엔트로피 디코딩부(310)의 구성요소일 수도 있다. 한편, 본 문서에 따른 디코딩 장치는 비디오/영상/픽처 디코딩 장치라고 불릴 수 있고, 상기 디코딩 장치는 정보 디코더(비디오/영상/픽처 정보 디코더) 및 샘플 디코더(비디오/영상/픽처 샘플 디코더)로 구분할 수도 있다. 상기 정보 디코더는 상기 엔트로피 디코딩부(310)를 포함할 수 있고, 상기 샘플 디코더는 상기 역양자화부(321), 역변환부(322), 가산부(340), 필터링부(350), 메모리(360), intra 예측부(332) 및 inter 예측부(331) 중 적어도 하나를 포함할 수 있다.
- [0066] 역양자화부(321)에서는 양자화된 변환 계수들을 역양자화하여 변환 계수들을 출력할 수 있다. 역양자화부(321)는 양자화된 변환 계수들을 2차원의 블록 형태로 재정렬할 수 있다. 이 경우 상기 재정렬은 인코딩 장치에서 수행된 계수 스캔 순서를 기반으로 재정렬을 수행할 수 있다. 역양자화부(321)는 양자화 파라미터(예를 들어 양자화 스텝 사이즈 정보)를 이용하여 양자화된 변환 계수들에 대한 역양자화를 수행하고, 변환 계수들(transform coefficient)을 획득할 수 있다.
- [0067] 역변환부(322)에서는 변환 계수들을 역변환하여 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플 어레이)를 획득하게 된다.
- [0068] 예측부는 현재 블록에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 엔트로피 디코딩부(310)로부터 출력된 상기 예측에 관한 정보를 기반으로 상기 현재 블록에 intra 예측이 적용되는지 또는 inter 예측이 적용되는지 결정할 수 있고, 구체적인 intra/inter 예측 모드를 결정할 수 있다.
- [0069] 예측부(320)는 후술하는 다양한 예측 방법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 하나의 블록에 대한 예측을 위하여 intra 예측 또는 inter 예측을 적용할 수 있을 뿐 아니라, intra 예측과 inter 예측을 동시에 적용할 수 있다. 이는 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부는 블록에 대한 예측을 위하여 intra 블록 카피(intra block copy, IBC) 예측 모드에 기반할 수도 있고

또는 팔레트 모드(palette mode)에 기반할 수도 있다. 상기 IBC 예측 모드 또는 팔레트 모드는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 콘텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 기본적으로 현재 픽처 내에서 예측을 수행하나 현재 픽처 내에서 참조 블록을 도출하는 점에서 인터 예측과 유사하게 수행될 수 있다. 즉, IBC는 본 문서에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다. 팔레트 모드는 인트라 코딩 또는 인트라 예측의 일 예로 볼 수 있다. 팔레트 모드가 적용되는 경우 팔레트 테이블 및 팔레트 인덱스에 관한 정보가 상기 비디오/영상 정보에 포함되어 시그널링될 수 있다.

- [0070] 인트라 예측부(331)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 예측 모드에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 예측에서 예측 모드들은 복수의 비방향성 모드와 복수의 방향성 모드를 포함할 수 있다. 인트라 예측부(331)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.
- [0071] 인터 예측부(332)는 참조 픽처 상에서 움직임 벡터에 의해 특정되는 참조 블록(참조 샘플 어레이)을 기반으로, 현재 블록에 대한 예측된 블록을 유도할 수 있다. 이때, 인터 예측 모드에서 전송되는 움직임 정보의 양을 줄이기 위해 주변 블록과 현재 블록 간의 움직임 정보의 상관성에 기초하여 움직임 정보를 블록, 서브블록 또는 샘플 단위로 예측할 수 있다. 상기 움직임 정보는 움직임 벡터 및 참조 픽처 인덱스를 포함할 수 있다. 상기 움직임 정보는 인터 예측 방향(L0 예측, L1 예측, Bi 예측 등) 정보를 더 포함할 수 있다. 인터 예측의 경우에, 주변 블록은 현재 픽처 내에 존재하는 공간적 주변 블록(spatial neighboring block)과 참조 픽처에 존재하는 시간적 주변 블록(temporal neighboring block)을 포함할 수 있다. 예를 들어, 인터 예측부(332)는 주변 블록들을 기반으로 움직임 정보 후보 리스트를 구성하고, 수신한 후보 선택 정보를 기반으로 상기 현재 블록의 움직임 벡터 및/또는 참조 픽처 인덱스를 도출할 수 있다. 다양한 예측 모드를 기반으로 인터 예측이 수행될 수 있으며, 상기 예측에 관한 정보는 상기 현재 블록에 대한 인터 예측의 모드를 지시하는 정보를 포함할 수 있다.
- [0072] 가산부(340)는 획득된 레지듀얼 신호를 예측부(인터 예측부(332) 및/또는 인트라 예측부(331) 포함)로부터 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)에 더함으로써 복원 신호(복원 픽처, 복원 블록, 복원 샘플 어레이)를 생성할 수 있다. 스킵 모드가 적용된 경우와 같이 처리 대상 블록에 대한 레지듀얼이 없는 경우, 예측된 블록이 복원 블록으로 사용될 수 있다.
- [0073] 가산부(340)는 복원부 또는 복원 블록 생성부라고 불릴 수 있다. 생성된 복원 신호는 현재 픽처 내 다음 처리 대상 블록의 인트라 예측을 위하여 사용될 수 있고, 후술하는 바와 같이 필터링을 거쳐서 출력될 수도 있고 또는 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.
- [0074] 한편, 픽처 디코딩 과정에서 LMCS (luma mapping with chroma scaling)가 적용될 수도 있다.
- [0075] 필터링부(350)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(350)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(360), 구체적으로 메모리(360)의 DPB에 전송할 수 있다. 상기 다양한 필터링 방법은 예를 들어, 디블록킹 필터링, 샘플 적응적 오프셋(sample adaptive offset), 적응적 루프 필터(adaptive loop filter), 양방향 필터(bilateral filter) 등을 포함할 수 있다.
- [0076] 메모리(360)의 DPB에 저장된 (수정된) 복원 픽처는 인터 예측부(332)에서 참조 픽처로 사용될 수 있다. 메모리(360)는 현재 픽처 내 움직임 정보가 도출된(또는 디코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(260)에 전달할 수 있다. 메모리(360)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(331)에 전달할 수 있다.
- [0077] 본 명세서에서, 인코딩 장치(100)의 필터링부(260), 인터 예측부(221) 및 인트라 예측부(222)에서 설명된 실시 예들은 각각 디코딩 장치(300)의 필터링부(350), 인터 예측부(332) 및 인트라 예측부(331)에도 동일 또는 대응되도록 적용될 수 있다.
- [0078] 상술한 바와 같이 비디오 코딩을 수행함에 있어 압축 효율을 높이기 위하여 예측을 수행한다. 이를 통하여 코딩 대상 블록인 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록을 생성할 수 있다. 여기서 상기 예측된 블록은 공간 도메인(또는 픽셀 도메인)에서의 예측 샘플들을 포함한다. 상기 예측된 블록은 인코딩 장치 및 디코딩 장치에서 동일하게 도출되며, 상기 인코딩 장치는 원본 블록의 원본 샘플 값 자체가 아닌 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼에 대한 정보(레지듀얼 정보)를 디코딩 장치로 시그널링함으로써 영상 코딩 효율을 높일 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 레지듀얼 샘플들을 포함하는 레지듀얼 블록

을 도출하고, 상기 레지듀얼 블록과 상기 예측된 블록을 합하여 복원 샘플들을 포함하는 복원 블록을 생성할 수 있고, 복원 블록들을 포함하는 복원 픽처를 생성할 수 있다.

[0079] 상기 레지듀얼 정보는 변환 및 양자화 절차를 통하여 생성될 수 있다. 예를 들어, 인코딩 장치는 상기 원본 블록과 상기 예측된 블록 간의 레지듀얼 블록을 도출하고, 상기 레지듀얼 블록에 포함된 레지듀얼 샘플들(레지듀얼 샘플 어레이)에 변환 절차를 수행하여 변환 계수들을 도출하고, 상기 변환 계수들에 양자화 절차를 수행하여 양자화된 변환 계수들을 도출하여 관련된 레지듀얼 정보를 (비트스트림을 통하여) 디코딩 장치로 시그널링할 수 있다. 여기서 상기 레지듀얼 정보는 상기 양자화된 변환 계수들의 값 정보, 위치 정보, 변환 기법, 변환 커널, 양자화 파라미터 등의 정보를 포함할 수 있다. 디코딩 장치는 상기 레지듀얼 정보를 기반으로 역양자화/역변환 절차를 수행하고 레지듀얼 샘플들(또는 레지듀얼 블록)을 도출할 수 있다. 디코딩 장치는 예측된 블록과 상기 레지듀얼 블록을 기반으로 복원 픽처를 생성할 수 있다. 인코딩 장치는 또한 이후 픽처의 인터 예측을 위한 참조를 위하여 양자화된 변환 계수들을 역양자화/역변환하여 레지듀얼 블록을 도출하고, 이를 기반으로 복원 픽처를 생성할 수 있다.

[0080] 도 4는 일 실시예에 따른 고주파 제로잉을 설명하기 위한 도면이다.

[0081] 본 명세서의 "고주파 제로잉(highfrequency zeroing)"은 제1가로크기(또는 길이) W_1 , 제1세로크기(또는 길이) H_1 을 갖는 (변환) 블록에서 일정 값 이상으로 높은 주파수와 관련된 변환 계수들을 제로화(즉, 0으로 결정)하는 과정을 의미한다. 고주파 제로잉이 적용되는 경우, 상기 (변환) 블록 내 변환 계수들 중 제2가로크기 W_2 및 제2세로크기 H_2 를 기반으로 구성되는 저주파 변환 계수 영역의 외부의 변환 계수들의 변환 계수 값은 모두 0으로 결정(설정)될 수 있다. 상기 저주파 변환 계수 영역의 외부는 고주파 변환 계수 영역으로 지칭될 수 있다. 일 예시에서, 상기 저주파 변환 계수 영역은 상기 (변환) 블록의 좌상단에서부터 위치하는 사각형 모양의 영역일 수 있다.

[0082] 본 명세서에서는 특정 정보 또는 개념을 정의하기 위한 특정 용어 또는 문장을 사용하고 있다. 예를 들어, 상술한 바와 같이 본 명세서에서는 제1가로크기(또는 길이) W_1 , 제1세로크기(또는 길이) H_1 을 갖는 (변환) 블록에서 일정 값 이상으로 높은 주파수와 관련된 변환 계수들을 제로화하는 과정을 "고주파 제로잉"이라고 정의하고 있고, 상기 고주파 제로잉을 통해 제로잉이 수행된 영역을 "고주파 변환 계수 영역"으로, 상기 제로잉이 수행되지 않은 영역을 "저주파 변환 계수 영역"으로 정의하고 있다. 상기 저주파 변환 계수 영역의 사이즈를 나타내기 위해 제2가로크기(또는 길이) W_2 와 제2세로크기(또는 길이) H_2 가 사용되고 있다.

[0083] 그러나, "고주파 제로잉"은 highfrequency zeroing, 하이프리퀀시 제로잉, 고주파 제로화 등 다양한 용어로 대체될 수 있고, 상기 "고주파 변환 계수 영역"은 고주파 제로잉 적용 영역, 하이프리퀀시 영역, 고주파 계수 영역 등 다양한 용어로 대체될 수 있으며, 상기 "저주파 변환 계수 영역"은 고주파 제로잉 미적용 영역, 로우프리퀀시 영역, 저주파 계수 영역, 제한 영역 등 다양한 용어로 대체될 수 있는 바, 본 명세서에서 특정 정보 또는 개념을 정의하기 위해 사용된 특정 용어 또는 문장을 명세서 전반에서 해석함에 있어서 그 명칭에 국한된 해석을 하여서는 안 되고, 상기 용어가 나타내고자 하는 내용에 따른 다양한 동작, 기능 및 효과에 주목하여 해석할 필요가 있다.

[0084] 일 실시예에서는, 고주파 제로잉이 적용되는 (변환) 블록(TB, TU 또는 CB)에 대한 선택스 요소 `last_sig_coeff_x_prefix` 및 `last_sig_coeff_y_prefix`의 이진화를 수행하는 방법이 제안될 수 있다. `last_sig_coeff_x_prefix`와 `last_sig_coeff_y_prefix`는 트런케이티드 라이스 코드(truncated Rice code)로 이진화 될 수 있고, 이 때 `cRiceParam`의 값은 0을 사용할 수 있다. 트런케이티드 라이스 코드에 대한 이진화에 사용되는 `cMax`의 값은, `last_sig_coeff_x_prefix`의 이진화를 수행할 시에는 수학적 식 1에 기반하고, `last_sig_coeff_y_prefix`의 이진화를 수행할 때는 수학적 식 2에 기반하여 결정될 수 있다.

[0085] [수학적 식 1]

[0086]
$$cMax = (\log_2 W_1 \ll 1) - 1$$

[0087] [수학적 식 2]

[0088]
$$cMax = (\log_2 H_1 \ll 1) - 1$$

[0089] 여기서, W_1 은 (변환) 블록의 가로 길이(또는 폭), H_1 은 변환 블록의 세로 길이(또는 높이)를 나타낼 수 있다. 예를 들어, 도 6과 같은 64x32 변환 블록의 경우 W_1 은 64, H_1 은 32이다. 따라서, `last_sig_coeff_x_prefix`의

이진화를 위한 cMax의 값은 11, last_sig_coeff_y_prefix의 이진화를 위한 cMax의 값은 9가 될 수 있다.

[0090] 아래의 표 1은 W1 또는 H1이 32인 경우의 이진화를, 아래의 표 2는 W1 또는 H1이 64인 경우의 이진화 코드워드를 나타낸다. 일 실시예에서는 변환 블록의 크기를 기반으로 트런케이티드 라이스 코드(truncated Rice code) 이진화를 수행할 수 있으므로, 아래의 표 2와 같이 LastSignificantCoeffX 또는 LastSignificantCoeffY의 값이 32 내지 47인 경우를 부호화하기 위한 last_sig_coeff_x_prefix 또는 last_sig_coeff_y_prefix의 코드워드는 '1111111110'이 될 수 있고, 48~63인 경우를 부호화하기 위한 코드워드는 '1111111111'이 될 수 있다, 두 가지 경우 모두 11개의 빈을 기반으로 이진화 될 수 있다. 상기 코드워드는 빈 스트링이라고 불릴 수 있다.

표 1

LastSignificantCoeffXorLastSignificantCoeffY	codeword	length of the codeword
0	0	1
1	10	2
2	110	3
3	1110	4
4, 5	11110	5
6, 7	111110	6
8~11	1111110	7
12~15	11111110	8
16~23	111111110	9
24~31	111111111	9

표 2

LastSignificantCoeffXorLastSignificantCoeffY	codeword	length of the codeword
0	0	1
1	10	2
2	110	3
3	1110	4
4, 5	11110	5
6, 7	111110	6
8~11	1111110	7
12~15	11111110	8
16~23	111111110	9
24~31	1111111110	10
32~47	11111111110	11
48~63	11111111111	11

[0093]한편, 일 실시예에 따른 인코딩 장치(엔트로피 인코딩부)는, (양자화된) 변환 계수들에 대한 레지듀얼 코딩 절차를 수행할 수 있다. 일 예시에서, 인코딩 장치는 현재 블록(현재 코딩 블록(CB) 또는 현재 변환 블록(TB)) 내 (양자화된) 변환 계수들을 스캔 순서에 따라 레지듀얼 코딩할 수 있다. 이때, 인코딩 장치는 예를 들어 아래의 표 3에 기재된 바와 같은 레지듀얼 정보에 관한 다양한 선택스 요소들을 생성 및 인코딩할 수 있다.

표 3

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if(transform_skip_enabled_flag && (cIdx != 0 cu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
transform_skip_flag[x0][y0][cIdx]	ae(v)
last_sig_coeff_x_prefix	ae(v)
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)

if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
numSbCoeff = 1 << (log2SbSize << 1)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) -1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][1]	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
for(n = (i == lastSubBlock) ? lastScanPos - 1 : numSbCoeff -1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
}	
if(sig_coeff_flag[xC][yC]) {	
par_level_flag[n]	ae(v)
rem_abs_gt1_flag[n]	ae(v)
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag [n] + 2 * rem_abs_gt1_flag[n]	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][par_level_flag[n]]	
}	
for(n = numSbCoeff - 1; n >= 0; n--) {	
if(rem_abs_gt1_flag[n])	
rem_abs_gt2_flag[n]	ae(v)

} for(n = numSbCoeff - 1; n >= 0; n--) { xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0] yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1] if(rem_abs_gt2_flag[n]) abs_remainder[n] AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * (rem_abs_gt2_flag[n] + abs_remainder[n]) } if(dep_quant_enabled_flag !sign_data_hiding_enabled_flag) signHidden = 0 else signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0) for(n = numSbCoeff - 1; n >= 0; n--) { xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0] yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1] if(sig_coeff_flag[xC][yC] && (!signHidden (n != firstSigScanPosSb))) coeff_sign_flag[n] } if(dep_quant_enabled_flag) { QState = startQStateSb for(n = numSbCoeff - 1; n >= 0; n--) { xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0] yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1] if(sig_coeff_flag[xC][yC]) TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 1 : 0)) * (1 - 2 * coeff_sign_flag[n]) QState = QStateTransTable[QState][par_level_flag[n]] } else { sumAbsLevel = 0 for(n = numSbCoeff - 1; n >= 0; n--) xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0] yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1] if(sig_coeff_flag[xC][yC]) { TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n]) if(signHidden) { sumAbsLevel += AbsLevel[xC][yC] if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)) TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC] } } } } } if(cu_mts_flag[x0][y0] && (cIdx == 0) && !transform_skip_flag[x0][y0][cIdx] && ((CuPredMode[x0][y0] == MODE_INTRA && numSigCoeff > 2) (CuPredMode[x0][y0] == MODE_INTER))) { mts_idx[x0][y0] } }	
	ae(v)
	ae(v)

}	
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if(transform_skip_enabled_flag && (cIdx != 0 cu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
transform_skip_flag[x0][y0][cIdx]	ae(v)
last_sig_coeff_x_prefix	ae(v)
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
numSbCoeff = 1 << (log2SbSize << 1)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) -1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][1]	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight -log2SbSize] [lastSubBlock][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
for(n = (i == lastSubBlock) ? lastScanPos - 1 : numSbCoeff -1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
}	
if(sig_coeff_flag[xC][yC]) {	
par_level_flag[n]	ae(v)
rem_abs_gt1_flag[n]	ae(v)
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	

firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + 2 * rem_abs_gt1_flag[n]	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][par_level_flag[n]]	
}	
for(n = numSbCoeff - 1; n >= 0; n--) {	
if(rem_abs_gt1_flag[n])	
rem_abs_gt2_flag[n]	ae(v)
}	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(rem_abs_gt2_flag[n])	
abs_remainder[n]	
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * (rem_abs_gt2_flag[n] + abs_remainder[n])	
}	
if(dep_quant_enabled_flag !sign_data_hiding_enabled_flag)	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC] && (!signHidden (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ae(v)
}	
if(dep_quant_enabled_flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n---- xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC])	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff_sign_flag[n])	
QState = QStateTransTable[QState][par_level_flag[n]]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n---	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC]) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1))	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	

}	
}	
}	
}	
if(cu_mts_flag[x0][y0] && (cIdx == 0) && !transform_skip_flag[x0][y0][cIdx] && ((CuPredMode[x0][y0] = = MODE_INTRA && numSigCoeff > 2) (CuPredMode[x0][y0] = = MODE_INTER))) {	
mts_idx[x0][y0]	ae(v)
}	

[0095] 상기 표 3에서 나타난 바와 같이, last_sig_coeff_x_prefix의 값이 3보다 큰 경우 last_sig_coeff_x_suffix가 더 시그널링될 수 있으며, 상기 last_sig_coeff_x_suffix의 값을 기반으로 LastSignificantCoeffX가 도출될 수 있다. 예를 들어, LastSignificantCoeffX의 값이 32~47인 경우를 부호화하기 위한 last_sig_coeff_x_prefix의 코드워드는 '1111111110'이 될 수 있고, 32~47 중 어떤 값이 사용되는지 여부는 상기 last_sig_coeff_x_suffix의 값을 기반으로 결정될 수 있다. 또한, 상기 표 1에서 나타난 바와 같이, last_sig_coeff_y_prefix의 값이 3보다 큰 경우 last_sig_coeff_y_prefix가 더 시그널링될 수 있으며, 상기 last_sig_coeff_y_suffix의 값을 기반으로 LastSignificantCoeffY가 도출될 수 있다. 예를 들어, LastSignificantCoeffY의 값이 32~47인 경우를 부호화하기 위한 last_sig_coeff_x_prefix의 코드워드는 '1111111110'이고, 32~47 중 어떤 값이 사용되는지 여부는 상기 last_sig_coeff_y_suffix의 값을 기반으로 결정될 수 있다.

[0096] LastSignificantCoeffX 또는 LastSignificantCoeffY를 도출하기 위한 구체적인 계산은 예를 들어 아래의 표 4와 같이 수행될 수 있다.

표 4

[0097] The column position of the last significant coefficient in scanning order within a transform block LastSignificantCoeffX is derived as follows:- If last_sig_coeff_x_suffix is not present, the following applies:LastSignificantCoeffX = last_sig_coeff_x_prefix- Otherwise (last_sig_coeff_x_suffix is present), the following applies:LastSignificantCoeffX=(1<<((last_sig_coeff_x_prefix>>1)-1)) * (2+(last_sig_coeff_x_prefix&1)) + last_sig_coeff_x_suffix
The row position of the last significant coefficient in scanning order within a transform block LastSignificantCoeffY is derived as follows:- If last_sig_coeff_y_suffix is not present, the following applies:LastSignificantCoeffY = last_sigcoeff_y_prefix- Otherwise (last_sig_coeff_y_suffix is present), the following applies:LastSignificantCoeffY=(1<<((last_sig_coeff_y_prefix>>1)-1))*(2+(last_sig_coeff_y_prefix&1))+last_sig_coeff_y_suffix

[0098] 고주파 제로잉은 제1가로크기(W1) 또는 제1세로크기(H1)을 갖는 변환 블록에서 일정 이상으로 높은 주파수의 계수들을 제로화 해(즉, 0으로 결정하여) 잔여 변환 계수를 제2가로크기(W2) 또는 제2세로크기(H2)로 제한하는 것을 의미한다. 이때 일 실시예에서는, 변환 블록의 크기(제1가로크기 또는 제1세로크기)에 기반한 트런케이티드 라이스 코드를 기반으로 이진화 하기 보다는, 고주파 제로잉을 통해 도출되는 제한 영역의 크기(제2가로크기 또는 제2세로크기)에 기반한 트런케이티드 라이스 코드를 기반으로 이진화를 수행하는 방안을 고려해 볼 수 있다. 제2가로크기와 제2세로크기를 이용하여 last_sig_coeff_x_prefix를 위한 cMax와 last_sig_coeff_y_prefix를 위한 cMax를 아래의 수학적 식 3 및 수학적 식 4로 각각 정의한 후, 트런케이티드 라이스 코드를 생성할 수 있다.

[0099] [수학적 식 3]

[0100] $cMax = (\log_2(\min(W_1, W_2)) \ll 1) - 1$

[0101] [수학적 식 4]

[0102] $cMax = (\log_2(\min(H_1, H_2)) \ll 1) - 1$

[0103] 일 예시에서, 제1가로크기 또는 제1세로크기가 64이고, 제2가로크기 또는 제2세로크기가 32일 때, 상기 수학적 식 3 또는 수학적 식 4를 기반으로 도출되는 트런케이티드 라이스 코드는 아래의 표 5와 같을 수 있다. 고주파 제로잉을 통해, 상기 제2가로크기 또는 상기 제2세로크기의 밖에서 형성되는 고주파 변환 계수 영역 내 고주파 계수들

에서 잔여 변환 계수가 없으므로, 하기의 표 6과 같은 이진화 코드워드 설계가 가능할 수 있다.

[0104] 일 실시예에서, W2와 H2는 고정된 값으로 설정될 수 있다. 또는 W2 및 H2는 W1 및 H1을 기반으로 결정될 수 있다. 또는 W2 및 H2를 나타내는 정보가 인코딩 장치에서 디코딩 장치로 시그널링될 수 있다. 일 예시에서, 상기 W2와 H2는 각각 32 또는 16으로 설정될 수 있다. 다른 예시에서, 상기 W2와 H2는 각각 W1의 1/2, H1의 1/2로 도출될 수 있다. 또 다른 예시에서, 상기 W2와 H2는 max(W1,H1)의 1/2로 도출될 수도 있다. 다만, 이는 예시로서 상기 W2와 H2는 인코딩 장치 및 디코딩 장치에서 설정된 또 다른 다양한 방법에 의하여 결정될 수도 있다. 제안하는 방법을 통해 LastSignificantCoeffX 또는 LastSignificantCoeffY의 일부 값에 대한 코드워드의 길이를 효과적으로 줄일 수 있다. 또한, 이를 통해 절감되는 부호화 빈(bin)이 문맥 부호화 빈이므로 처리량 (throughput) 면에서 이점을 가질 수 있다.

표 5

LastSignificantCoeffXorLastSignificantCoeffY	codeword	length of the codeword
0	0	1
1	10	2
2	110	3
3	1110	4
4, 5	11110	5
6, 7	111110	6
8~11	1111110	7
12~15	11111110	8
16~23	111111110	9
24~31	111111111	9
32~47	N/A	N/A
48~63	N/A	N/A

[0106] 일 예시에서, 도 4에서 설명한 실시예들을 기반으로 도 7a 내지 도 8b에서 후술할 레지듀얼 코딩 방법이 수행될 수 있다. 다른 일 예시에서, 도 4에서 설명한 실시예들은 기반으로 도 9에서 후술할 인코딩 방법 또는 도 11에서 후술할 디코딩 방법이 수행될 수 있다.

[0107] 도 5는 고주파 제로잉이 적용된 현재 블록에 대한 변환 계수들을 스캔하는 방법의 일 예시를 설명하기 위한 도면이고, 도 6은 고주파 제로잉이 적용된 현재 블록에 대한 변환 계수들을 스캔하는 방법의 다른 일 예시를 설명하기 위한 도면이다.

[0108] 일 실시예에서는, 마지막 0이 아닌 계수가 포함된 마지막 변환 계수 그룹의 위치로부터 역 대각 방향으로 변환 계수 그룹을 스캔하면서 레지듀얼 코딩을 수행할 수 있다. 반면, 이에 추가하여 또는 이를 대체하여 (additionally or alternatively), 다른 일 실시예에서는 고주파 제로잉이 적용되는 변환 블록에 대해 변환 계수 그룹의 스캔 순서를 변경할 수 있다.

[0109] 도 5는 고주파 제로잉이 적용된 64x64 변환 블록에서 4x4 변환 계수 그룹이 스캔되는 방법을 나타내고 있다. 도 5에서, L로 표시된 부분이 마지막 0이 아닌 계수가 포함된 4x4 변환 계수 그룹을 의미한다. 고주파 제로잉은 제 1가로크기(W1) 및/또는 제 1세로크기(H1)를 갖는 변환 블록에서 일정 주파수 이상의 고주파와 관련된 변환 계수들을 제로화하여(또는 제로아웃하여) 잔여 변환 계수를 제2가로크기(W2) 및/또는 제2세로크기(H2)로 제한하는 것을 의미하며, 도 5에서 도트(dot) 표시된 4x4 블록들은 고주파 제로잉을 통해 제로아웃 되는 영역을 나타낸다. 도 5에서 상기 빗금친 영역은 저주파 변환 계수 영역으로, 상기 도트 표시된 영역은 고주파 변환 계수 영역으로 지칭될 수 있다.

[0110] W2 및/또는 H2는 고정된 값으로 설정되거나, W1 및/또는 H1을 기반으로 결정될 수 있다. 또는, W2 및/또는 H2를 나타내는 정보가 인코딩 장치에서 디코딩 장치로 시그널링될 수 있다. 일 예시에서, 상기 W2, H2는 각각 32 또는 16으로 설정될 수 있다. 다른 일 예시에서, 상기 W2 및/또는 H2는 각각 W1의 1/2 및/또는 H1의 1/2로 도출될 수 있다. 또 다른 일 예시에서, 상기 W2 및/또는 H2는 max(W1,H1)의 1/2로 도출될 수도 있다. 다만, 상기된 예시들은 일부에 불과하며, 상기 W2 및/또는 H2는 인코딩 장치 및 디코딩 장치에서 다양한 방법을 기반으로 결정될 수도 있다.

- [0111] 도 5에 따른 변환 계수 그룹 스캔 방법을 사용하면, 고주파 제로잉을 통해 제로아웃 되는 영역 또한 역 대각선 스캔 순서에 따라 스캔되어야 함을 확인할 수 있다. 이와 같이 불필요한 블록을 스캔하는 경우, 코딩 복잡도 (complexity)가 상승할 수 있고, 제로아웃 영역(도 5에 x 표시된 4x4 블록들이 구성하는 영역)에서 매번 코디드 서브 블록 플래그(상기 현재 블록 내 서브 블록에 대한 변환 계수들의 변환 계수 레벨이 모두 0인지 여부를 나타내는 플래그로서, 예를 들어 coded_sub_block_flag로 표현될 수 있다)가 0임을 부호화해야 하는 부하(burden)가 발생할 수 있다.
- [0112] 따라서, 고주파 제로잉이 적용되는 경우, 변환 계수 그룹의 스캔을 고주파 제로잉에서 제한되는 영역(또는, 저주파 변환 계수 영역)의 사이즈 (제2가로크기 또는 제2세로크기) 내에서만 수행하는 것으로 제한하는 방법이 고려될 수 있다. 일 실시예에서는, 도 5에 따른 변환 계수 그룹 스캔 대신, 도 6과 같이 변환 계수 그룹 스캔을 수행하는 방법을 제안한다.
- [0113] 도 6에 따른 변환 계수 그룹 스캔 방법을 수행하는 경우, 불필요한 스캔 과정을 생략할 수 있고, 코디드 서브 블록 플래그가 불필요하게 부호화되는 것을 방지할 수 있다. 즉, 도 6에 따른 일 실시예에 의하면, 인코딩 장치는 현재 (변환) 블록 내에서 W2 및/또는 H2를 초과하는 영역(즉, 고주파 변환 계수 영역)에 위치하는 서브 블록에 대한 코디드 서브 블록 플래그를 레지듀얼 코딩 선택스(또는 비트스트림)에 포함하지 않을 수 있다. 즉, W2 및/또는 H2를 초과하는 영역에 위치하는 서브 블록에 대한 선택스 요소 coded_sub_block_flag에 관한 비트가 할당되지 않을 수 있다. 이러한 경우, 디코딩 장치는 W2 및/또는 H2를 초과하는 영역에 위치하는 서브 블록에 대하여는 스캔을 수행하지 않고도 상기 W2 및/또는 H2를 초과하는 영역에는 마지막 유효 계수가 위치하지 않음을 추론할 수 있다. 또한, 디코딩 장치는 W2 및/또는 H2를 초과하는 영역에 위치하는 서브 블록에 대한 선택스 요소 coded_sub_block_flag를 레지듀얼 코딩 선택스(또는 비트스트림)로부터 파싱(parsing)을 하지 않고, 해당 선택스 요소 coded_sub_block_flag의 값을 0으로 추론할 수 있다.
- [0114] 따라서, 도 6에 따른 일 실시예에서는, 도 5에 따른 일 실시예와 비교할 때, 상기 64x64 현재 블록에 포함된 총 49개의 4x4 서브 블록들에 대하여, 'coded_sub_block_flag = 0'임을 부호화 하지 않을 수 있다. coded_sub_block_flag는 문맥 부호화 빈으로 부호화 되므로, 도 6에 따른 일 실시예에 의하면, 처리량 (throughput) 면에서도 이점을 가질 수 있다. 즉, 레지듀얼 코딩 과정에서 고주파 제로잉이 적용되어 제로아웃 영역 내 0이 아닌 유효 계수가 존재하지 않는 것으로 간주되는 경우, 변환 과정에서 요구되는 산술적 복잡도 (computational complexity)가 메모리 관점에서 줄어들 수 있다.
- [0115] 일 예시에서, 도 5 및 도 6에서 설명한 실시예들을 기반으로 도 7a 내지 도 8b에서 후술할 레지듀얼 코딩 방법이 수행될 수 있다. 다른 일 예시에서, 도 5 및 도 6에서 설명한 실시예들은 기반으로 도 9에서 후술할 인코딩 방법 또는 도 11에서 후술할 디코딩 방법이 수행될 수 있다.
- [0116] 도 7a 및 도 7b는 일 실시예에 따른 엔트로피 인코딩부의 구성 및 동작을 설명하기 위한 도면이다.
- [0117] 도 7a 및 도 7b를 참조하면, 인코딩 장치(엔트로피 인코딩부)는 (양자화된) 변환 계수들에 대한 레지듀얼 코딩 절차를 수행할 수 있다. 상기 (양자화된) 변환 계수들은, 일 예시에서는 고주파 제로잉이 적용된 변환 계수들일 수 있고, 다른 일 예시에서는 0이 아닌 고주파 계수들을 포함할 수 있으며, 이때 엔트로피 인코딩부에서 수행되는 레지듀얼 코딩 절차에서 고주파 계수들은 0으로 간주 또는 처리될 수 있다. 인코딩 장치는 도 4에서 전술한 바와 같이 현재 블록(현재 코딩 블록(CB) 또는 현재 변환 블록(TB)) 내 (양자화된) 변환 계수들을 스캔 순서에 따라 레지듀얼 코딩할 수 있다. 인코딩 장치는, 예를 들어 상기 표 3에 기재된 바와 같은 레지듀얼 정보에 관한 다양한 선택스 요소들을 생성 및 인코딩할 수 있다. S700 및 S710은 도 2의 레지듀얼 정보 인코딩 절차에 포함될 수 있다.
- [0118] 인코딩 장치는 레지듀얼 관련 선택스 요소들에 대한 이진화를 수행할 수 있다(S700). 일 예시에서, last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix에 대하여는 상기 도 4에서 전술된 실시예들에 따른 이진화가 수행될 수 있다. 상기 last_sig_coeff_x_prefix 및 상기 last_sig_coeff_y_prefix는 현재 블록 내 마지막 유효 계수의 위치를 기반으로 도출될 수 있다. 또한, 상기 표 3의 나머지 선택스 요소들에 대하여도 정해진 방법에 따라 이진화가 진행될 수 있다. 다른 일 예시에서, transform_skip_flag, last_sig_coeff_x_suffix, last_sig_coeff_y_suffix, coded_sub_block_flag, sig_coeff_flag, par_level_flag, rem_abs_gt1_flag, rem_abs_gt2_flag, coeff_sign_flag, mts_idx 등에 대하여는 고정 길이 이진화 과정(Fixed-length binarization process)에 따른 이진화가 수행될 수 있고, abs_remainder에 대하여는 그에 상응하는 이진화가 수행될 수 있다.

- [0119] 상기 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix는 현재 블록에 대한 변환 계수들 중 0이 아닌 마지막 변환 계수의 위치에 대한 라스트 유효 계수 프리픽스 정보(last significant coefficient prefix information)의 일 예시를 나타낼 수 있다. 보다 구체적으로, last_sig_coeff_x_prefix는 상기 라스트 유효 계수 프리픽스 정보 중 하나인 x축 프리픽스 정보의 일 예시를 나타낼 수 있고, last_sig_coeff_y_prefix는 상기 라스트 유효 계수 프리픽스 정보 중 하나인 y축 프리픽스 정보의 일 예시를 나타낼 수 있다. 상기 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix에 대한 이진화 과정에서, cRiceParam의 값은 0이 사용될 수 있다. 인코딩 장치는 상기 이진화 절차를 통하여 상기 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix 각각에 대한 빈 스트링(bin string)을 도출할 수 있다. 상기 이진화 절차는 엔트로피 인코딩부(240) 내의 이진화부(242)에 의하여 수행될 수 있다.
- [0120] 일 실시예에 따르면, 고주파 제로잉이 적용되는지 여부를 기반으로 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix 각각에 대한 cMax 값을 도출할 수 있다. cMax를 도출하기 위한 구체적인 수학적식은 도 4에서 전술한 바 있다. 상기 cMax는 last_sig_coeff_x_prefix 또는 last_sig_coeff_y_prefix에 대한 이진화 과정에서 도출되는 코드워드(빈 스트링)의 최대 길이를 나타낼 수 있다. cMax의 값을 줄이는 경우 last_sig_coeff_x_prefix 또는 last_sig_coeff_y_prefix에 대한 코드워드의 길이를 효과적으로 줄일 수 있으며, 상기 코드워드를 줄임으로써 절감되는 부호화 빈(bin)이 문맥 부호화(context coding) 빈이므로 영상 코딩 처리량 (throughput) 면에서 이점을 가질 수 있다.
- [0121] 인코딩 장치는 상기 레지듀얼 코딩 관련 신텍스 요소들에 대한 엔트로피 인코딩을 수행할 수 있다(S710). 인코딩 장치는 상술한 고주파 제로잉이 적용되는 영역에 대하여는 변환 계수 스캔을 생략할 수 있고, 상기 고주파 제로잉이 적용되는 영역에 대하여는 coded_sub_block_flag를 인코딩하지 않을 수 있다. 즉, 인코딩 장치는 레지듀얼 코딩 신텍스(또는 비트스트림)을 생성함에 있어서, 고주파 제로잉이 적용되는 영역에 위치하는 서브 블록에 대한 신텍스 요소 coded_sub_block_flag를 포함하지 않을 수 있다. 인코딩 장치는 고주파 제로잉이 적용되지 않는 영역, 즉 좌상단 변환 계수 영역(또는 저주파 변환 계수 영역)에 대하여만 coded_sub_block_flag를 인코딩하여 레지듀얼 코딩 신텍스(또는 비트스트림)에 포함시킬 수 있다. 이를 통해, 레지듀얼 코딩에 할당되는 비트수가 절감될 수 있다.
- [0122] 인코딩 장치는 CABAC (context-adaptive arithmetic coding) 또는 CAVLC (context-adaptive variable length coding) 등의 엔트로피 코딩 기법을 기반으로 상기 빈 스트링을 컨텍스트 기반 또는 바이패스 기반 인코딩할 수 있으며, 그 출력은 비트스트림에 포함될 수 있다. 상기 엔트로피 인코딩 절차는 엔트로피 인코딩부(240) 내의 엔트로피 인코딩 처리부(244)에 의하여 수행될 수 있다. 상기 비트스트림은 상술한 바와 같이 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix에 대한 정보를 포함하는 레지듀얼 정보 이외에도 예측 정보 등 영상/비디오 디코딩을 위한 다양한 정보를 포함할 수 있다. 상기 비트스트림은 (디지털) 저장매체 또는 네트워크를 통하여 디코딩 장치로 전달될 수 있다.
- [0123] 도 8a 및 도 8b는 일 실시예에 따른 엔트로피 디코딩부의 구성 및 동작 방법을 설명하기 위한 도면이다.
- [0124] 도 8a 및 도 8b를 참조하면, 디코딩 장치(엔트로피 디코딩부)는 인코딩된 레지듀얼 정보를 디코딩하여 (양자화된) 변환 계수들을 도출할 수 있다. 디코딩 장치는 도 4에서 전술된 바와 같이 현재 블록(현재 코딩 블록(Coding Block) 또는 현재 변환 블록(Transform Block))에 대한 인코딩된 레지듀얼 정보를 디코딩하여 (양자화된) 변환 계수들을 도출할 수 있다. 예를 들어, 디코딩 장치는 상기 표 3에 기재된 바와 같은 레지듀얼 정보에 관한 다양한 신텍스 요소들을 디코딩하고, 관련 신텍스 요소들의 값을 해석하고, 해석된 신텍스 요소들의 값을 기반으로 상기 (양자화된) 변환 계수들을 도출할 수 있다. S800 내지 S810은 상술한 도 3의 (양자화된) 변환 계수들을 도출하는 절차에 포함될 수 있다.
- [0125] 디코딩 장치는 레지듀얼 관련 신텍스 요소들에 대한 이진화를 수행할 수 있다(S800). 예를 들어, last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix에 대하여 상기 도 4에서 전술된 실시예들에 기반한 이진화를 수행할 수 있다. 이때 cRiceParam의 값은 0이 사용될 수 있다. 상기 이진화 절차는 디코딩 장치는 상기 이진화 절차를 통하여 상기 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix 각각의 가용 값들에 대한 가용 빈 스트링들을 도출할 수 있다. 상기 이진화 절차는 엔트로피 디코딩부(310) 내의 이진화부(312)에 의하여 수행될 수 있다. 일 실시예에 따르면, 고주파 제로잉이 적용되는지 여부를 기반으로 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix 각각에 대한 cMax 값을 도출할 수 있다. cMax를 도출하기 위한 구체적인 수학적식은 도 4에서 전술된 바 있다.
- [0126] 상기 cMax는 last_sig_coeff_x_prefix 또는 last_sig_coeff_y_prefix에 대한 이진화 과정에서 도출되는 코드워

드(빈 스트링)의 최대 길이를 나타낼 수 있다. cMax의 값을 줄이는 경우 last_sig_coeff_x_prefix 또는 last_sig_coeff_y_prefix에 대한 코드워드의 길이를 효과적으로 줄일 수 있으며, 상기 코드워드를 줄임으로써 절감되는 부호화 빈(bin)이 문맥 부호화(context coding) 빈이므로 영상 코딩 처리량 (throughput) 면에서 이점을 가질 수 있다.

[0127] 한편, 상기 표 3의 나머지 선택 요소들에 대하여도 정해진 방법에 따라 이진화가 진행될 수 있다. 예를 들어, transform_skip_flag, last_sig_coeff_x_suffix, last_sig_coeff_y_suffix, coded_sub_block_flag, sig_coeff_flag, par_level_flag, rem_abs_gt1_flag, rem_abs_gt2_flag, coeff_sign_flag, mts_idx 등에 대하여는 고정 길이 이진화 과정(Fixed-length binarization process)에 따른 이진화가 수행될 수 있고, abs_remainder에 대하여는 그에 상응하는 이진화가 수행될 수 있다.

[0128] 디코딩 장치는 상기 레지듀얼 코딩 관련 선택 요소들에 대한 엔트로피 디코딩을 수행할 수 있다(S810). 디코딩 장치는 각 선택 요소에 대한 각 빈들을 순차적으로 파싱 및 디코딩하면서, 도출된 빈 스트링을 상기 가용 빈 스트링들과 비교할 수 있다. 만약 도출된 빈 스트링이 상기 가용 빈 스트링들 중 하나와 같으면, 해당 빈 스트링에 대응하는 값이 해당 선택 요소의 값으로 도출될 수 있다. 만약 도출된 빈 스트링이 상기 가용 빈 스트링들 중 어느 하나와도 같지 않은 경우, 상기 비트스트림 내 다음 비트를 더 파싱하고 디코딩한 후 상기 비교 절차를 수행할 수 있다. 이러한 과정을 통하여 비트스트림 내에 특정 정보(특정 선택 요소)에 대한 시작 비트나 끝 비트를 사용하지 않고도 가변 길이 비트를 이용하여 해당 정보를 시그널링할 수 있고, 이를 통하여 낮은 값에 대하여는 상대적으로 더 적은 비트를 할당함으로써 전반적인 코딩 효율을 높일 수 있다.

[0129] 디코딩 장치는 고주파 제로잉이 적용되는 영역에 대하여는 변환 계수 스캔을 생략할 수 있고, 고주파 제로잉이 적용되는 영역에 대하여는 coded_sub_block_flag를 파싱 및/또는 디코딩하지 않고, 0으로 간주할 수 있다. 즉, 디코딩 장치는 고주파 제로잉이 적용되는 영역에 위치하는 서브 블록에 대하여는 스캔을 수행하지 않고도 마지막 유효 계수가 위치하지 않음을 추론할 수 있고, 고주파 제로잉이 적용되는 영역에 위치하는 서브 블록에 대한 선택 요소 coded_sub_block_flag를 레지듀얼 코딩 선택(또는 비트스트림)으로부터 파싱(parsing)하지 않을 수 있으며, 해당 선택 요소 coded_sub_block_flag의 값을 0으로 추론할 수 있다.

[0130] 또한, 디코딩 장치는 현재 블록에 대한 레지듀얼 정보(예를 들어, 레지듀얼 코딩 선택) 내에서 파싱 및/또는 디코딩된 선택 요소 coded_sub_block_flag가 고주파 제로잉이 적용되는 영역에 대한 것이 아님을 추론할 수 있고, 좌상단 변환 블록에 대한 레지듀얼 정보(예를 들어, 레지듀얼 선택) 내에서 처음 파싱 및/또는 디코딩된 선택 요소 coded_sub_block_flag는 스캔 순서를 기반으로 도출되는 첫 번째 서브 블록에 대한 것임을 추론할 수 있다.

[0131] 디코딩 장치는 CABAC 또는 CAVLC 등의 엔트로피 코딩 기법을 기반으로 비트스트림으로부터 상기 빈 스트링 내 각 빈들을 컨텍스트 기반 또는 바이패스 기반으로 디코딩할 수 있다. 상기 엔트로피 디코딩 절차는 엔트로피 디코딩부(310) 내의 엔트로피 디코딩 처리부(314)에 의하여 수행될 수 있다. 디코딩 장치는 상기 last_sig_coeff_x_prefix의 값 및 last_sig_coeff_y_prefix의 값을 기반으로 마지막 유효 계수의 위치를 도출할 수 있다. 구체적인 계산은 예를 들어 아래의 표 6을 기반으로 수행될 수 있다.

표 6

[0132]

The column position of the last significant coefficient in scanning order within a transform block LastSignificantCoeffX is derived as follows:- If last_sig_coeff_x_suffix is not present, the following applies:LastSignificantCoeffX = last_sig_coeff_x_prefix- Otherwise (last_sig_coeff_x_suffix is present), the following applies:LastSignificantCoeffX=(1<<((last_sig_coeff_x_prefix>>1)-1)) * (2+(last_sig_coeff_x_prefix&1)) + last_sig_coeff_x_suffix The row position of the last significant coefficient in scanning order within a transform block LastSignificantCoeffY is derived as follows:- If last_sig_coeff_y_suffix is not present, the following applies:LastSignificantCoeffY = last_sigcoeff_y_prefix- Otherwise (last_sig_coeff_y_suffix is present), the following applies:LastSignificantCoeffY=(1<<((last_sig_coeff_y_prefix>>1)-1))*(2+(last_sig_coeff_y_prefix&1))+last_sig_coeff_y_suffix
--

[0133] 표 6에서 LastSignificantCoeffX는 현재 (변환) 블록 내 0이 아닌 마지막 유효 계수의 x축 위치를 나타내고, LastSignificantCoeffY는 현재 (변환) 블록 내 0이 아닌 마지막 유효 계수의 y축 위치를 나타낼 수 있다.

[0134] 상기 비트스트림은 상술한 바와 같이 last_sig_coeff_x_prefix 및 last_sig_coeff_y_prefix에 대한 정보를 포

합하는 레지듀얼 정보 이외에도 예측 정보 등 영상/비디오 디코딩을 위한 다양한 정보를 포함할 수 있다. 상기 비트스트림은 (디지털) 저장매체 또는 네트워크를 통하여 디코딩 장치로 전달될 수 있음은 전술한 바와 같다.

- [0135] 디코딩 장치는 상기 (양자화된) 변환 계수들을 기반으로 역양자화 및/또는 역변환 절차를 수행하여 현재 블록에 대한 레지듀얼 샘플들을 도출할 수 있다. 상기 레지듀얼 샘플들 및 인터/인트라 예측을 통하여 도출된 예측 샘플들을 기반으로 복원 샘플들이 생성될 수 있으며, 상기 복원 샘플들을 포함하는 복원 픽처가 생성될 수 있다.
- [0136] 도 9는 일 실시예에 따른 인코딩 장치의 동작을 도시하는 흐름도이고, 도 10은 일 실시예에 따른 인코딩 장치의 구성을 도시하는 블록도이다.
- [0137] 도 9 및 도 10에 따른 인코딩 장치는 도 11 및 도 12에 따른 디코딩 장치와 대응되는 동작들을 수행할 수 있다. 따라서, 도 11 및 도 12에서 후술될 디코딩 장치의 동작들은 도 9 및 도 10에 따른 인코딩 장치에도 마찬가지로 적용될 수 있다.
- [0138] 도 9에 개시된 각 단계는 도 2에 개시된 인코딩 장치(200)에 의하여 수행될 수 있다. 보다 구체적으로, S900은 도 2에 개시된 감산부(231)에 의하여 수행될 수 있고, S910은 도 2에 개시된 변환부(232)에 의하여 수행될 수 있고, S920은 도 2에 개시된 양자화부(233)에 의하여 수행될 수 있고, S930은 도 2에 개시된 엔트로피 인코딩부(240)에 의하여 수행될 수 있다. 더불어 S900 내지 S930에 따른 동작들은, 도 4 내지 도 8에서 전술된 내용들 중 일부를 기반으로 한 것이다. 따라서, 도 2 및 도 4 내지 도 8b에서 전술된 내용과 중복되는 구체적인 내용은 설명을 생략하거나 간단히 하기로 한다.
- [0139] 도 10에 도시된 바와 같이, 일 실시예에 따른 인코딩 장치는 감산부(231), 변환부(232), 양자화부(233) 및 엔트로피 인코딩부(240)를 포함할 수 있다. 그러나, 경우에 따라서는 도 10에 도시된 구성 요소 모두가 인코딩 장치의 필수 구성 요소가 아닐 수 있고, 인코딩 장치는 도 10에 도시된 구성 요소보다 많거나 적은 구성 요소에 의해 구현될 수 있다.
- [0140] 일 실시예에 따른 인코딩 장치에서 감산부(231), 변환부(232), 양자화부(233) 및 엔트로피 인코딩부(240)는 각각 별도의 칩(chip)으로 구현되거나, 적어도 둘 이상의 구성 요소가 하나의 칩을 통해 구현될 수도 있다.
- [0141] 일 실시예에 따른 인코딩 장치는, 현재 블록에 대한 레지듀얼 샘플들을 도출할 수 있다(S900). 보다 구체적으로, 인코딩 장치의 감산부(231)는 현재 블록에 대한 레지듀얼 샘플들을 도출할 수 있다.
- [0142] 일 실시예에 따른 인코딩 장치는, 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출할 수 있다(S910). 보다 구체적으로, 인코딩 장치의 변환부(232)는 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출할 수 있다.
- [0143] 일 실시예에 따른 인코딩 장치는, 양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출할 수 있다(S920). 보다 구체적으로, 인코딩 장치의 양자화부(233)는 양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출할 수 있다.
- [0144] 일 실시예에 따른 인코딩 장치는, 상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩할 수 있다(S930). 보다 구체적으로, 인코딩 장치의 엔트로피 인코딩부(240)는 상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩할 수 있다.
- [0145] 일 실시예에서, 상기 현재 블록은 현재 변환 블록(TB 또는 TU) 또는 현재 코딩 블록(CB 또는 CU)을 나타낼 수 있다.
- [0146] 일 실시예에서, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역에 관련될 수 있다.
- [0147] 일 실시예에서, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝이 수행될 수 있다.
- [0148] 일 실시예에서, 상기 레지듀얼 정보는 상기 현재 블록 내 서브 블록에 대한 변환 계수들의 변환 계수 레벨이 모두 0인지 여부를 나타내는 코디드 서브 블록 플래그(coded subblock flag)를 포함하고, 상기 서브 블록은 상기 저주파 변환 계수 영역과 관련될 수 있다. 일 예시에서, 상기 코디드 서브 블록 플래그는 coded_sub_block_flag로 표기될 수 있다.
- [0149] 일 실시예에서, 상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대해서는 상기 변환 계수 스캐닝이 수행되

지 않을 수 있다.

- [0150] 일 실시예에서, 상기 저주파 변환 계수 영역의 폭 또는 상기 저주파 변환 계수 영역의 높이는 32 이하일 수 있다.
- [0151] 일 실시예에서, 상기 현재 블록의 폭이 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 상기 현재 블록의 상기 폭과 동일한 것으로 결정될 수 있다. 상기 현재 블록의 높이가 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 상기 현재 블록의 상기 높이와 동일한 것으로 결정될 수 있다. 상기 현재 블록의 상기 폭이 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 32로 결정될 수 있다. 상기 현재 블록의 상기 높이가 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 32로 결정될 수 있다.
- [0152] 일 실시예에서, 상기 저주파 변환 계수 영역의 상기 폭 및 상기 저주파 변환 계수 영역의 상기 높이는 아래의 수학적 식 5를 기반으로 결정될 수 있다.
- [0153] [수학적 식 5]
- [0154] $\log_2(\text{ZoTbWidth}) = \text{Min}(\log_2(\text{TbWidth}), 5)$
- [0155] $\log_2(\text{ZoTbHeight}) = \text{Min}(\log_2(\text{TbHeight}), 5)$
- [0156] 상기 수학적 식 5에서 상기 ZoTbWidth는 상기 저주파 변환 계수 영역의 상기 폭을 나타내고, 상기 ZoTbHeight는 상기 저주파 변환 계수 영역의 상기 높이를 나타내고, 상기 TbWidth는 상기 현재 블록의 폭을 나타내고, 상기 TbHeight는 상기 현재 블록의 높이를 나타낼 수 있다.
- [0157] 일 실시예에서, 상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대하여 제로잉이 적용되고, 상기 제로잉이 적용된 변환 계수들의 개수는 상기 저주파 변환 계수 영역의 상기 폭 또는 상기 저주파 변환 계수 영역의 상기 높이에 기반하여 결정될 수 있다.
- [0158] 도 9 및 도 10의 인코딩 장치 및 인코딩 장치의 동작 방법에 따르면, 인코딩 장치는 현재 블록에 대한 레지듀얼 샘플들을 도출하고(S900), 상기 현재 블록에 대한 상기 레지듀얼 샘플들을 변환하여 상기 현재 블록에 대한 변환 계수들을 도출하고(S910), 양자화 과정에 기반하여 상기 변환 계수들로부터 양자화된 변환 계수들을 도출하고(S920), 상기 양자화된 변환 계수들에 대한 정보를 포함하는 레지듀얼 정보를 인코딩(S930) 할 수 있고, 이때 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역에 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝(transform coefficient scanning)이 수행될 수 있다. 즉, 고주파 제로잉이 적용되는 영역과 관련된 변환 계수 그룹에 대한 변환 계수 스캔을 생략하고, 상기 변환 계수 그룹에 대한 코디드 서브 블록 플래그(coded subblock flag)의 시그널링을 생략함으로써 영상 코딩 효율을 높일 수 있다.
- [0159] 도 11은 일 실시예에 따른 디코딩 장치의 동작을 도시하는 흐름도이고, 도 12는 일 실시예에 따른 디코딩 장치의 구성을 도시하는 블록도이다.
- [0160] 도 11에 개시된 각 단계는 도 3에 개시된 디코딩 장치(300)에 의하여 수행될 수 있다. 보다 구체적으로, S1100 및 S1110은 도 3에 개시된 엔트로피 디코딩부(310)에 의하여 수행될 수 있고, S1120은 도 3에 개시된 역양자화부(321)에 의하여 수행될 수 있고, S1130은 도 3에 개시된 역변환부(322)에 의하여 수행될 수 있고, S1140은 도 3에 개시된 가산부(340)에 의하여 수행될 수 있다. 더불어 S1100 내지 S1140에 따른 동작들은, 도 4 내지 도 8b에서 전술된 내용들 중 일부를 기반으로 한 것이다. 따라서, 도 3 내지 도 8b에서 전술된 내용과 중복되는 구체적인 내용은 설명을 생략하거나 간단히 하기로 한다.
- [0161] 도 12에 도시된 바와 같이, 일 실시예에 따른 디코딩 장치는 엔트로피 디코딩부(310), 역양자화부(321), 역변환부(322) 및 가산부(340)를 포함할 수 있다. 그러나, 경우에 따라서는 도 12에 도시된 구성 요소 모두가 디코딩 장치의 필수 구성 요소가 아닐 수 있고, 디코딩 장치는 도 12에 도시된 구성 요소보다 많거나 적은 구성 요소에 의해 구현될 수 있다.
- [0162] 일 실시예에 따른 디코딩 장치에서 엔트로피 디코딩부(310), 역양자화부(321), 역변환부(322) 및 가산부(340)는 각각 별도의 칩(chip)으로 구현되거나, 적어도 둘 이상의 구성 요소가 하나의 칩을 통해 구현될 수도 있다.
- [0163] 일 실시예에 따른 디코딩 장치는, 레지듀얼 정보를 포함하는 비트스트림을 수신할 수 있다(S1100). 보다 구체적

으로, 디코딩 장치의 엔트로피 디코딩부(310)는 레지듀얼 정보를 포함하는 비트스트림을 수신할 수 있다.

- [0164] 일 실시예에 따른 디코딩 장치는, 비트스트림에 포함된 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출할 수 있다(S1110). 보다 구체적으로, 디코딩 장치의 엔트로피 디코딩부(310)는 비트스트림에 포함된 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출할 수 있다.
- [0165] 일 실시예에 따른 디코딩 장치는, 역양자화 과정에 기반하여 양자화된 변환 계수들로부터 변환 계수들을 도출할 수 있다(S1120). 보다 구체적으로, 디코딩 장치의 역양자화부(321)는 역양자화 과정에 기반하여 양자화된 변환 계수들로부터 변환 계수들을 도출할 수 있다.
- [0166] 일 실시예에 따른 디코딩 장치는, 도출된 변환 계수들에 역변환을 적용하여 현재 블록에 대한 레지듀얼 샘플들을 도출할 수 있다(S1130). 보다 구체적으로, 디코딩 장치의 역변환부(322)는 도출된 변환 계수들에 역변환을 적용하여 현재 블록에 대한 레지듀얼 샘플들을 도출할 수 있다.
- [0167] 일 실시예에 따른 디코딩 장치는, 현재 블록에 대한 레지듀얼 샘플을 기반으로 복원 픽처를 생성할 수 있다(S1140). 보다 구체적으로, 디코딩 장치의 가산부(340)는 현재 블록에 대한 레지듀얼 샘플을 기반으로 복원 픽처를 생성할 수 있다.
- [0168] 일 실시예에서, 상기 현재 블록의 단위는 변환 블록(Transform Block, TB)일 수 있다. 다른 일 실시예에서, 상기 현재 블록의 단위는 코딩 블록(Coding Block, CB)일 수 있다.
- [0169] 일 실시예에서, 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역에 관련될 수 있다.
- [0170] 일 실시예에서, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝(transform coefficient scanning)이 수행될 수 있다.
- [0171] 일 실시예에서, 상기 레지듀얼 정보는 상기 현재 블록 내 서브 블록에 대한 변환 계수들의 변환 계수 레벨이 모두 0인지 여부를 나타내는 코디드 서브 블록 플래그(coded subblock flag)를 포함하고, 상기 서브 블록은 상기 저주파 변환 계수 영역과 관련될 수 있다. 일 예시에서, 상기 코디드 서브 블록 플래그는 coded_sub_block_flag로 표기될 수 있다.
- [0172] 일 실시예에서, 상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대해서는 상기 변환 계수 스캐닝이 수행되지 않을 수 있다.
- [0173] 일 실시예에서, 상기 저주파 변환 계수 영역의 폭 또는 상기 저주파 변환 계수 영역의 높이는 32 이하일 수 있다.
- [0174] 일 실시예에서, 상기 현재 블록의 폭이 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 상기 현재 블록의 상기 폭과 동일한 것으로 결정될 수 있다. 상기 현재 블록의 높이가 32 미만이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 상기 현재 블록의 상기 높이와 동일한 것으로 결정될 수 있다. 상기 현재 블록의 상기 폭이 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 폭은 32로 결정될 수 있다. 상기 현재 블록의 상기 높이가 32 이상이라는 결정을 기반으로, 상기 저주파 변환 계수 영역의 상기 높이는 32로 결정될 수 있다.
- [0175] 일 실시예에서, 상기 저주파 변환 계수 영역의 상기 폭 및 상기 저주파 변환 계수 영역의 상기 높이는 아래의 수학적 식 6을 기반으로 결정될 수 있다.
- [0176] [수학적 식 6]
- [0177] $\log_2(\text{ZoTbWidth}) = \text{Min}(\log_2(\text{TbWidth}), 5)$
- [0178] $\log_2(\text{ZoTbHeight}) = \text{Min}(\log_2(\text{TbHeight}), 5)$
- [0179] 상기 수학적 식 6에서 상기 ZoTbWidth는 상기 저주파 변환 계수 영역의 상기 폭을 나타내고, 상기 ZoTbHeight는 상기 저주파 변환 계수 영역의 상기 높이를 나타내고, 상기 TbWidth는 상기 현재 블록의 폭을 나타내고, 상기 TbHeight는 상기 현재 블록의 높이를 나타낼 수 있다.
- [0180] 일 실시예에서, 상기 고주파 변환 계수 영역과 관련된 변환 계수들에 대하여 제로잉(zeroing)이 적용되고, 상기 제로잉이 적용된 변환 계수들의 개수는 상기 저주파 변환 계수 영역의 상기 폭 또는 상기 저주파 변환 계수 영

역의 상기 높이에 기반하여 결정될 수 있다.

[0181] 도 11 및 도 12에 개시된 디코딩 장치 및 디코딩 장치의 동작 방법에 따르면, 디코딩 장치는 레지듀얼 정보를 포함하는 비트스트림을 수신하고(S1100), 상기 비트스트림에 포함된 상기 레지듀얼 정보를 기반으로 현재 블록에 대한 양자화된 변환 계수들을 도출하고(S1110), 역양자화 과정에 기반하여 양자화된 변환 계수들로부터 변환 계수들을 도출하고(S1120), 도출된 변환 계수들에 역변환을 적용하여 현재 블록에 대한 레지듀얼 샘플들을 도출하고(S1130), 현재 블록에 대한 레지듀얼 샘플들을 기반으로 복원 픽처를 생성(S1140)할 수 있고, 이때 상기 현재 블록에 대한 상기 변환 계수들 각각은, 변환 계수 0으로 구성된 고주파 변환 계수 영역 또는 적어도 하나의 유효 변환 계수를 포함하는 저주파 변환 계수 영역에 관련되고, 상기 현재 블록에 대한 변환 계수들 중 상기 저주파 변환 계수 영역과 관련된 변환 계수들에 대해서 변환 계수 스캐닝(transform coefficient scanning)이 수행되는 것을 특징으로 할 수 있다. 즉, 즉, 고주파 제로잉이 적용되는 영역과 관련된 변환 계수 그룹에 대한 변환 계수 스캔을 생략하고, 상기 변환 계수 그룹에 대한 코디드 서브 블록 플래그(coded subblock flag)의 시그널링을 생략함으로써 영상 코딩 효율을 높일 수 있다.

[0182] 일 실시예에서, 도 4 내지 도 10에서 전술된 레지듀얼 코딩 과정은 아래의 영문 스펙의 내용에 기반할 수 있다.

[0183] Abstract

[0184] In this proposal, the binarization of last significant coefficient position is modified to reduce the maximum number of context coded bins. Specifically, the number of context coded bins for large block (i.e., 64×64 , $64 \times N$, $N \times 64$) in worst case scenario is reduced from 11 to 9. Experimental results show 0.01%, 0%, and -0.02% BD-rate reductions on Y, Cb, and Cr components, respectively, compared to VTM3.0 in all-intra configuration, and 0.01%, -0.01%, and -0.01% BD-rate reductions in random access configuration.

[0185] 1 Introduction

[0186] It is known that large block partitions typically present less residuals and the energy is more concentrated in low-frequency coefficients in the transform domain. In VTM3.0 [1], high frequency transform coefficients are zeroed out for the transform blocks with size (width or height, or both width and height) equal to 64. Thus, for a $W \times H$ transform block, where W indicates the block width and H the block height, only the top-left $(W == 64 ? 32 : W) \times (H == 64 ? 32 : H)$ transform coefficients are retained.

[0187] In [1], the prefix `last_sig_coeff_x_prefix` and `last_sig_coeff_y_prefix` are both context coded using truncated Rice binarization with $cMax = (\log_2 TbSize \ll 1) - 1$. Here, if the syntax element to be parsed is `last_sig_coeff_x_prefix`, $\log_2 TbSize$ is set equal to $\log_2 TbWidth$ and otherwise, $\log_2 TbSize$ is set equal to $\log_2 TbHeight$. That is, the maximum possible magnitude is determined by the transform block width or height. In the worst case scenario, the number of the bins that use context modelling is equal to 11. 표 7 shows the binarization for $W = 64$ or $H = 64$ in VTM3.0, wherein X means 0 or 1.

표 7

[0188]

Magnitude of last position component	<code>last_sig_coeff_x_prefix</code> <code>last_sig_coeff_y_prefix</code> (context coded bin)	<code>last_sig_coeff_x_suffix</code> <code>last_sig_coeff_y_suffix</code> (bypass bin)
0	0	
1	10	
2	110	
3	1110	
4, 5	11110	X
6, 7	111110	X
8~11	1111110	XX
12~15	11111110	XX
16~23	111111110	XXX
24~31	1111111110	XXX
32~47	11111111110	XXXX
48~63	11111111111	XXXX

[0189] 2. Proposed method This contribution is proposed on top of JVET-M0250 [2]. In the proposed method, whether the current coefficient group belongs to the high-frequency zeroing region is checked. Using this information, the unnecessary coded_sub_block_flag (CSBF) coding for the high-frequency zeroing region can be skipped.

[0190] In one embodiment, CSBF coding method considering the high-frequency zeroing region may be proposed. If a first condition for Last or first coefficient group is satisfied, the value of CSBF may be inferred to 1. If the first condition for last or first coefficient group is not satisfied, a second condition for high-frequency zeroing region is checked. If the second condition for high-frequency zeroing region is satisfied, there is no CSBF coding. If the second condition for high-frequency zeroing is not satisfied, context index is derived and CSBF coding is performed.

[0191] A last position coding scheme is proposed for large block-size transforms. Compared to VTM3.0, the proposed coding scheme uses less context coded bins in the worst case scenario. The codeword in the proposed scheme still starts with a truncated Rice code and followed by a fixed length code. After high-frequency zeroing, for a $W \times H$ transform block, only the top-left $\min(W, 32) \times \min(H, 32)$ transform coefficients are kept. Thus, the maximum possible codeword length of the prefix last_sig_coeff_x_prefix or last_sig_coeff_y_prefix is derived as:

[0192] $c_{Max} = (\min(\log_2 T_bSize, 5) \ll 1) - 1.$

[0193] 表 8 shows the binarization for $W = 64$ or $H = 64$, wherein X means 0 or 1. Here, the different parts are emphasized.

表 8

[0194]

Magnitude of last position component	<i>last_sig_coeff_x_prefix</i> or <i>last_sig_coeff_y_prefix</i> (context coded bin)	<i>last_sig_coeff_x_suffix</i> or <i>last_sig_coeff_y_suffix</i> (bypass bin)
0	0	
1	10	
2	110	
3	1110	
4, 5	11110	X
6, 7	111110	X
8~11	1111110	XX
12~15	11111110	XX
16~23	111111110	XXX
24~31	111111111	XXX
32~47	N/A	N/A
48~63	N/A	N/A

表 9

[0195]

Magnitude of last position component	VTM3.0	Proposed method
0	1	1
1	2	2
2	3	3
3	4	4
4, 5	5	5
6, 7	6	6
8~11	7	7
12~15	8	8
16~23	9	9
24~31	10	9

32~47	11	N/A
48~63	11	N/A

[0196] 표 9 shows comparison of codeword length for the prefix (context coded bins) when W = 64 or H = 64 and VTM3.0. The context coded bins can be as long as 9 in the proposed method, while it is up to 11 bins in VTM3.0. Note that when the magnitude of the last position component in the range of 24-31, the number of context coded bins is reduced from 10 to 9. 3. Experimental results

[0197] The proposed method has been implemented on the VTM3.0 software. The simulations were performed following the common test conditions defined in JVET-L1010 [3]. In every case, the anchor is the VTM3.0 software. Encoding time and decoding time come from the cross-check results [4]. 표 10 shows Experimental results for all-intra (AI) test condition; anchor is VTM3.0.

표 10

[0198]

	All Intra Main10				
	Over VTM-3.0				
	Y	U	V	EncT	DecT
Class A1	0.03%	-0.07%	0.03%	100%	101%
Class A2	0.02%	0.00%	-0.03%	100%	100%
Class B	0.01%	0.05%	-0.02%	100%	102%
Class C	0.00%	0.01%	-0.03%	100%	101%
Class E	0.02%	-0.02%	-0.03%	100%	99%
Overall	0.01%	0.00%	-0.02%	100%	101%
Class D	0.00%	0.01%	-0.01%	100%	100%
Class F	-0.01%	0.03%	0.02%	100%	99%

[0199] 표 11 shows Experimental results for random-access (RA) test condition; anchor is VTM3.0.

표 11

[0200]

	Random access Main10				
	Over VTM-3.0				
	Y	U	V	EncT	DecT
Class A1	0.05%	-0.09%	0.00%		
Class A2	0.01%	0.07%	0.03%		
Class B	-0.01%	-0.02%	-0.04%	100%	100%
Class C	0.00%	0.02%	-0.01%	99%	100%
Class E					
Overall	0.01%	-0.01%	-0.01%		
Class D	0.00%	0.03%	0.06%	100%	100%
Class F	-0.01%	-0.03%	-0.07%	100%	99%

[0201] 4. Reference[1] B. Bross, et al., "Video Coding (Draft 3)," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC 29/WG 11 JVET-L1001, 12th meeting, Macao, CN, 3-12 Oct. 2018.

[0202] [2] J. Choi, et al., "Non-CE7: Simplified CSBF coding for large block-size transforms," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC 29/WG 11 JVET-M0250, 13th meeting, Marrakech, MA, 9-18 Jan. 2019.

[0203] [3] F. Bossen, et al., "common test conditions and software reference configurations for SDR video Joint Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC 29/WG 11 JVET-L1010, 12th Meeting, Macao, CN, 3-12 Oct. 2018.

[0204] [4] H. Schwarz, "Crosscheck of JVET-M0251 (Non-CE7: Last position coding for large block-size

transforms),” Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC 29/WG 11 JVET-M0646, 13th meeting, Marrakech, MA, 9-18 Jan. 2019.

[0205] 5. Patent rights declaration(s)

[0206] LG Electronics Inc. may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

[0207] 6. Specification

표 12

[0208]	residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
	if(transform_skip_enabled_flag && (cIdx != 0 tu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
	transform_skip_flag[x0][y0][cIdx]	ae(v)
	last_sig_coeff_x_prefix	ae(v)
	last_sig_coeff_y_prefix	ae(v)
	if(last_sig_coeff_x_prefix > 3)	
	last_sig_coeff_x_suffix	ae(v)
	if(last_sig_coeff_y_prefix > 3)	
	last_sig_coeff_y_suffix	ae(v)
	log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
	numSbCoeff = 1 << (log2SbSize << 1)	
	lastScanPos = numSbCoeff	
	log2TbWidth = Min(log2TbWidth, 5)	
	log2TbHeight = Min(log2TbHeight, 5)	
	lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) - 1	
	do {	
	if(lastScanPos == 0) {	
	lastScanPos = numSbCoeff	
	lastSubBlock--	
	}	
	lastScanPos--	
	xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][0]	
	yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][1]	
	xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize] [lastScanPos][0]	
	yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize] [lastScanPos][1]	
	} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
	numSigCoeff = 0	
	QState = 0	
	for(i = lastSubBlock; i >= 0; i--) {	
	startQStateSb = QState	
	xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][0]	
	yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][1]	
	inferSbDcSigCoeffFlag = 0	
	if((i < lastSubBlock) && (i > 0)) {	
	coded_sub_block_flag[xS][yS]	ae(v)
	inferSbDcSigCoeffFlag = 1	

} firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
remBinsPass1 = (log2SbSize < 2 ? 6 : 28)	
remBinsPass2 = (log2SbSize < 2 ? 2 : 4)	
firstPosMode0 = (i == lastSubBlock ? lastScanPos - 1 : numSbCoeff - 1)	
firstPosMode1 = -1	
firstPosMode2 = -1	
for(n = (i == firstPosMode0; n >= 0 && remBinsPass1 >= 3; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][
0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][
1]	
if(coded_sub_block_flag[xS][yS] && (n > 0	
!inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
numSigCoeff++	
abs_level_gt1_flag[n]	ae(v)
remBinsPass1--	
if(abs_level_gt1_flag[n]) {	
par_level_flag[n]	ae(v)
remBinsPass1--	
if(remBinsPass2 > 0) {	
remBinsPass2--	
if(remBinsPass2 == 0)	
firstPosMode1 = n - 1	
}	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] +	
par_level_flag[n] + abs_level_gt1_flag[n]	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
if(remBinsPass1 < 3)	
firstPosMode2 = n - 1	
}	
if(firstPosMode1 < firstPosMode2)	
firstPosMode1 = firstPosMode2	
for(n = numSbCoeff - 1; n >= firstPosMode2; n--)	
if(abs_level_gt1_flag[n])	
abs_level_gt3_flag[n]	ae(v)
for(n = numSbCoeff - 1; n >= firstPosMode1; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][
0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][
1]	
if(abs_level_gt3_flag[n])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] +	2 * (
abs_level_gt3_flag[n] + abs_remainder[n]))
}	

for(n = firstPosMode1; n > firstPosMode2; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(abs_level_gt1_flag[n])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
}	
for(n = firstPosMode2; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
dec_abs_level[n]	ae(v)
if(AbsLevel[xC][yC] > 0)	
firstSigScanPosSb = n	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
if(dep_quant_enabled_flag !sign_data_hiding_enabled_flag)	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC] && (!signHidden (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ae(v)
}	
if(dep_quant_enabled_flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC])	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff_sign_flag[n])	
QState = QStateTransTable[QState][par_level_flag[n]]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC]) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	

}	
}	
}	
}	
if(tu_mts_flag[x0][y0] && (cIdx == 0))	
mts_idx[x0][y0][cIdx]	ae(v)
}	

[0209] last_sig_coeff_x_prefix specifies the prefix of the column position of the last significant coefficient in scanning order within a transform block. The values of last_sig_coeff_x_prefix shall be in the range of 0 to $(\text{Min}(\log_2\text{TbWidth}, 5) \ll 1) - 1$, inclusive. last_sig_coeff_y_prefix specifies the prefix of the row position of the last significant coefficient in scanning order within a transform block. The values of last_sig_coeff_y_prefix shall be in the range of 0 to $(\text{Min}(\log_2\text{TbHeight}, 5) \ll 1) - 1$, inclusive.

[0210] ㉟ 13 below shows syntax elements and associated binarizations.

㉟ 13

Syntax structure	Syntax element	Binarization	
		Process	Input parameters
tile_group_data()	end_of_tile_group_flag	FL	cMax = 1
coding_tree_unit()	alf_ctb_flag[][]	FL	cMax = 1
sao()	sao_merge_left_flag	FL	cMax = 1
	sao_merge_up_flag	FL	cMax = 1
	sao_type_idx_luma	TR	cMax = 2, cRiceParam = 0
	sao_type_idx_chroma	TR	cMax = 2, cRiceParam = 0
	sao_offset_abs[][][]	TR	cMax = $(1 \ll (\text{Min}(\text{bitDepth}, 10) - 5)) - 1$, cRiceParam = 0
	sao_offset_sign[][][]	FL	cMax = 1
	sao_band_position[][]	FL	cMax = 31
	sao_eo_class_luma	FL	cMax = 3
	sao_eo_class_chroma	FL	cMax = 3
coding_quadtree()	qt_split_cu_flag[][]	FL	cMax = 1
multi_type_tree()	mtt_split_cu_flag	FL	cMax = 1
	mtt_split_cu_vertical_flag	FL	cMax = 1
	mtt_split_cu_binary_flag	FL	cMax = 1
coding_unit()	cu_skip_flag[][]	FL	cMax = 1
	pred_mode_flag	FL	cMax = 1
	pcm_flag[][]	FL	cMax = 1
	intra_luma_ref_idx[][]	TR	cMax = 2, cRiceParam = 0

[0211]

	intra_luma_mpm_flag[][]	FL	cMax = 1
	intra_luma_mpm_idx[][]	TR	cMax = 5, cRiceParam = 0
	intra_luma_mpm_remainder[][]	TB	cMax = 60
	intra_chroma_pred_mode[][]	9.5.3.7	-
	merge_flag[][]	FL	cMax = 1
	inter_pred_idc[x0][y0]	9.5.3.8	cbWidth, cbHeight
	inter_affine_flag[][]	FL	cMax = 1
	cu_affine_type_flag[][]	FL	cMax = 1
	ref_idx_l0[][]	TR	cMax = num_ref_idx_l0_active_minus1, cRiceParam = 0
	mvp_l0_flag[][]	FL	cMax = 1
	ref_idx_l1[][]	TR	cMax = num_ref_idx_l1_active_minus1, cRiceParam = 0
	mvp_l1_flag[][]	FL	cMax = 1
	avmr_flag[][]	FL	cMax = 1
	avmr_4pel_flag[][]	FL	cMax = 1
	gbi_idx[][]	TR	cMax = NoBackwardPredFlag ? 4 : 2
	cu_cbf	FL	cMax = 1

[0212]

	tu_cbf_cr[][][]	FL	cMax = 1
	cu_qp_delta_abs	9.5.3.9	-
	cu_qp_delta_sign_flag	FL	cMax = 1
	tu_mts_flag[][]	FL	cMax = 1
residual_coding()	transform_skip_flag[][][]	FL	cMax = 1
	last_sig_coeff_x_prefix	TR	cMax = (Min(log2TbWidth, 5) << 1) - 1, cRiceParam = 0
	last_sig_coeff_y_prefix	TR	cMax = (Min(log2TbHeight, 5) << 1) - 1, cRiceParam = 0
	last_sig_coeff_x_suffix	FL	cMax = (1 << ((last_sig_coeff_x_prefix >> 1) - 1) - 1)
	last_sig_coeff_y_suffix	FL	cMax = (1 << ((last_sig_coeff_y_prefix >> 1) - 1) - 1)
	coded_sub_block_flag[][]	FL	cMax = 1
	sig_coeff_flag[][]	FL	cMax = 1
	par_level_flag[]	FL	cMax = 1
	abs_level_gt1_flag[]	FL	cMax = 1
	abs_level_gt3_flag[]	FL	cMax = 1
	abs_remainder[]	9.5.3.10	cIdx, current sub-block index i, x0, y0
	dec_abs_level[]	9.5.3.11	cIdx, x0, y0, xC, yC, log2TbWidth, log2TbHeight
	coeff_sign_flag[]	FL	cMax = 1
	mts_idx[][][]	FL	cMax = 3

[0213]

[0214]

상술한 실시예에서, 방법들은 일련의 단계 또는 블록으로써 순서도를 기초로 설명되고 있지만, 본 개시의 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당업자라면 순서도에 나타내어진 단계들이 배타적이지 않고, 다른 단계가 포함되거나 순서도의 하나 또는 그 이상의 단계가 본 개시의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.

[0215]

상술한 본 개시에 따른 방법은 소프트웨어 형태로 구현될 수 있으며, 본 개시에 따른 인코딩 장치 및/또는 디코딩 장치는 예를 들어 TV, 컴퓨터, 스마트폰, 셋톱박스, 디스플레이 장치 등의 영상 처리를 수행하는 장치에 포함될 수 있다.

[0216]

본 개시에서 실시예들이 소프트웨어로 구현될 때, 상술한 방법은 상술한 기능을 수행하는 모듈(과정, 기능 등)

로 구현될 수 있다. 모듈은 메모리에 저장되고, 프로세서에 의해 실행될 수 있다. 메모리는 프로세서 내부 또는 외부에 있을 수 있고, 잘 알려진 다양한 수단으로 프로세서와 연결될 수 있다. 프로세서는 ASIC(application-specific integrated circuit), 다른 칩셋, 논리 회로 및/또는 데이터 처리 장치를 포함할 수 있다. 메모리는 ROM(read-only memory), RAM(random access memory), 플래쉬 메모리, 메모리 카드, 저장 매체 및/또는 다른 저장 장치를 포함할 수 있다. 즉, 본 개시에서 설명한 실시예들은 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 예를 들어, 각 도면에서 도시한 기능 유닛들은 컴퓨터, 프로세서, 마이크로 프로세서, 컨트롤러 또는 칩 상에서 구현되어 수행될 수 있다. 이 경우 구현을 위한 정보(ex. information on instructions) 또는 알고리즘이 디지털 저장 매체에 저장될 수 있다.

[0217] 또한, 본 개시가 적용되는 디코딩 장치 및 인코딩 장치는 멀티미디어 방송 송수신 장치, 모바일 통신 단말, 홈 시네마 비디오 장치, 디지털 시네마 비디오 장치, 감시용 카메라, 비디오 대화 장치, 비디오 통신과 같은 실시간 통신 장치, 모바일 스트리밍 장치, 저장 매체, 캠코더, 주문형 비디오(VoD) 서비스 제공 장치, OTT 비디오(Over the top video) 장치, 인터넷 스트리밍 서비스 제공 장치, 3차원(3D) 비디오 장치, VR(virtual reality) 장치, AR(argumente reality) 장치, 화상 전화 비디오 장치, 운송 수단 단말(ex. 차량(자율주행차량 포함) 단말, 비행기 단말, 선박 단말 등) 및 의료용 비디오 장치 등에 포함될 수 있으며, 비디오 신호 또는 데이터 신호를 처리하기 위해 사용될 수 있다. 예를 들어, OTT 비디오(Over the top video) 장치로는 게임 콘솔, 블루레이 플레이어, 인터넷 접속 TV, 홈시어터 시스템, 스마트폰, 태블릿 PC, DVR(Digital Video Recoder) 등을 포함할 수 있다.

[0218] 또한, 본 개시가 적용되는 처리 방법은 컴퓨터로 실행되는 프로그램의 형태로 생산될 수 있으며, 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 본 개시에 따른 데이터 구조를 가지는 멀티미디어 데이터도 또한 컴퓨터가 판독할 수 있는 기록 매체에 저장될 수 있다. 상기 컴퓨터가 판독할 수 있는 기록 매체는 컴퓨터로 읽을 수 있는 데이터가 저장되는 모든 종류의 저장 장치 및 분산 저장 장치를 포함한다. 상기 컴퓨터가 판독할 수 있는 기록 매체는, 예를 들어, 블루레이 디스크(BD), 범용 직렬 버스(USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크 및 광학적 데이터 저장 장치를 포함할 수 있다. 또한, 상기 컴퓨터가 판독할 수 있는 기록 매체는 반송파(예를 들어, 인터넷을 통한 전송)의 형태로 구현된 미디어를 포함한다. 또한, 인코딩 방법으로 생성된 비트스트림이 컴퓨터가 판독할 수 있는 기록 매체에 저장되거나 유무선 통신 네트워크를 통해 전송될 수 있다.

[0219] 또한, 본 개시의 실시예는 프로그램 코드에 의한 컴퓨터 프로그램 제품으로 구현될 수 있고, 상기 프로그램 코드는 본 개시의 실시예에 의해 컴퓨터에서 수행될 수 있다. 상기 프로그램 코드는 컴퓨터에 의해 판독가능한 캐리어 상에 저장될 수 있다.

[0220] 도 13은 본 문서의 개시가 적용될 수 있는 콘텐츠 스트리밍 시스템의 예를 나타낸다.

[0221] 도 13을 참조하면, 본 개시가 적용되는 콘텐츠 스트리밍 시스템은 크게 인코딩 서버, 스트리밍 서버, 웹 서버, 미디어 저장소, 사용자 장치 및 멀티미디어 입력 장치를 포함할 수 있다.

[0222] 상기 인코딩 서버는 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들로부터 입력된 콘텐츠를 디지털 데이터로 압축하여 비트스트림을 생성하고 이를 상기 스트리밍 서버로 전송하는 역할을 한다. 다른 예로, 스마트폰, 카메라, 캠코더 등과 같은 멀티미디어 입력 장치들이 비트스트림을 직접 생성하는 경우, 상기 인코딩 서버는 생략될 수 있다.

[0223] 상기 비트스트림은 본 개시가 적용되는 인코딩 방법 또는 비트스트림 생성 방법에 의해 생성될 수 있고, 상기 스트리밍 서버는 상기 비트스트림을 전송 또는 수신하는 과정에서 일시적으로 상기 비트스트림을 저장할 수 있다.

[0224] 상기 스트리밍 서버는 웹 서버를 통한 사용자 요청에 기초하여 멀티미디어 데이터를 사용자 장치에 전송하고, 상기 웹 서버는 사용자에게 어떠한 서비스가 있는지를 알려주는 매개체 역할을 한다. 사용자가 상기 웹 서버에 원하는 서비스를 요청하면, 상기 웹 서버는 이를 스트리밍 서버에 전달하고, 상기 스트리밍 서버는 사용자에게 멀티미디어 데이터를 전송한다. 이때, 상기 콘텐츠 스트리밍 시스템은 별도의 제어 서버를 포함할 수 있고, 이 경우 상기 제어 서버는 상기 콘텐츠 스트리밍 시스템 내 각 장치 간 명령/응답을 제어하는 역할을 한다.

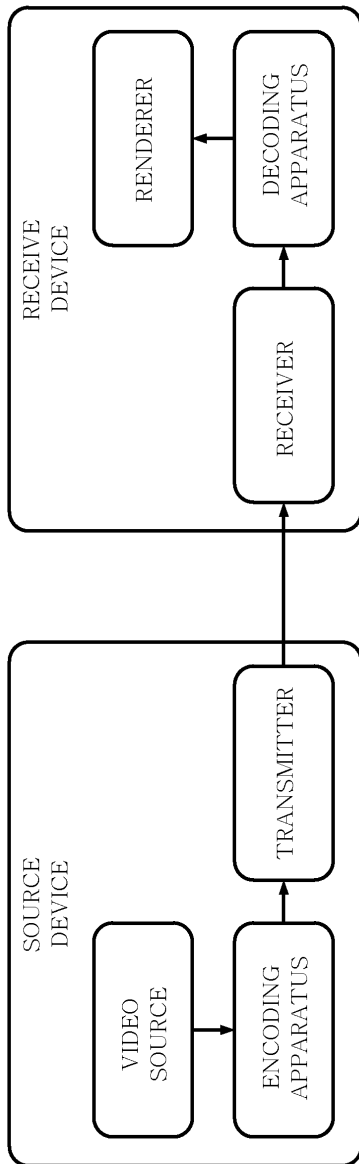
[0225] 상기 스트리밍 서버는 미디어 저장소 및/또는 인코딩 서버로부터 콘텐츠를 수신할 수 있다. 예를 들어, 상기 인코딩 서버로부터 콘텐츠를 수신하게 되는 경우, 상기 콘텐츠를 실시간으로 수신할 수 있다. 이 경우, 원활한 스트리밍 서비스를 제공하기 위하여 상기 스트리밍 서버는 상기 비트스트림을 일정 시간동안 저장할 수 있다.

[0226] 상기 사용자 장치의 예로는, 휴대폰, 스마트 폰(smart phone), 노트북 컴퓨터(laptop computer), 디지털방송용 단말기, PDA(personal digital assistants), PMP(portable multimedia player), 네비게이션, 슬레이트 PC(slate PC), 태블릿 PC(tablet PC), 울트라북(ultrabook), 웨어러블 디바이스(wearable device, 예를 들어, 워치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)), 디지털 TV, 데스크탑 컴퓨터, 디지털 사이니지 등이 있을 수 있다.

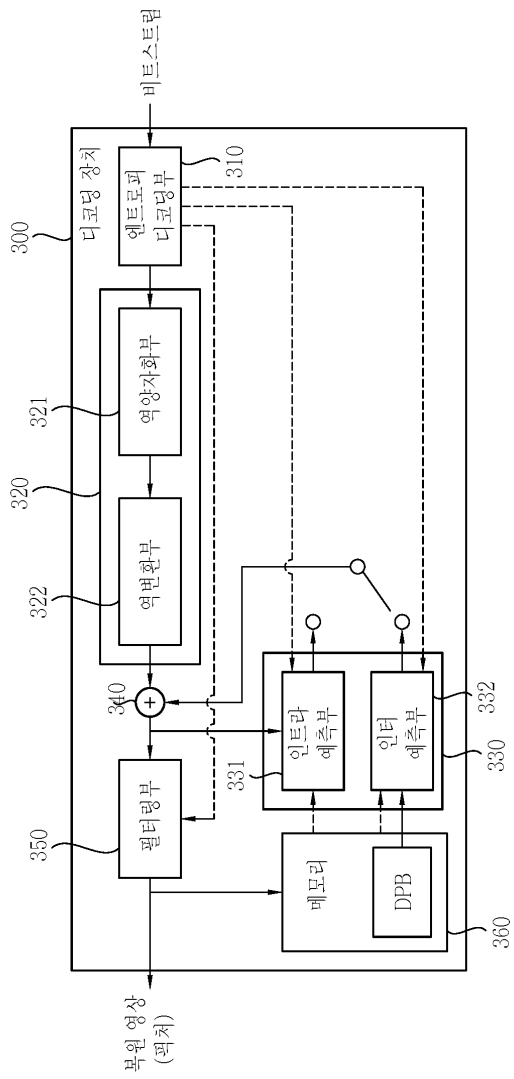
[0227] 상기 콘텐츠 스트리밍 시스템 내 각 서버들은 분산 서버로 운영될 수 있으며, 이 경우 각 서버에서 수신하는 데이터는 분산 처리될 수 있다.

도면

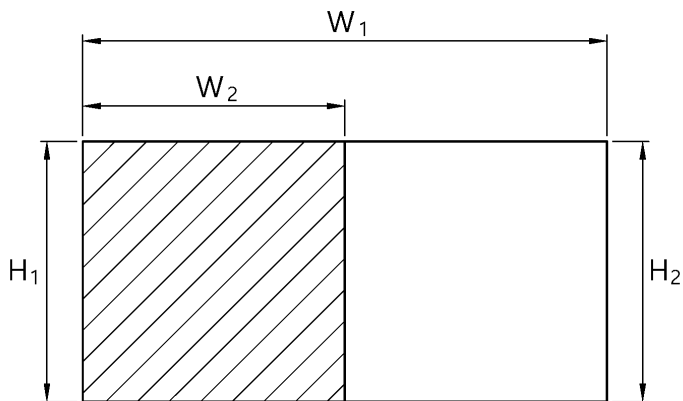
도면1



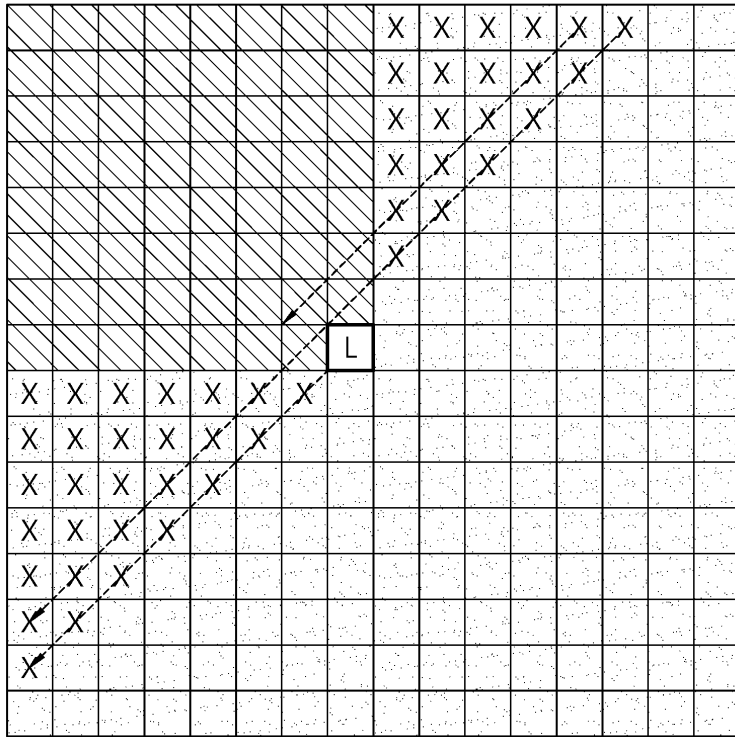
도면3



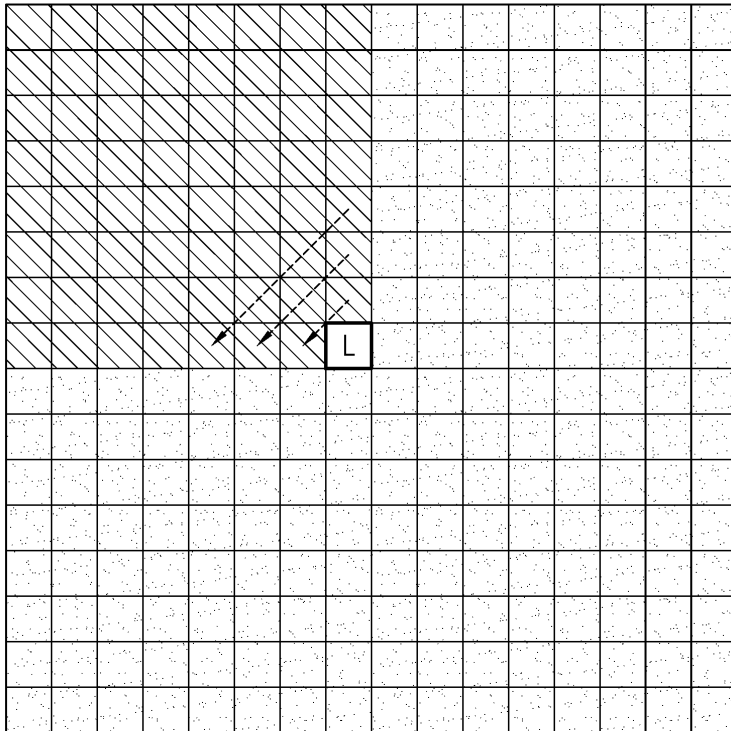
도면4



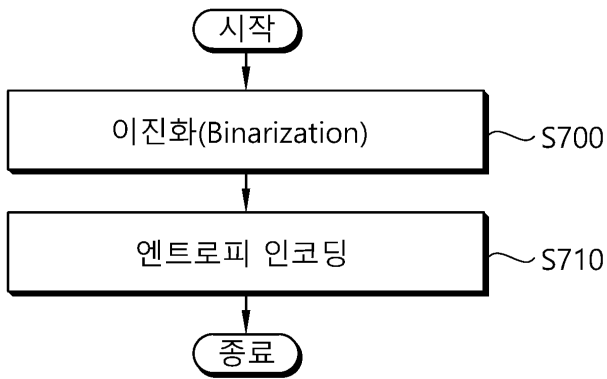
도면5



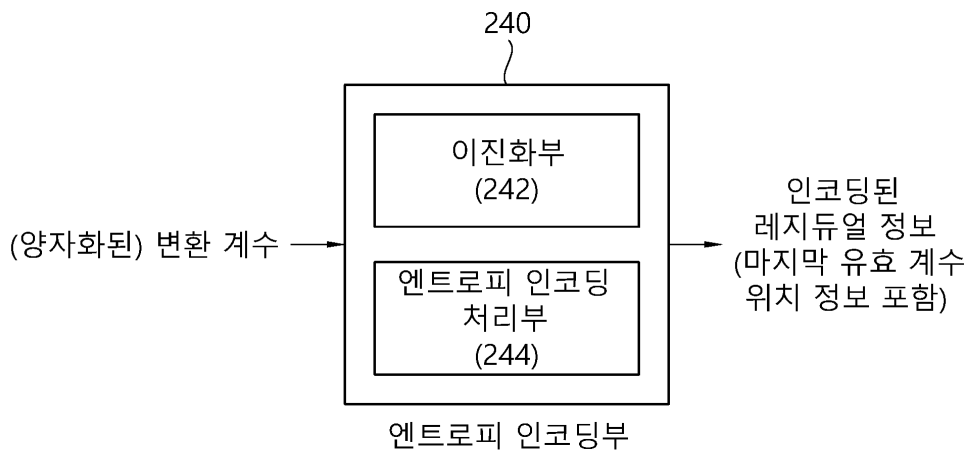
도면6



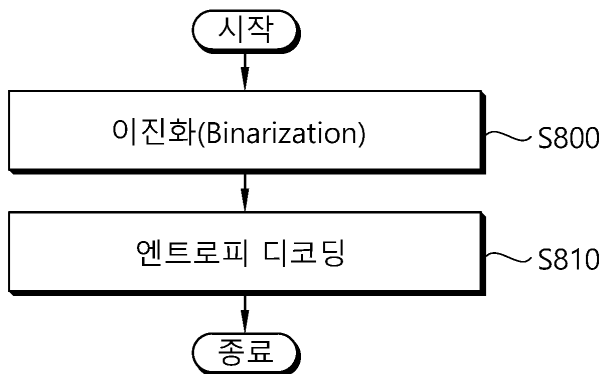
도면7a



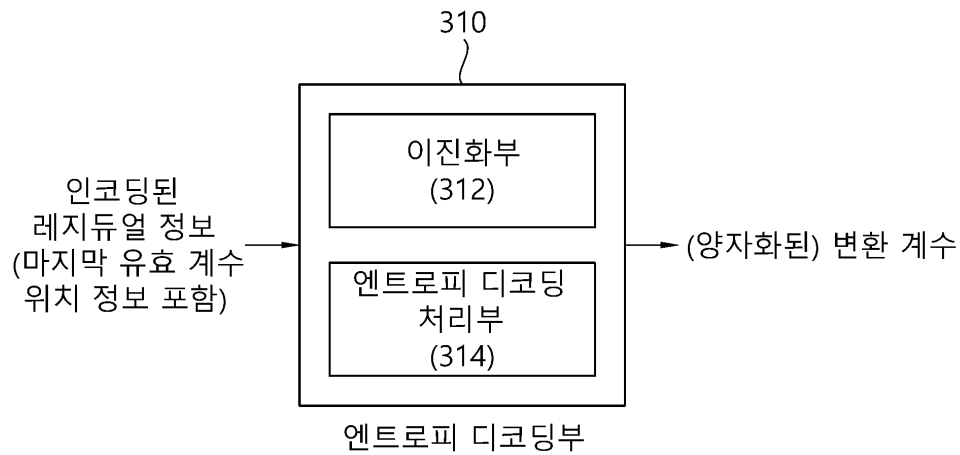
도면7b



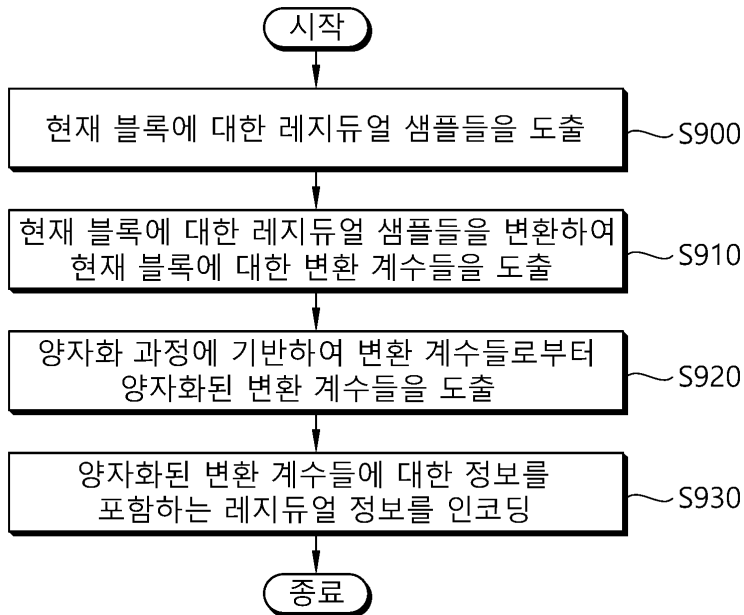
도면8a



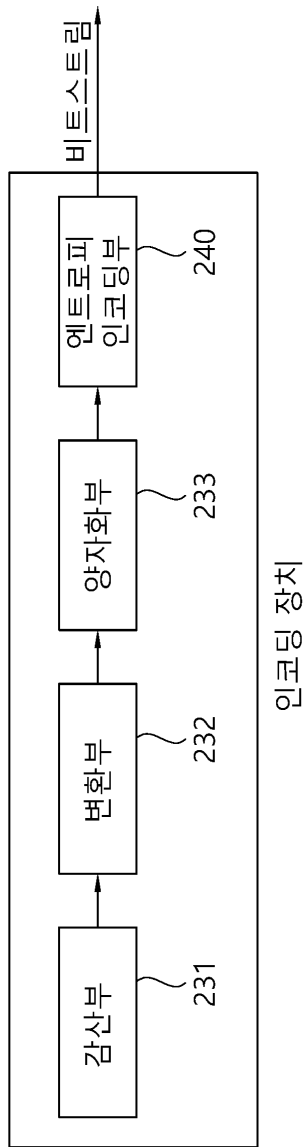
도면8b



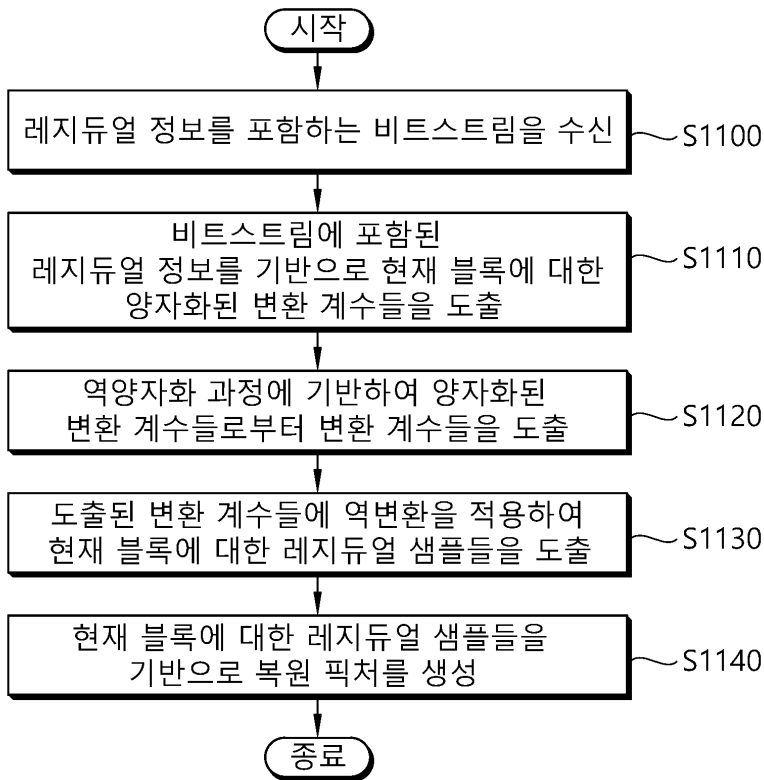
도면9



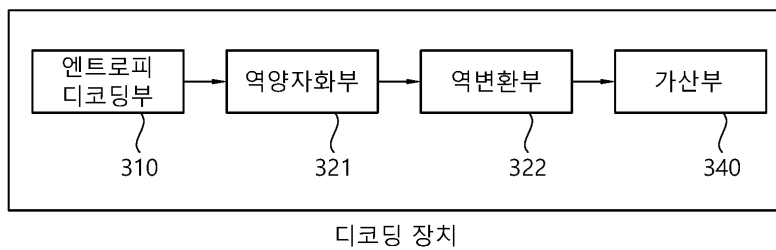
도면10



도면11



도면12



도면13

