

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2025년 1월 9일 (09.01.2025)

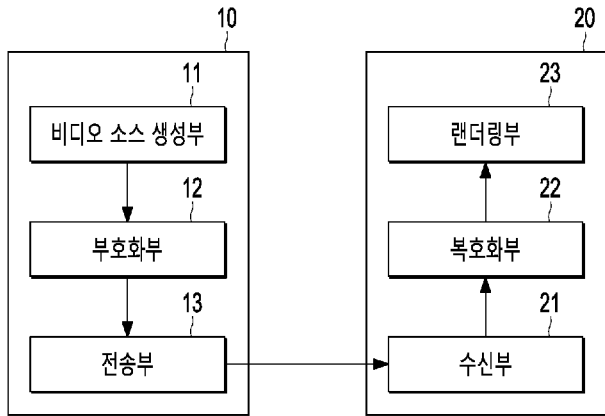


(10) 국제공개번호
WO 2025/009938 A1

- (51) 국제특허분류: H04N 19/117 (2014.01) H04N 19/85 (2014.01)
H04N 19/70 (2014.01) G06N 3/02 (2006.01)
- (21) 국제출원번호: PCT/KR2024/009604
- (22) 국제출원일: 2024년 7월 5일 (05.07.2024)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보: 63/525,146 2023년 7월 5일 (05.07.2023) US
- (71) 출원인: 엘지전자 주식회사 (LG ELECTRONICS INC.) [KR/KR]; 07336 서울시 영등포구 여의대로 128, Seoul (KR).
- (72) 발명자: 탄헨드리 (TAN, Hendry); 06772 서울시 서초구 양재대로11길 19 LG전자 특허센터, Seoul (KR).
- (74) 대리인: 특허법인 에스씨엘 (SCL IP LAW FIRM); 06719 서울시 서초구 남부순환로325길 7, 3층, Seoul (KR).
- (81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.
- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ,

(54) Title: IMAGE ENCODING/DECODING METHOD, RECORDING MEDIUM STORING BITSTREAM, AND METHOD OF TRANSMITTING BITSTREAM

(54) 발명의 명칭: 영상 부호화/복호화 방법, 비트스트림을 저장한 기록 매체 및 비트스트림을 전송하는 방법



- 11 ... Video source generation unit
- 12 ... Encoding unit
- 13 ... Transmission unit
- 21 ... Reception unit
- 22 ... Decoding unit
- 23 ... Rendering unit

(57) Abstract: An image decoding method according to the present disclosure may comprise the steps of: obtaining a neural-network post-filter (NNPF) supplemental enhancement information (SEI) message, the NNPF SEI message including an NNPF characteristics (NNPFC) SEI message and an NNPF activation (NNPFA) SEI message, and determining a neural network that can be used as a post-processing filter on the basis of the NNPFC SEI message; and determining whether to activate a target NNPF applicable to a current picture on the basis of the NNPFA SEI message, wherein, until a picture of a current layer related to an NNPFA SEI message having the same target NNPF identification information as the current SEI message is output after the current picture in an output order, the target NNPF is used to perform post-processing filtering for the current picture and subsequent pictures in the current layer in the output order.

WO 2025/009938 A1

UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

(57) 요약서: 본 개시에 따른 영상 복호화 방법은, NNPF(neural-network post-filter) SEI(supplemental enhancement information) 메시지를 획득하되, 상기 NNPF SEI 메시지는 NNPF(NNPF characteristics) SEI 메시지 및 NNPFA(NNPF activation) SEI 메시지를 포함하고, 상기 NNPF SEI 메시지에 기반하여 후처리 필터로 이용될 수 있는 신경망을 결정하는 단계; 및 상기 NNPFA SEI 메시지에 기반하여, 상기 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 결정하는 단계를 포함하고, 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 NNPFA SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 영상 복호화 방법일 수 있다.

명세서

발명의 명칭: 영상 부호화/복호화 방법, 비트스트림을 저장한 기록 매체 및 비트스트림을 전송하는 방법

기술분야

- [1] 본 개시는 영상 부호화/복호화 방법, 비트스트림을 저장한 기록 매체 및 비트스트림을 전송하는 방법에 관한 것으로서, 신경망 포스트 필터의 지속성 취소 여부를 결정하는 방법에 관한 것이다.

배경기술

- [2] 최근 HD(High Definition) 영상 및 UHD(Ultra High Definition) 영상과 같은 고해상도, 고품질의 영상에 대한 수요가 다양한 분야에서 증가하고 있다. 영상 데이터가 고해상도, 고품질이 될수록 기존의 영상 데이터에 비해 상대적으로 전송되는 정보량 또는 비트량이 증가하게 된다. 전송되는 정보량 또는 비트량의 증가는 전송 비용과 저장 비용의 증가를 초래한다.
- [3] 이에 따라, 고해상도, 고품질 영상의 정보를 효과적으로 전송하거나 저장하고, 재생하기 위한 고효율의 영상 압축 기술이 요구된다.

발명의 상세한 설명

기술적 과제

- [4] 본 개시는 부호화/복호화 효율이 향상된 영상 부호화/복호화 방법 및 장치를 제공하는 것을 목적으로 한다.
- [5] 또한, 본 개시는 동일한 식별 정보를 가지는 둘 이상의 NNPF 활성화가 존재할 때 어느 NNPF의 지속성을 취소할지 명확히 구별하는 방법을 제공하는 것을 목적으로 한다.
- [6] 또한, 본 개시는 동일한 식별 정보를 가지는 둘 이상의 NNPF 활성화가 동시에 이루어지는 것을 방지하는 방법을 제공하는 것을 목적으로 한다.
- [7] 또한, 본 개시는 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체를 제공하는 것을 목적으로 한다.
- [8] 또한, 본 개시는 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법을 제공하는 것을 목적으로 한다.
- [9] 또한, 본 개시는 본 개시에 따른 영상 복호화 장치에 의해 수신되고 복호화되어 영상의 복원에 이용되는 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체를 제공하는 것을 목적으로 한다.
- [10] 본 개시에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제 해결 수단

- [11] 본 개시의 일 양상에 따른 영상 복호화 방법은, 영상 복호화 장치에 의해 수행되는 영상 복호화 방법으로서, NNPF(neural-network post-filter) SEI(supplemental enhancement information) 메시지를 획득하되, 상기 NNPF SEI 메시지는 NNPF(neural-network post-filter characteristics) SEI 메시지 및 NNPFA(neural-network post-filter activation) SEI 메시지를 포함하고, 상기 NNPF SEI 메시지에 기반하여 후처리(post-processing) 필터로 이용될 수 있는 신경망을 결정하는 단계; 및 상기 NNPFA SEI 메시지에 기반하여, 상기 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 결정하는 단계를 포함하고, 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 NNPFA SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 영상 복호화 방법일 수 있다.
- [12] 본 개시의 다른 양상에 따른 영상 부호화 방법은, 영상 부호화 장치에 의해 수행되는 영상 부호화 방법으로서, 후처리(post-processing) 필터로 이용될 수 있는 신경망을 NNPF(neural-network post-filter characteristics) SEI(supplemental enhancement information) 메시지로 부호화하는 단계; 및 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 NNPFA(neural-network post-filter activation) SEI 메시지로 부호화하는 단계를 포함하고, 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 NNPFA SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 영상 부호화 방법일 수 있다.
- [13] 본 개시의 또 다른 양상에 따른 컴퓨터 판독 가능한 기록 매체는, 본 개시의 영상 부호화 방법 또는 장치에 의해 생성된 비트스트림을 저장할 수 있다.
- [14] 본 개시의 또 다른 양상에 따른 전송 방법은, 본 개시의 영상 부호화 방법 또는 장치에 의해 생성된 비트스트림을 전송할 수 있다.
- [15] 본 개시에 대하여 위에서 간략하게 요약된 특징들은 후술하는 본 개시의 상세한 설명의 예시적인 양상일 뿐이며, 본 개시의 범위를 제한하는 것은 아니다.

발명의 효과

- [16] 본 개시에 따르면, 부호화/복호화 효율이 향상된 영상 부호화/복호화 방법 및 장치가 제공될 수 있다.
- [17] 또한, 본 개시에 따르면, 동일한 식별 정보를 가지는 둘 이상의 NNPF 활성화가 존재할 때 지속성을 취소할 NNPF를 명확하게 식별할 수 있다.
- [18] 또한, 본 개시에 따르면, 동일한 식별 정보를 가지는 둘 이상의 NNPF 활성화가 동시에 이루어지는 것을 방지할 수 있다.

- [19] 또한, 본 개시에 따르면, 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 비밀시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [20] 또한, 본 개시에 따르면, 본 개시에 따른 영상 복호화 장치에 의해 수신되고 복호화되어 영상의 복원에 이용되는 비트스트림을 저장하는 비밀시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [21] 또한, 본 개시에 따르면, 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법이 제공될 수 있다.
- [22] 본 개시에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

- [23] 도 1은 본 개시에 따른 실시예가 적용될 수 있는 비디오 코딩 시스템을 개략적으로 나타내는 도면이다.
- [24] 도 2는 본 개시에 따른 실시예가 적용될 수 있는 영상 부호화 장치를 개략적으로 나타내는 도면이다.
- [25] 도 3은 본 개시에 따른 실시예가 적용될 수 있는 영상 복호화 장치를 개략적으로 나타내는 도면이다.
- [26] 도 4는 코딩된 비디오/영상에 대한 계층 구조를 예시적으로 나타낸다.
- [27] 도 5는 루마 채널의 유도를 위한 인터리브(interleaved) 방식을 설명하기 위한 도면이다.
- [28] 도 6은 영상 부호화 장치의 영상 부호화 방법을 설명하기 위한 도면이다.
- [29] 도 7은 영상 복호화 장치의 영상 복호화 방법을 설명하기 위한 도면이다.
- [30] 도 8은 대상 NNPF의 활성화 여부를 결정하는 실시예에 대한 영상 복호화 방법을 설명하기 위한 도면이다.
- [31] 도 9 내지 도 11은 대상 NNPF의 지속성 취소 여부에 대한 다양한 실시예들을 설명하기 위한 도면이다.
- [32] 도 12는 본 개시에 따른 실시예가 적용될 수 있는 콘텐츠 스트리밍 시스템을 예시적으로 나타내는 도면이다.

발명의 실시를 위한 형태

- [33] 이하에서는 첨부한 도면을 참고로 하여 본 개시의 실시예에 대하여 본 개시가 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나, 본 개시는 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.
- [34] 본 개시의 실시예를 설명함에 있어서 공지 구성 또는 기능에 대한 구체적인 설명이 본 개시의 요지를 흐릴 수 있다고 판단되는 경우에는 그에 대한 상세한 설명은 생략한다. 그리고, 도면에서 본 개시에 대한 설명과 관계없는 부분은

생략하였으며, 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

- [35] 본 개시에 있어서, 어떤 구성요소가 다른 구성요소와 "연결", "결합" 또는 "접속"되어 있다고 할 때, 이는 직접적인 연결관계뿐만 아니라, 그 중간에 또 다른 구성요소가 존재하는 간접적인 연결관계도 포함할 수 있다. 또한 어떤 구성요소가 다른 구성요소를 "포함한다" 또는 "가진다"고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 배제하는 것이 아니라 또 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [36] 본 개시에 있어서, 제1, 제2 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용되며, 특별히 언급되지 않는 한 구성요소들 간의 순서 또는 중요도 등을 한정하지 않는다. 따라서, 본 개시의 범위 내에서 일 실시예에서의 제1 구성요소는 다른 실시예에서 제2 구성요소라고 칭할 수도 있고, 마찬가지로 일 실시예에서의 제2 구성요소를 다른 실시예에서 제1 구성요소라고 칭할 수도 있다.
- [37] 본 개시에 있어서, 서로 구별되는 구성요소들은 각각의 특징을 명확하게 설명하기 위함이며, 구성요소들이 반드시 분리되는 것을 의미하지는 않는다. 즉, 복수의 구성요소가 통합되어 하나의 하드웨어 또는 소프트웨어 단위로 이루어질 수도 있고, 하나의 구성요소가 분산되어 복수의 하드웨어 또는 소프트웨어 단위로 이루어질 수도 있다. 따라서, 별도로 언급하지 않더라도 이와 같이 통합된 또는 분산된 실시예도 본 개시의 범위에 포함된다.
- [38] 본 개시에 있어서, 다양한 실시예에서 설명하는 구성요소들이 반드시 필수적인 구성요소들은 의미하는 것은 아니며, 일부는 선택적인 구성요소일 수 있다. 따라서, 일 실시예에서 설명하는 구성요소들의 부분집합으로 구성되는 실시예도 본 개시의 범위에 포함된다. 또한, 다양한 실시예에서 설명하는 구성요소들에 추가적으로 다른 구성요소를 포함하는 실시예도 본 개시의 범위에 포함된다.
- [39] 본 개시는 영상의 부호화 및 복호화에 관한 것이다. 예를 들어 이 문서에서 개시된 방법 및 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.
- [40] 본 개시는 비디오/영상 코딩에 관한 다양한 실시예들을 제시하며, 다른 언급이 없는 한 상기 실시예들은 서로 조합되어 수행될 수도 있다.
- [41] 본 개시에서 사용되는 용어는, 본 개시에서 새롭게 정의되지 않는 한 본 개시가 속한 기술 분야에서 통용되는 통상의 의미를 가질 수 있다.
- [42] 본 개시에서 "픽처(picture)"는 일반적으로 특정 시간대의 하나의 영상을 나타내는 단위를 의미하며, 슬라이스(slice)/타일(tile)은 픽처의 일부를 구성하는 부호화 단위로서, 하나의 픽처는 하나 이상의 슬라이스/타일로 구성될 수 있다. 또한, 슬라이스/타일은 하나 이상의 CTU(coding tree unit)를 포함할 수 있다.

하나의 픽처는 하나 이상의 타일 그룹으로 구성될 수 있다. 하나의 타일 그룹은 하나 이상의 타일들을 포함할 수 있다. 브릭은 픽처 내 타일의 CTU 행들의 사각 영역을 나타낼 수 있다. 이 문서에서 타일 그룹과 슬라이스는 혼용될 수 있다. 예를 들어 본 문서에서 tile group/tile group header는 slice/slice header로 불릴 수 있다.

- [43] 본 개시에서 "픽셀(pixel)" 또는 "펠(pel)"은 하나의 픽처(또는 영상)를 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/픽셀값만을 나타낼 수도 있다.
- [44] 본 개시에서 "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(예를 들어, cb, cr) 블록을 포함할 수 있다. 유닛은 경우에 따라서 "샘플 어레이", "블록(block)" 또는 "영역(area)" 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, $M \times N$ 블록은 M 개의 열과 N 개의 행으로 이루어진 샘플들(또는 샘플 어레이) 또는 변환 계수(transform coefficient)들의 집합(또는 어레이)을 포함할 수 있다.
- [45] 본 개시에서 "현재 블록"은 "현재 코딩 블록", "현재 코딩 유닛", "부호화 대상 블록", "복호화 대상 블록" 또는 "처리 대상 블록" 중 하나를 의미할 수 있다. 예측이 수행되는 경우, "현재 블록"은 "현재 예측 블록" 또는 "예측 대상 블록"을 의미할 수 있다. 변환(역변환)/양자화(역양자화)가 수행되는 경우, "현재 블록"은 "현재 변환 블록" 또는 "변환 대상 블록"을 의미할 수 있다. 필터링이 수행되는 경우, "현재 블록"은 "필터링 대상 블록"을 의미할 수 있다.
- [46] 본 개시에서 "현재 블록"은 크로마 블록이라는 명시적인 기제가 없는 한 루마 성분 블록과 크로마 성분 블록을 모두 포함하는 블록 또는 "현재 블록의 루마 블록"을 의미할 수 있다. 현재 블록의 루마 성분 블록은 명시적으로 "루마 블록" 또는 "현재 루마 블록"과 같이 루마 성분 블록이라는 명시적인 기제를 포함하여 표현될 수 있다. 또한, 현재 블록의 크로마 성분 블록은 명시적으로 "크로마 블록" 또는 "현재 크로마 블록"과 같이 크로마 성분 블록이라는 명시적인 기제를 포함하여 표현될 수 있다.
- [47] 본 개시에서 "/"와 ","는 "및/또는"으로 해석될 수 있다. 예를 들어, "A/B"와 "A, B"는 "A 및/또는 B"로 해석될 수 있다. 또한, "A/B/C"와 "A, B, C"는 "A, B 및/또는 C 중 적어도 하나"를 의미할 수 있다.
- [48] 본 개시에서 "또는"은 "및/또는"으로 해석될 수 있다. 예를 들어, "A 또는 B"는, 1) "A" 만을 의미하거나 2) "B" 만을 의미하거나, 3) "A 및 B"를 의미할 수 있다. 또는, 본 개시에서 "또는"은 "추가적으로 또는 대체적으로(Additionally or alternatively)"를 의미할 수 있다.
- [49] 비디오 코딩 시스템 개요

- [50] 도 1은 본 개시에 따른 실시예가 적용될 수 있는 비디오 코딩 시스템을 개략적으로 나타내는 도면이다.
- [51] 일 실시예에 따른 비디오 코딩 시스템은 부호화 장치(10) 및 복호화 장치(20)를 포함할 수 있다. 부호화 장치(10)는 부호화된 비디오(video) 및/또는 영상(image) 정보 또는 데이터를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 복호화 장치(20)로 전달할 수 있다.
- [52] 일 실시예에 따른 부호화 장치(10)는 비디오 소스 생성부(11), 부호화부(12), 전송부(13)를 포함할 수 있다. 일 실시예에 따른 복호화 장치(20)는 수신부(21), 복호화부(22) 및 렌더링부(23)를 포함할 수 있다. 상기 부호화부(12)는 비디오/영상 부호화부라고 불릴 수 있고, 상기 복호화부(22)는 비디오/영상 복호화부라고 불릴 수 있다. 전송부(13)는 부호화부(12)에 포함될 수 있다. 수신부(21)는 복호화부(22)에 포함될 수 있다. 렌더링부(23)는 디스플레이부를 포함할 수도 있고, 디스플레이부는 별개의 디바이스 또는 외부 컴포넌트로 구성될 수도 있다.
- [53] 비디오 소스 생성부(11)는 비디오/영상의 캡처, 합성 또는 생성 과정 등을 통하여 비디오/영상을 획득할 수 있다. 비디오 소스 생성부(11)는 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 갈음될 수 있다.
- [54] 부호화부(12)는 입력 비디오/영상을 부호화할 수 있다. 부호화부(12)는 압축 및 부호화 효율을 위하여 예측, 변환, 양자화 등 일련의 절차를 수행할 수 있다. 부호화부(12)는 부호화된 데이터(부호화된 비디오/영상 정보)를 비트스트림(bitstream) 형태로 출력할 수 있다.
- [55] 전송부(13)는 비트스트림 형태로 출력된 부호화된 비디오/영상 정보 또는 데이터를 획득할 수 있으며, 이를 파일 또는 스트리밍 형태로 디지털 저장매체 또는 네트워크를 통하여 복호화 장치(20)의 수신부(21) 또는 다른 외부 객체로 전달할 수 있다. 디지털 저장 매체는 USB, SD, CD, DVD, 블루레이, HDD, SSD 등 다양한 저장 매체를 포함할 수 있다. 전송부(13)는 미리 정해진 파일 포맷을 통하여 미디어 파일을 생성하기 위한 엘리먼트를 포함할 수 있고, 방송/통신 네트워크를 통한 전송을 위한 엘리먼트를 포함할 수 있다. 전송부(13)는 부호화 장치(12)와는 별개의 전송 장치로 구비될 수 있으며, 이 경우 전송 장치는 비트스트림 형태로 출력된 부호화된 비디오/영상 정보 또는 데이터를 획득하는 적어도 하나의 프로세서와 이를 파일 또는 스트리밍 형태로 전달하는 전송부를 포함할 수 있다. 수신부(21)는 상기 저장매체 또는 네트워크로부터 상기

- 비트스트림을 추출/수신하여 복호화부(22)로 전달할 수 있다.
- [56] 복호화부(22)는 부호화부(12)의 동작에 대응하는 역양자화, 역변환, 예측 등 일련의 절차를 수행하여 비디오/영상을 복호화할 수 있다.
- [57] 렌더링부(23)는 복호화된 비디오/영상을 렌더링할 수 있다. 렌더링된 비디오/영상은 디스플레이부를 통하여 디스플레이될 수 있다.
- [58] 영상 부호화 장치 개요
- [59] 도 2는 본 개시에 따른 실시예가 적용될 수 있는 영상 부호화 장치를 개략적으로 도시한 도면이다.
- [60] 도 2에 도시된 바와 같이, 영상 부호화 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)은 감산부(subtractor, 231)를 더 포함할 수 있다. 가산부(250)는 복원부(reconstructor) 또는 복원 블록 생성부(reconstructed 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(Dcoded Picture Buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구성될 수도 있다. 상기 하드웨어 컴포넌트는 메모리(270)를 내/외부 컴포넌트로 더 포함할 수도 있다.
- [61] 영상 분할부(210)는 영상 부호화 장치(200)에 입력된 입력 영상(또는, 픽처, 프레임)을 하나 이상의 처리 유닛(processing unit)으로 분할할 수 있다. 일 예로, 상기 처리 유닛은 코딩 유닛(coding unit, CU)이라고 불릴 수 있다. 코딩 유닛은 코딩 트리 유닛(coding tree unit, CTU) 또는 최대 코딩 유닛(largest coding unit, LCU)을 QT/BT/TT (Quad-tree/binary-tree/ternary-tree) 구조에 따라 재귀적으로(recursively) 분할함으로써 획득될 수 있다. 예를 들어, 하나의 코딩 유닛은 쿼드 트리 구조, 바이너리 트리 구조 및/또는 터너리 트리 구조를 기반으로 하위(deeper) 탭스의 복수의 코딩 유닛들로 분할될 수 있다. 코딩 유닛의 분할을 위해, 쿼드 트리 구조가 먼저 적용되고 바이너리 트리 구조 및/또는 터너리 트리 구조가 나중에 적용될 수 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 수도 있고, 필요에 따라서는 최대 코딩 유닛이 재귀적으로(recursively) 보다 하위 탭스의 코딩 유닛들로 분할되어 최적의 사이즈의 코딩 유닛이 최종 코딩 유닛으로 사용될 수도 있다. 여기서 코딩

절차라 함은 후술하는 예측, 변환 및/또는 복원 등의 절차를 포함할 수 있다. 다른 예로, 상기 코딩 절차의 처리 유닛은 예측 유닛(PU: Prediction Unit) 또는 변환 유닛(TU: Transform Unit)일 수 있다. 상기 예측 유닛 및 상기 변환 유닛은 각각 상기 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.

- [62] 영상 부호화 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 영상 부호화 장치(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. 예측부(220)는 처리 대상 블록(현재 블록)에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부(220)는 현재 블록 또는 CU 단위로 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있다. 예측부(220)는 현재 블록의 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.
- [63] 인트라 예측부(222)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 상기 참조되는 샘플들은 인트라 예측 모드 및/또는 인트라 예측 기법에 따라 상기 현재 블록의 주변(neighbor)에 위치할 수 있고, 또는 떨어져서 위치할 수도 있다. 인트라 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.
- [64] 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 기반하여 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 경우, 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, 서로 다를 수도 있다. 상기 시간적 주변 블록은 동일 위치 참조 블록(collocated reference block), 동일 위치 CU(colCU) 등의 이름으로 불릴 수 있다. 상기 시간적 주변 블록을 포함하는 참조 픽처는 동일 위치 픽처(collocated picture, colPic)라고 불릴 수 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) 및 움직임 벡터 예측자에 대한 지시자(indicator)를 부호화함으로써 현재 블록의 움직임 벡터를 시그널링할 수 있다. 움직임 벡터 차분은 현재 블록의 움직임 벡터와 움직임 벡터 예측자 간의 차이를 의미할 수 있다.

- [65] 예측부(220)는 후술하는 다양한 예측 방법 및/또는 예측 기법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부(220)는 현재 블록의 예측을 위해 인트라 예측 또는 인터 예측을 적용할 수 있을 뿐 아니라, 인트라 예측과 인터 예측을 동시에 적용할 수 있다. 현재 블록의 예측을 위해 인트라 예측과 인터 예측을 동시에 적용하는 예측 방법은 combined inter and intra prediction (CIIP)라고 불릴 수 있다. 또한, 예측부(220)는 현재 블록의 예측을 위해 인트라 블록 카피(intra block copy, IBC)를 수행할 수도 있다. 인트라 블록 카피는 예를 들어 SCC(screen content coding) 등과 같이 게임 등의 콘텐츠 영상/동영상 코딩을 위하여 사용될 수 있다. IBC는 현재 블록으로부터 소정의 거리만큼 떨어진 위치의 현재 픽처 내 기록된 참조 블록을 이용하여 현재 블록을 예측하는 방법이다. IBC가 적용되는 경우, 현재 픽처 내 참조 블록의 위치는 상기 소정의 거리에 해당하는 벡터(블록 벡터)로서 부호화될 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 개시에서 설명되는 인터 예측 기법들 중 적어도 하나를 이용할 수 있다.

- [66] 예측부(220)를 통해 생성된 예측 신호는 복원 신호를 생성하기 위해 이용되거나 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 감산부(231)는 입력 영상 신호(원본 블록, 원본 샘플 어레이)로부터 예측부(220)에서 출력된 예측 신호(예측된 블록, 예측 샘플 어레이)를 감산하여 레지듀얼 신호(residual signal,

잔여 블록, 잔여 샘플 어레이)를 생성할 수 있다. 생성된 레지듀얼 신호는 변환부(232)로 전송될 수 있다.

- [67] 변환부(232)는 레지듀얼 신호에 변환 기법을 적용하여 변환 계수들(transform coefficients)을 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)을 이용하여 예측 신호를 생성하고 그에 기반하여 획득되는 변환을 의미한다. 변환 과정은 정사각형의 동일한 크기를 갖는 픽셀 블록에 적용될 수도 있고, 정사각형이 아닌 가변 크기의 블록에도 적용될 수 있다.
- [68] 양자화부(233)는 변환 계수들을 양자화하여 엔트로피 인코딩부(240)로 전송할 수 있다. 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.
- [69] 엔트로피 인코딩부(240)는 예를 들어 지수 곱셈(exponential Golomb), CAVLC(context-adaptive variable length coding), CABAC(context-adaptive binary arithmetic coding) 등과 같은 다양한 인코딩 방법을 수행할 수 있다. 엔트로피 인코딩부(190)는 양자화된 변환 계수들 외 비디오/이미지 복원에 필요한 정보들(예를 들어 신택스 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)를 더 포함할 수 있다. 본 개시에서 언급된 시그널링되는 정보, 전송되는 정보 및/또는 신택스 요소들은 비디오/영상 정보에 포함될 수 있다. 상기 비디오/영상 정보는 상술한 인코딩 절차를 통하여 인코딩되어 상기 비트스트림에 포함될 수 있다.
- [70] 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 신호를 전송하는 전송부(미도시) 및/또는 저장하는 저장부(미도시)가 영상

부호화 장치(200)의 내/외부 엘리먼트로서 구비될 수 있고, 또는 전송부는 엔트로피 인코딩부(240)의 구성요소로서 구비될 수도 있다.

- [71] 양자화부(233)로부터 출력된 양자화된 변환 계수들은 레지듀얼 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.
- [72] 가산부(250)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)를 생성할 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.
- [73] 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(170)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. 필터링에 관한 정보는 엔트로피 인코딩부(240)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.
- [74] 메모리(270)에 전송된 수정된 복원 픽처는 인터 예측부(221)에서 참조 픽처로 사용될 수 있다. 영상 부호화 장치(200)는 이를 통하여 인터 예측이 적용되는 경우, 영상 부호화 장치(200)와 영상 복호화 장치에서의 예측 미스매치를 피할 수 있고, 부호화 효율도 향상시킬 수 있다.
- [75] 메모리(270) 내 DPB는 인터 예측부(221)에서의 참조 픽처로 사용하기 위해 수정된 복원 픽처를 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 전달될 수 있다. 메모리(270)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(222)에 전달할 수 있다.
- [76] **영상 복호화 장치 개요**
- [77] 도 3은 본 개시에 따른 실시예가 적용될 수 있는 영상 복호화 장치를 개략적으로 도시한 도면이다.

- [78] 도 3에 도시된 바와 같이, 영상 복호화 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(332) 및 인트라 예측부(331)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. 비디오/영상 정보를 포함하는 비트스트림이 입력되면 영상 복호화 장치(300)는 도 2의 영상 부호화 장치(200)에서 수행된 프로세스에 대응하는 프로세스를 수행하여 영상을 복원할 수 있다. 예를 들어, 영상 복호화 장치(300)는 영상 부호화 장치(200)에서 적용된 처리 유닛을 이용하여 디코딩을 수행할 수 있다. 따라서 디코딩의 처리 유닛은 예를 들어 코딩 유닛일 수 있다. 코딩 유닛은 코딩 트리 유닛이거나 또는 최대 코딩 유닛을 쿼드 트리 구조, 바이너리 트리 구조 및/또는 터너리 트리 구조를 따라서 분할하여 획득할 수 있다. 그리고, 영상 복호화 장치(300)를 통해 디코딩 및 출력된 복원 영상 신호는 재생 장치(미도시)를 통해 재생될 수 있다.
- [79] 영상 복호화 장치(300)는 도 2의 영상 부호화 장치(200)로부터 출력된 신호를 비트스트림 형태로 수신할 수 있다. 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(예를 들어, 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)를 더 포함할 수 있다. 영상 복호화 장치(300)는 영상을 디코딩하기 위해 상기 파라미터 세트에 관한 정보 및/또는 상기 일반 제한 정보를 추가적으로 이용할 수 있다. 본 개시에서 언급된 시그널링 정보, 수신되는 정보 및/또는 신택스 요소들은 상기 디코딩 절차를 통하여 디코딩됨으로써 상기 비트스트림으로부터 획득될 수 있다. 예를 들어, 엔트로피 디코딩부(310)는 지수 콜롬 부호화, CAVLC 또는 CABAC 등의 코딩 방법을 기초로 비트스트림 내 정보를 디코딩하고, 영상 복원에 필요한 신택스 엘리먼트의 값, 레지듀얼에 관한 변환 계수의 양자화된 값들을 출력할 수 있다. 보다 상세하게, CABAC 엔트로피 디코딩 방법은, 비트스트림에서 각 구문 요소에 해당하는 빈을 수신하고, 디코딩 대상 구문 요소 정보와 주변 블록 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 구문 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(330)(인터 예측부(332) 및 인트라 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. 또한, 엔트로피 디코딩부(310)에서 디코딩된 정보 중 필터링에 관한 정보는 필터링부(350)로 제공될 수 있다. 한편, 영상 부호화 장치(200)로부터 출력된 신호를 수신하는 수신부(미도시)가 영상 복호화 장치(300)의 내/외부 엘리먼트로서 추가적으로 구비될 수 있고, 또는 수신부는 엔트로피 디코딩부(310)의 구성요소로서 구비될 수도 있다.

- [80] 한편, 본 개시에 따른 영상 복호화 장치(300)는 비디오/영상/픽처 복호화 장치라고 불릴 수 있다. 상기 영상 복호화 장치(300)는 정보 디코더(비디오/영상/픽처 정보 디코더) 및/또는 샘플 디코더(비디오/영상/픽처 샘플 디코더)를 포함할 수도 있다. 상기 정보 디코더는 엔트로피 디코딩부(310)를 포함할 수 있고, 상기 샘플 디코더는 역양자화부(321), 역변환부(322), 가산부(340), 필터링부(350), 메모리(360), 인터 예측부(332) 및 인트라 예측부(331) 중 적어도 하나를 포함할 수 있다.
- [81] 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 영상 부호화 장치(200)에서 수행된 계수 스캔 순서에 기반하여 수행될 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.
- [82] 역변환부(322)에서는 변환 계수들을 역변환하여 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플 어레이)를 획득할 수 있다.
- [83] 예측부(330)는 현재 블록에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부(330)는 엔트로피 디코딩부(310)로부터 출력된 상기 예측에 관한 정보를 기반으로 상기 현재 블록에 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있고, 구체적인 인트라/인터 예측 모드(예측 기법)를 결정할 수 있다.
- [84] 예측부(330)가 후술하는 다양한 예측 방법(기법)을 기반으로 예측 신호를 생성할 수 있음은 영상 부호화 장치(200)의 예측부(220)에 대한 설명에서 언급된

바와 동일하다.

- [85] 인트라 예측부(331)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 인트라 예측부(222)에 대한 설명은 인트라 예측부(331)에 대해서도 동일하게 적용될 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.
- [86] 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 기반하여 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 모드(기법)를 기반으로 인터 예측이 수행될 수 있으며, 상기 예측에 관한 정보는 상기 현재 블록에 대한 인터 예측의 모드(기법)를 지시하는 정보를 포함할 수 있다.
- [87] 가산부(340)는 획득된 레지듀얼 신호를 예측부(330)(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. 가산부(250)에 대한 설명은 가산부(340)에 대해서도 동일하게 적용될 수 있다. 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 다음 픽처의 인터 예측을 위하여 사용될 수도 있다.
- [88] 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 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.

- [89] 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(332)에 전달할 수 있다. 메모리(360)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(331)에 전달할 수 있다.
- [90] 본 명세서에서, 영상 부호화 장치(200)의 필터링부(260), 인터 예측부(221) 및 인트라 예측부(222)에서 설명된 실시예들은 각각 영상 복호화 장치(300)의 필터링부(350), 인터 예측부(332) 및 인트라 예측부(331)에도 동일 또는 대응되도록 적용될 수 있다.
- [91] 도 4는 코딩된 비디오/영상에 대한 계층 구조를 예시적으로 나타낸다.
- [92] 도 4를 참조하면, 코딩된 비디오/영상은 비디오/영상의 디코딩 처리 및 그 자체를 다루는 VCL(Video Coding Layer, 비디오 코딩 계층), 부호화된 정보를 전송하고 저장하는 하위 시스템, 그리고 VCL과 하위 시스템 사이에 존재하며 네트워크 적응 기능을 담당하는 NAL(Network Abstraction Layer, 네트워크 추상 계층)로 구분되어 있다.
- [93] VCL에서는 압축된 영상 데이터(슬라이스 데이터)를 포함하는 VCL 데이터를 생성하거나, 혹은 픽처 파라미터 세트(Picture Parameter Set: PPS), 시퀀스 파라미터 세트(Sequence Parameter Set: SPS), 비디오 파라미터 세트(Video Parameter Set: VPS) 등의 정보를 포함하는 파라미터 세트 또는 영상의 디코딩 과정에 부가적으로 필요한 SEI(Supplemental Enhancement Information) 메시지를 생성할 수 있다.
- [94] NAL에서는 VCL에서 생성된 RBSP(Raw Byte Sequence Payload)에 헤더 정보(NAL 유닛 헤더)를 추가하여 NAL 유닛을 생성할 수 있다. 이때, RBSP는 VCL에서 생성된 슬라이스 데이터, 파라미터 세트, SEI 메시지 등을 말한다. NAL 유닛 헤더에는 해당 NAL 유닛에 포함되는 RBSP 데이터에 따라 특정되는 NAL 유닛 타입 정보를 포함할 수 있다.
- [95] 도 4에 도시된 바와 같이, NAL 유닛은 VCL에서 생성된 RBSP의 따라 VCL NAL 유닛과 Non-VCL NAL 유닛으로 구분될 수 있다. VCL NAL 유닛은 영상에 대한 정보(슬라이스 데이터)를 포함하고 있는 NAL 유닛을 의미할 수 있고, Non-VCL NAL 유닛은 영상을 디코딩하기 위하여 필요한 정보(파라미터 세트 또는 SEI 메시지)를 포함하고 있는 NAL 유닛을 의미할 수 있다.
- [96] 상술한 VCL NAL 유닛, Non-VCL NAL 유닛은 하위 시스템의 데이터 규격에 따라 헤더 정보를 붙여서 네트워크를 통해 전송될 수 있다. 예컨대, NAL 유닛은 H.266/VVC 파일 포맷, RTP(Real-time Transport Protocol), TS(Transport Stream) 등과 같은 소정 규격의 데이터 형태로 변형되어 다양한 네트워크를 통해 전송될

수 있다.

- [97] 상술한 바와 같이, NAL 유닛은 해당 NAL 유닛에 포함되는 RBSP 데이터 구조(structure)에 따라 NAL 유닛 타입이 특정될 수 있으며, 이러한 NAL 유닛 타입에 대한 정보는 NAL 유닛 헤더에 저장되어 시그널링될 수 있다.
- [98] 예를 들어, NAL 유닛이 영상에 대한 정보(슬라이스 데이터)를 포함하는지 여부에 따라 크게 VCL NAL 유닛 타입과 Non-VCL NAL 유닛 타입으로 분류될 수 있다. VCL NAL 유닛 타입은 VCL NAL 유닛이 포함하는 픽처의 성질 및 종류 등에 따라 분류될 수 있으며, Non-VCL NAL 유닛 타입은 파라미터 세트의 종류 등에 따라 분류될 수 있다.
- [99] 아래는 Non-VCL NAL 유닛 타입이 포함하는 파라미터 세트의 종류 등에 따라 특정된 NAL 유닛 타입의 일예이다.
- [100] - APS (Adaptation Parameter Set) NAL unit: APS를 포함하는 NAL 유닛에 대한 타입
- [101] - DPS(Decoding Parameter Set) NAL unit: DPS를 포함하는 NAL 유닛에 대한 타입
- [102] - VPS(Video Parameter Set) NAL unit: VPS를 포함하는 NAL 유닛에 대한 타입
- [103] - SPS(Sequence Parameter Set) NAL unit: SPS를 포함하는 NAL 유닛에 대한 타입
- [104] - PPS(Picture Parameter Set) NAL unit: PPS를 포함하는 NAL 유닛에 대한 타입
- [105] 상술한 NAL 유닛 타입들은 NAL 유닛 타입을 위한 선택스 정보를 가지며, 상기 선택스 정보는 NAL 유닛 헤더에 저장되어 시그널링될 수 있다. 예컨대, 상기 선택스 정보는 `nal_unit_type`일 수 있으며, NAL 유닛 타입들은 `nal_unit_type` 값으로 특정될 수 있다.
- [106] 슬라이스 헤더(슬라이스 헤더 선택스, 슬라이스 헤더 정보)는 상기 슬라이스에 공통적으로 적용할 수 있는 정보/파라미터를 포함할 수 있다. 상기 APS(APS 선택스) 또는 PPS(PPS 선택스)는 하나 이상의 슬라이스 또는 픽처에 공통적으로 적용할 수 있는 정보/파라미터를 포함할 수 있다. 상기 SPS(SPS 선택스)는 하나 이상의 시퀀스에 공통적으로 적용할 수 있는 정보/파라미터를 포함할 수 있다. 상기 VPS(VPS 선택스)는 다중 레이어에 공통적으로 적용할 수 있는 정보/파라미터를 포함할 수 있다. 상기 DPS(DPS 선택스)는 비디오 전반에 공통적으로 적용할 수 있는 정보/파라미터를 포함할 수 있다. 상기 DPS는 CVS(coded video sequence)의 concatenation에 관련된 정보/파라미터를 포함할 수 있다. 본 개시에서 상위 레벨 선택스(High Level Syntax, HLS)라 함은 상기 APS 선택스, PPS 선택스, SPS 선택스, VPS 선택스, DPS 선택스, 또는 슬라이스 헤더 선택스 중 적어도 하나를 포함할 수 있다.
- [107] 본 개시에서 영상 부호화 장치에서 부호화되어 비트스트림 형태로 시그널링되는 영상/비디오 정보는 픽처 내 파티셔닝 관련 정보, 인트라/인터 예측 정보, 레지듀얼 정보, 인루프 필터링 정보 등을 포함할 뿐 아니라, 상기 슬라이스 헤더에 포함된 정보, 상기 APS에 포함된 정보, 상기 PPS에 포함된

정보, 상기 SPS에 포함된 정보, 상기 VPS에 포함된 정보 및/또는 상기 DPS에 포함된 정보를 포함할 수 있다.

- [108] **신경망 포스트 필터(Neural-Network Post-Filter)를 이용한 일반적인 후처리 필터링 절차**
- [109] 이 절차에 대한 입력은 비트스트림 BitstreamToFilter이고, 출력은 NNPF 출력 픽처들의 리스트 ListNnpfOutputPics이다. 먼저, BitstreamToFilter가 디코딩되고, 리스트 CroppedDecodedPictures는 BitstreamToFilter를 디코딩한 결과 출력 순서대로 크롭 및 디코딩된 픽처들의 리스트로 설정된다. 두 번째로, 하나의 픽처에 대한 필터링 프로세스는 CroppedDecodedPictures에 존재하고, 하나 이상의 NNPF가 활성화된 각각의 크롭 및 디코딩된 픽처(cropped decoded picture)에 대해 출력 순서대로 반복적으로 적용된다. ListNnpfOutputPics의 픽처들의 순서는 출력 순서대로이다.
- [110] 특정 시간 인스턴스와 관련된 픽처는 ListNnpfOutputPics 내에 하나 이상 있어서는 안 된다. CroppedDecodedPictures 내의 어느 특정 픽처에 대해 다수의 NNPF가 활성화되어 있고 NNPF 중 어느 것이든 선택할 수 있음에도 불구하고 오직 하나만 적용되도록 선택이 허용되는 경우에는, 특정 픽처에 대해 어느 NNPF가 적용되도록 선택되는지와 무관하게 상기 제약이 적용될 수 있다.
- [111] 후술하는 필터링 절차는 CroppedDecodedPictures에 존재하고 하나 이상의 NNPF가 활성화된 각각의 크롭 및 디코딩된 픽처(현재 픽처로 지칭됨)에 적용될 수 있다. 현재 픽처에 NNPF를 적용할 때, 필터링된 및/또는 보간된 픽처들은 NNPF SEI 메시지의 시맨틱스에 명시된 NNPF 절차를 현재 픽처에 패치 단위로 적용함으로써 NNPF에 의해 생성된다.
- [112] 현재 픽처에 NNPF를 적용할 때, NNPF의 출력 텐서(tensor)에 저장되는 NNPF를 적용함으로써 NNPF에 의해 생성되는 픽처들의 순서는 출력 순서이다. 적용된 NNPF가 현재 픽처에 적용된 마지막 NNPF인 경우, NNPF에 의해 생성되고 NNPF 절차에 의해 출력된 픽처들은 NNPF의 출력 텐서에 픽처가 저장되는 순서와 같은 순서로 ListNnpfOutputPics에 포함될 수 있다.
- [113] **신경망 포스트 필터 특성(Neural-network post-filter characteristics, NNPF)**
- [114] 표 1 내지 표 3의 결합은 NNPF 신택스 구조를 나타낸다.

[115] [35.1]

nn_post_filter_characteristics(payloadSize) {	Descriptor
nnpfc_purpose	u(16)
nnpfc_id	ue(v)
nnpfc_base_flag	u(1)
nnpfc_mode_idc	ue(v)
if(nnpfc_mode_idc == 1) {	
while(!byte_aligned())	
nnpfc_reserved_zero_bit_a	u(1)
nnpfc_tag_uri	st(v)
nnpfc_uri	st(v)
}	
nnpfc_property_present_flag	u(1)
if(nnpfc_property_present_flag) {	
/* input and output formatting */	
nnpfc_num_input_pics_minus1	ue(v)
if(nnpfc_num_input_pics_minus1 > 0) {	
for(i = 0; i <= nnpfc_num_input_pics_minus1; i++)	
nnpfc_input_pic_output_flag[i]	u(1)
nnpfc_absent_input_pic_zero_flag	u(1)
}	
if(chromaUpsamplingFlag)	
nnpfc_out_sub_c_flag	u(1)
if(colourizationFlag)	
nnpfc_out_colour_format_idc	u(2)
if(resolutionResamplingFlag) {	
nnpfc_pic_width_num_minus1	ue(v)
nnpfc_pic_width_denom_minus1	ue(v)
nnpfc_pic_height_num_minus1	ue(v)
nnpfc_pic_height_denom_minus1	ue(v)
}	
if(pictureRateUpsamplingFlag)	
for(i = 0; i < nnpfc_num_input_pics_minus1; i++)	
nnpfc_interpolated_pics[i]	ue(v)
nnpfc_component_last_flag	u(1)

[116] [35.2]

nnpfc_inp_format_idc	ue(v)
nnpfc_auxiliary_inp_idc	ue(v)
nnpfc_inp_order_idc	ue(v)
if(nnpfc_inp_format_idc == 1) {	
if(nnpfc_inp_order_idc != 1)	
nnpfc_inp_tensor_luma_bitdepth_minus8	ue(v)
if(nnpfc_inp_order_idc != 0)	
nnpfc_inp_tensor_chroma_bitdepth_minus8	ue(v)
}	
nnpfc_out_format_idc	ue(v)
nnpfc_out_order_idc	ue(v)
if(nnpfc_out_format_idc == 1) {	
if(nnpfc_out_order_idc != 1)	
nnpfc_out_tensor_luma_bitdepth_minus8	ue(v)
if(nnpfc_out_order_idc != 0)	
nnpfc_out_tensor_chroma_bitdepth_minus8	ue(v)
}	
nnpfc_separate_colour_description_present_flag	u(1)
if(nnpfc_separate_colour_description_present_flag) {	
nnpfc_colour_primaries	u(8)
nnpfc_transfer_characteristics	u(8)
if(nnpfc_out_format_idc == 1) {	
nnpfc_matrix_coeffs	u(8)
nnpfc_full_range_flag	u(1)
}	
}	
nnpfc_chroma_loc_info_present_flag	u(1)
if(nnpfc_chroma_loc_info_present_flag)	
nnpfc_chroma_sample_loc_type_frame	ue(v)
nnpfc_overlap	ue(v)
nnpfc_constant_patch_size_flag	u(1)
if(nnpfc_constant_patch_size_flag) {	
nnpfc_patch_width_minus1	ue(v)
nnpfc_patch_height_minus1	ue(v)

[117] [표 3]

} else {	
nnpfc_extended_patch_width_cd_delta_minus1	ue(v)
nnpfc_extended_patch_height_cd_delta_minus1	ue(v)
}	
nnpfc_padding_type	ue(v)
if(nnpfc_padding_type == 4) {	
if(nnpfc_inp_order_idc != 1)	
nnpfc_luma_padding_val	ue(v)
if(nnpfc_inp_order_idc != 0) {	
nnpfc_cb_padding_val	ue(v)
nnpfc_cr_padding_val	ue(v)
}	
}	
nnpfc_complexity_info_present_flag	u(1)
if(nnpfc_complexity_info_present_flag) {	
nnpfc_parameter_type_idc	u(2)
if(nnpfc_parameter_type_idc != 2)	
nnpfc_log2_parameter_bit_length_minus3	u(2)
nnpfc_num_parameters_idc	u(6)
nnpfc_num_kmac_operations_idc	ue(v)
nnpfc_total_kilobyte_size	ue(v)
}	
nnpfc_metadata_extension_num_bits	ue(v)
if(nnpfc_metadata_extension_num_bits > 0)	
nnpfc_reserved_metadata_extension	u(v)
}	
/* ISO/IEC 15938-17 bitstream */	
if(nnpfc_mode_idc == 0) {	
while(!byte_aligned())	
nnpfc_reserved_zero_bit_b	u(1)
for(i = 0; more_data_in_payload(); i++)	
nnpfc_payload_byte[i]	b(8)
}	
}	

[118] 표 1 내지 표 3의 NNPFC 선택스 구조는 SEI(supplemental enhancement information) 메시지 형태로 시그널링될 수 있다. 표 1 내지 표 3의 NNPFC 선택스 구조를 시그널링하는 SEI 메시지는 NNPFC SEI 메시지라고 지칭될 수 있다.

[119] NNPFC SEI 메시지는 후처리(post-processing) 필터로 이용될 수 있는 신경망을

특정할 수 있다. 특정 픽처들에 대해 특정된 후처리 필터들(NNPFs)의 사용은 신경망 포스트 필터 활성화 SEI 메시지들(neural-network post-filter activation SEI messages)을 이용하여 나타낼 수 있다. 여기서, '후처리 필터'와 '포스트 필터'는 동일한 의미를 가질 수 있다.

- [120] 이러한 SEI 메시지들을 사용하려면 아래와 같은 변수들의 정의가 필요할 수 있다.
- [121] - 루마 샘플 단위의 입력 픽처 너비와 높이는 각각 `CroppedWidth` 및 `CroppedHeight`로 나타낼 수 있다.
- [122] - 0 내지 `numInputPics-1`의 범위의 인덱스 `idx`를 가지는 입력 픽처들의 루마 샘플 어레이인 `CroppedYPic[idx]`와, 크로마 샘플 어레이인 `CroppedCbPic[idx]` 및 `CroppedCrPic[idx]`는 이들이 존재하는 경우에 NNPF에 대한 입력으로 이용될 수 있다.
- [123] - `BitDepthY`는 입력 픽처들의 루마 샘플 어레이에 대한 비트 뎀스를 나타낼 수 있다.
- [124] - `BitDepthC`는 입력 픽처들의 크로마 샘플 어레이들(존재하는 경우)의 비트 뎀스를 나타낼 수 있다.
- [125] - `ChromaFormatIdc`는 크로마 포맷 식별자를 나타낼 수 있다.
- [126] - `nnpfc_auxiliary_inp_idc`의 값이 1인 경우, 필터링 강도 제어 값 어레이 `StrengthControlVal[idx]`는 0 내지 `numInputPics-1`의 범위의 인덱스 `idx`를 가지는 입력 픽처들의 0 내지 1 범위의 실수를 포함해야 한다.
- [127] 인덱스 0을 가지는 입력 픽처는 상기 NNPF SEI 메시지에 의해 정의되는 NNPF가 NNPF SEI 메시지에 의해서 활성화되는 픽처에 대응할 수 있다. 0 내지 `numInputPics-1`의 범위의 인덱스 `i`를 가지는 입력 픽처는 출력 순서상 인덱스 `i-1`를 가지는 입력 픽처 보다 선행할 수 있다.
- [128] 변수 `SubWidthC` 및 `SubHeightC`는 `ChromaFormatIdc`로부터 유도될 수 있다. 동일한 픽처에 대해 둘 이상의 NNPF SEI 메시지가 존재할 수 있다. 서로 다른 `nnpfc_id` 값을 가지는 둘 이상의 NNPF SEI 메시지가 동일한 픽처에 대해 존재하거나 활성화된 경우, 둘 이상의 NNPF SEI 메시지는 서로 같거나 다른 `nnpfc_purpose` 및 `nnpfc_mode_idx` 값을 가질 수 있다.
- [129] `nnpfc_purpose`는 표 3에 명시되는 NNPF의 목적을 나타낼 수 있다. 여기서, $(nnpfc_purpose \& bitMask)$ 가 0이 아닌 것은 NNPF가 표 3의 `bitMask` 값과 연관된 목적을 가지고 있음을 나타낼 수 있다. `nnpfc_purpose`가 0보다 크고 $(nnpfc_purpose \& bitMask)$ 가 0인 경우 `bitMask` 값과 연관된 목적은 NNPF에 적용되지 않을 수 있다. `nnpfc_purpose`가 0이면 NNPF가 사용될 수 있다.
- [130] `nnpfc_purpose` 값은 비트스트림에서 0부터 63까지의 범위에 존재해야 한다. `nnpfc_purpose`에 대한 64 내지 65535 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에 존재하지 않을 수 있다. 디코더들은 64 내지 65535 범위의 `nnpfc_purpose`를 포함하는 NNPF SEI 메시지를 무시해야 한다.

[131] [표4]

bitMask	Interpretation
0x01	일반적인 시각적 품질 개선
0x02	크로마 업샘플링 (4:2:0 크로마 포맷으로부터 4:2:2 또는 4:4:4 크로마 포맷으로, 또는 4:2:2 크로마 포맷으로부터 4:4:4 크로마 포맷으로)
0x04	해상도 리샘플링 (너비 또는 높이 증가 또는 감소)
0x08	픽처 레이트 업샘플링
0x10	비트 뎁스 업샘플링 (루마 비트 뎁스 또는 크로마 비트 뎁스 증가)
0x20	색상화

[132] nnpfc_purpose가 크로마 업샘플링, 해상도 리샘플링, 사진 속도 업샘플링, 비트 뎁스 업샘플링 및 색상화를 각각 포함하는 NNPF의 목적을 나타내는지 여부를 특정하는 변수들 chromaUpsamplingFlag, resolutionResamplingFlag, pictureRateUpsamplingFlag, bitDepthUpsamplingFlag, 및 colourizationFlag는 다음의 표 5와 같이 유도될 수 있다.

[133] [표5]

```

chromaUpsamplingFlag = ( ( nnpfc_purpose & 0x02 ) > 0 ) ? 1 : 0
resolutionResamplingFlag = ( ( nnpfc_purpose & 0x04 ) > 0 ) ? 1 : 0
pictureRateUpsamplingFlag = ( ( nnpfc_purpose & 0x08 ) > 0 ) ? 1 : 0
bitDepthUpsamplingFlag = ( ( nnpfc_purpose & 0x10 ) > 0 ) ? 1 : 0
colourizationFlag = ( ( nnpfc_purpose & 0x20 ) > 0 ) ? 1 : 0

```

[134] nnpfc_purpose의 예약된 값이 향후 사용되는 경우 상기 SEI 메시지의 신택스는 해당 값과 일치하는 nnpfc_purpose에 의해 존재 여부가 결정되는 신택스 엘리먼트로 확장될 수 있다.

[135] ChromaFormatIdc가 3이면 chromaUpsamplingFlag는 0과 동일해야 한다.

[136] ChromaFormatIdc 또는 chromaUpsamplingFlag가 0이 아니면 colourizationFlag는 0과 동일해야 한다.

[137] pictureRateUpsamplingFlag가 1이고, 인덱스 0인 입력 픽처가 5와 동일한 fp_arrangement_type을 가지는 프레임 패킹 배열 SEI 메시지(frame packing arrangement SEI message)에 연관되면 모든 입력 픽처는 5와 동일한 fp_arrangement_type 및 동일 값의 fp_current_frame_is_frame0_flag를 가지는 프레임 패킹 배열 SEI 메시지와 연관될 수 있다.

[138] nnpfc_id는 후처리 필터를 식별하는 데 사용할 수 있는 식별 번호를 포함할 수 있다. nnpfc_id 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다. 256 내지 511 범위 및

2^{31} 내지 $2^{32} - 2$ 범위의 `nnpfc_id` 값은 향후 사용을 위해 예약될 수 있다. 디코더들은 256 내지 511 범위 또는 2^{31} 내지 $2^{32} - 2$ 범위의 `nnpfc_id`를 가진 NNPFC SEI 메시지를 무시해야 한다.

- [139] NNPFC SEI 메시지가 현재 CLVS(Coded Layer Video Sequence) 내에서 특정 `nnpfc_id` 값을 가지는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지인 경우에는 다음이 적용될 수 있다.
- [140] - 상기 SEI 메시지는 기본(base) 후처리 필터(NNPF)를 나타낼 수 있다.
- [141] - 상기 SEI 메시지는, 출력 순서에서, 현재 CLVS가 종료될 때까지 현재 디코딩된 픽처 및 현재 레이어의 모든 후속하는 디코딩된 픽처들과 관련될 수 있다.
- [142] `nnpfc_base_flag`의 값 1은 SEI 메시지가 기본 NNPF를 명시하는 것을 나타낼 수 있다. `nnpfc_base_flag`의 값 0은 SEI 메시지가 기본 NNPF에 관련된 업데이트를 명시하는 것을 나타낼 수 있다.
- [143] `nnpfc_base_flag` 값에는 다음의 제약 조건이 적용될 수 있다.
- [144] - NNPFC SEI 메시지가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는 디코딩 순서상 첫 번째 NNPFC SEI 메시지인 경우 `nnpfc_base_flag` 값은 1과 동일해야 한다.
- [145] - NNPFC SEI 메시지 `nnpfcB`가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지가 아니고 `nnpfc_base_flag` 값이 1과 동일한 경우 NNPFC SEI 메시지는 디코딩 순서상, 동일한 `nnpfc_id` 값을 가지는 첫 번째 NNPFC SEI 메시지 `nnpfcA`의 반복이어야 한다. 즉, `nnpfcB`의 페이로드 콘텐츠는 `nnpfcA`의 페이로드 콘텐츠와 동일해야 한다.
- [146] `nnpfc_base_flag`이 0이면 다음이 적용될 수 있다.
- [147] - 상기 SEI 메시지는 동일한 `nnpfc_id` 값을 가지는 디코딩 순서에서 선행하는 기본 NNPF에 관련된 업데이트를 정의할 수 있다. 업데이트들은 누적되지 않고 오히려 각 업데이트가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는, 디코딩 순서상 첫 번째 NNPFC SEI 메시지에 의해 명시된 NNPF인 기본 NNPF에 적용될 수 있다. 상기 SEI 메시지에 의해 정의된 NNPF는 동일한 `nnpfc_id` 값을 가지는 기본 NNPF와 관하여 상기 SEI 메시지에 의해 정의된 업데이트를 적용하여 획득될 수 있다.
- [148] - 상기 SEI 메시지는, 현재 CLVS가 종료될 때까지 또는 현재 CLVS 내에서 출력 순서상 현재 디코딩된 픽처 뒤에 오는 디코딩 픽처를 제외할 때까지 출력 순서상 현재 디코딩된 픽처 및 현재 레이어의 모든 후속하는 디코딩된 픽처들과 관련되고, `nnpfc_base_flag`가 0이고 현재 CLVS 내 더 빠른 특정 `nnpfc_id` 값을 갖는, 디코딩 순서에서 후속하는 NNPFC SEI 메시지와 관련될 수 있다.
- [149] `nnpfc_mode_idc`의 값 0은 상기 SEI 메시지가 기본 NNPF를 명시하거나(`nnpfc_base_flag`이 1인 경우) 동일한 `nnpfc_id` 값을 갖는 기본 NNPF에 연관된 업데이트인(`nnpfc_base_flag`가 0인 경우) ISO/IEC 15938-17를 준수하는

비트스트림을 포함함을 나타낼 수 있다.

- [150] `nnpfc_mode_idc`가 1인 경우 `nnpfc_mode_idc`의 값 1은 `nnpfc_id` 값에 관련된 기본 NNPF가 신경망임을 나타낼 수 있으며, 상기 신경망은 태그 URI `nnpfc_tag_uri`로 식별되는 형식을 사용하여 `nnpfc_uri`로 표시된 URI에 의해 식별되는 신경망일 수 있다. `nnpfc_mode_idc`가 0인 경우, `nnpfc_mode_idc`의 값 1은 동일한 `nnpfc_id` 값을 가진 기본 NNPF에 대한 업데이트가 태그 URI `nnpfc_tag_uri`로 식별되는 형식을 사용하여 `nnpfc_uri`로 표시된 URI로 정의됨을 나타낼 수 있다.
- [151] `nnpfc_mode_idc`의 값은 비트스트림에서 0 내지 1의 범위에 존재해야 한다. `nnpfc_mode_idc`에 대한 2 내지 255 범위의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에 존재하지 않을 수 있다. 디코더들은 2 내지 255 범위의 `nnpfc_mode_idc`를 가지는 NNPF SEI 메시지를 무시해야 한다. 255 보다 큰 `nnpfc_mode_idc`의 값들은 비트스트림에 존재하지 않으며, 향후 사용을 위해 예약되지 않을 수 있다.
- [152] `nnpfc_reserved_zero_bit_a`는 비트스트림에서 0과 동일해야 한다. 디코더들은 `nnpfc_reserved_zero_bit_a`의 값이 0이 아닌 NNPF SEI 메시지들을 무시해야 한다.
- [153] `nnpfc_tag_uri`는 기본 NNPF로 이용되는 신경망 또는 `nnpfc_uri`에 의해 특정되는 동일한 `nnpfc_id` 값을 가지는 기본 NNPF와 관하여 업데이트에 대한 형식 및 관련 정보를 식별하는 IETF RFC 4151에 특정된 신택스 및 시멘틱스를 가지는 태그 URI를 포함할 수 있다.
- [154] `nnpfc_tag_uri`는 중앙 등록 기관 없이도 `nnrpf_uri`에 의해 명시된 신경망 데이터의 형식을 고유하게 식별할 수 있다.
- [155] "tag:iso.org,2023:15938-17"과 동일한 `nnpfc_tag_uri`는 `nnpfc_uri`로 식별된 신경망 데이터가 ISO/IEC 15938-17을 준수함을 나타낼 수 있다.
- [156] `nnpfc_uri`는 기본 NNPF로 사용되는 신경망 또는 동일한 `nnpfc_id` 값을 가지는 기본 NNPF에 대한 업데이트를 식별하는 IETF Internet Standard 66에 명시된 신택스 및 시멘틱스를 가지는 URI를 포함할 수 있다.
- [157] `nnpfc_formatting_and_purpose_flag`의 값 1은 필터 목적, 입력 포매팅, 출력 포매팅 및 복잡성과 관련된 신택스 엘리먼트가 존재함을 나타낼 수 있다. `nnpfc_formatting_and_purpose_flag`의 값 0은 필터 목적, 입력 포매팅, 출력 포매팅 및 복잡성과 관련된 신택스 엘리먼트가 존재하지 않음을 나타낼 수 있다.
- [158] `nnpfc_base_flag`가 1인 경우 `nnpfc_property_present_flag`는 1과 동일해야 한다.
- [159] `nnpfc_property_present_flag`가 0인 경우 `nnpfc_property_present_flag`가 1인 경우에만 존재할 수 있는 모든 신택스 엘리먼트들의 값들은 상기 SEI 메시지가 업데이트를 제공하는 기본 NNPF를 포함하는 NNPF SEI 메시지에서 그들의 해당 신택스 엘리먼트들 각각과 동일하다고 추론될 수 있다.
- [160] NNPF SEI 메시지 `nnpfcCurr`가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는, 디코딩 순서상 첫번째 NNPF SEI 메시지가 아니고, 상기 특정 `nnpfc_id` 값을

- 찾는 첫번째 NNPFC SEI 메시지의 반복이 아니며 (즉, `nnpfc_base_flag` 값이 0과 동일함), `nnpfc_property_present_flag`의 값이 1과 동일한 경우 다음의 제약 조건이 적용될 수 있다.
- [161] - NNPFC SEI 메시지의 `nnpfc_purpose` 값은 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는, 디코딩 순서상 첫번째 NNPFC SEI 메시지의 `nnpfc_purpose` 값과 동일해야 한다.
- [162] - NNPFC SEI 메시지에서 디코딩 순서상 `nnpfc_complexity_info_present_flag` 앞에 오고 `nnpfc_property_present_flag` 뒤에 오는 선택스 엘리먼트들의 값들은 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는, 디코딩 순서상 첫번째 NNPFC SEI 메시지의 해당 선택스 엘리먼트들의 값들과 동일해야 한다.
- [163] 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는, 디코딩 순서상 첫번째 NNPFC SEI 메시지(하기의 `nnpfcBase`로 표시됨)
`nnpfc_complexity_info_present_flag`는 0과 동일하거나
`nnpfc_complexity_info_present_flag` 모두 1과 동일해야 하고 다음이 모두 적용될 수 있다.
- [164] - `nnpfcCurr`의 `nnpfc_parameter_type_idc`는 `nnpfcBase`의 `nnpfc_parameter_type_idc`와 동일해야 한다.
- [165] - `nnpfcCurr`의 `nnpfc_log2_parameter_bit_length_minus3`는 존재하는 경우 `nnpfcBase`의 `nnpfc_log2_parameter_bit_length_minus3`와 작거나 같아야 한다.
- [166] - `nnpfcBase`의 `nnpfc_num_parameters_idc`이 0이면, `nnpfcCurr`의 `nnpfc_num_parameters_idc`는 0과 동일해야 한다.
- [167] - 그렇지 않으면 (`nnpfcBase`의 `nnpfc_num_parameters_idc`가 0보다 큼)
`nnpfcCurr`의 `nnpfc_num_parameters_idc`는 0보다 크고 `nnpfcBase`의 `nnpfc_num_parameters_idc`보다 작거나 같아야 한다.
- [168] - `nnpfcBase`의 `nnpfc_num_kmac_operations_idc`가 0이면, `nnpfcCurr`의 `nnpfc_num_kmac_operations_idc`는 0과 동일해야 한다.
- [169] - 그렇지 않으면 (`nnpfcBase`의 `nnpfc_num_kmac_operations_idc`가 0보다 큼)
`nnpfcCurr`의 `nnpfc_num_kmac_operations_idc`는 0보다 크고 `nnpfcBase`의 `nnpfc_num_kmac_operations_idc`보다 작거나 같아야 한다.
- [170] - `nnpfcBase`의 `nnpfc_total_kilobyte_size`가 0이면, `nnpfcCurr`의 `nnpfc_total_kilobyte_size`는 0과 동일해야 한다.
- [171] - 그렇지 않으면 (`nnpfcBase`의 `nnpfc_total_kilobyte_size`가 0보다 큼) `nnpfcCurr`의 `nnpfc_total_kilobyte_size`는 0보다 크고 `nnpfcBase`의 `nnpfc_total_kilobyte_size`보다 작거나 같아야 한다.
- [172] `nnpfc_num_input_pics_minus1 + 1`은 NNPF의 입력으로 사용되는 픽처들의 수를 나타낼 수 있다. `nnpfc_num_input_pics_minus1`의 값은 0 내지 63의 범위에 존재해야 한다. `pictureRateUpsamplingFlag`이 1과 동일한 경우 `nnpfc_num_input_pics_minus1`의 값은 0보다 커야 한다.

- [173] NNPF에 대한 입력으로 사용되는 픽처들의 수를 나타내는 변수 numInputPics는 다음의 수식 1과 같이 유도될 수 있다.
- [174] [수식 1]
- [175]
$$\text{numInputPics} = \text{nnpfc_num_input_pics_minus1} + 1$$
- [176] nnpfc_input_pic_output_flag[i]의 값 1은 i번째 입력 픽처에 대하여 NNPF가 해당 출력 픽처를 생성함을 나타낼 수 있다. nnpfc_input_pic_output_flag[i]의 값 0은 i번째 입력 픽처에 대하여 NNPF가 해당 출력 픽처를 생성하지 않음을 나타낼 수 있다. nnpfc_num_input_pics_minus1이 0과 동일한 경우 nnpfc_input_pic_output_flag[0]은 1로 추론될 수 있다. pictureRateUpsamplingFlag이 0과 동일하고 nnpfc_num_input_pics_minus1이 0보다 큰 경우 0 내지 nnpfc_num_input_pics_minus1 범위에 있는 i의 적어도 하나의 값에 대하여 nnpfc_input_pic_output_flag[i]는 1과 동일해야 한다.
- [177] nnpfc_absent_input_pic_zero_flag의 값 1은 NNPF가, 비트스트림에 존재하지 않는 입력 픽처가 샘플 값들이 0인 샘플 어레이들로 표현될 것으로 예상함을 나타낼 수 있다. nnpfc_absent_input_pic_zero_flag의 값 0은 NNPF가, 비트스트림에 존재하지 않는 입력 픽처가 비트스트림 내 출력 순서상 가장 가까운 입력 픽처로 표현될 것으로 예상함을 나타낼 수 있다. nnpfc_out_sub_c_flag는 chromaUpsamplingFlag가 1인 경우 변수들 outSubWidthC 및 outSubHeightC의 값들을 나타낼 수 있다. nnpfc_out_sub_c_flag의 값 1은 outSubWidthC의 값이 1이고 outSubHeightC의 값이 1임을 나타낼 수 있다. nnpfc_out_sub_c_flag의 값 0은 outSubWidthC의 값이 2이고 outSubHeightC의 값이 1임을 나타낼 수 있다. ChromaFormatIdc의 값이 2이고 nnpfc_out_sub_c_flag가 존재하는 경우, nnpfc_out_sub_c_flag의 값은 1과 동일해야 한다.
- [178] nnpfc_out_colour_format_idc는, colourizationFlag가 1인 경우 NNPF의 색상 형식 및 결과적으로 변수들 outSubWidthC 및 outSubHeightC의 값들을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 1은 NNPF 출력의 색상 형식이 4:2:0이고, outSubWidthC 및 outSubHeightC 둘 다 2인 것을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 2는 NNPF 출력의 색상 형식이 4:2:2이고, outSubWidthC는 2이고, outSubHeightC는 1인 것을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 3은 NNPF 출력의 색상 형식이 4:4:4이고, outSubWidthC 및 outSubHeightC 둘 다 1인 것을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값은 0이 아니어야 한다.
- [179] chromaUpsamplingFlag 및 colourizationFlag 둘 다 0인 경우 outSubWidthC 및 outSubHeightC는 SubWidthC 및 SubHeightC와 각각 동일하다고 추론될 수 있다. nnpfc_pic_width_num_minus1 + 1 및 nnpfc_pic_width_denom_minus1 + 1은, CroppedWidth에 관한 NNPF 출력 픽처 너비의 리샘플링 비율에 대하여 각각 분자 및 분모를 나타낼 수 있다. $(\text{nnpfc_pic_width_num_minus1} + 1) \div$

- ($\text{nnpfc_pic_width_denom_minus1} + 1$)의 값은 $1 \div 16$ 내지 16의 범위에 존재해야 한다. $\text{nnpfc_pic_width_num_minus1}$ 및 $\text{nnpfc_pic_width_denom_minus1}$ 이 존재하지 않으면 $\text{nnpfc_pic_width_num_minus1}$ 및 $\text{nnpfc_pic_width_denom_minus1}$ 의 값들은 둘 다 0과 동일하다고 추론될 수 있다.
- [180] nnpfc_id 에 의해 식별된 NNPF를 입력 픽처(들)에 적용한 결과로 발생하는 픽처(들)의 루마 샘플 어레이들의 너비를 나타내는 변수 $\text{nnpfcOutputPicWidth}$ 는 다음의 수식 2와 같이 유도될 수 있다.
- [181] [수식 2]
- [182]
$$\text{nnpfcOutputPicWidth} = \text{Ceil}(\text{CroppedWidth} * (\text{nnpfc_pic_width_num_minus1} + 1) \div (\text{nnpfc_pic_width_denom_minus1} + 1))$$
- [183] $\text{nnpfcOutputPicWidth} \% \text{outSubWidthC}$ 의 값이 0과 동일해야 하는 것은 비트스트림 적합성의 요구사항이다.
- [184] $\text{nnpfc_pic_height_num_minus1} + 1$ 및 $\text{nnpfc_pic_height_denom_minus1} + 1$ 은 CroppedHeight 에 관한 NNPF 출력 픽처 높이의 리샘플링 비율에 대하여 각각 분자 및 분모를 나타낼 수 있다. ($\text{nnpfc_pic_height_num_minus1} + 1$) \div ($\text{nnpfc_pic_height_denom_minus1} + 1$)의 값은 $1 \div 16$ 내지 16의 범위에 존재해야 한다. $\text{nnpfc_pic_height_num_minus1}$ 및 $\text{nnpfc_pic_height_denom_minus1}$ 이 존재하지 않으면 $\text{nnpfc_pic_height_num_minus1}$ 및 $\text{nnpfc_pic_height_denom_minus1}$ 의 값들은 둘 다 0과 동일하다고 추론될 수 있다.
- [185] nnpfc_id 에 의해 식별된 NNPF를 입력 픽처(들)에 적용한 결과로 발생하는 픽처(들)의 루마 샘플 어레이들의 높이를 나타내는 변수 $\text{nnpfcOutputPicHeight}$ 는 다음의 수식 3과 같이 유도될 수 있다.
- [186] [수식 3]
- [187]
$$\text{nnpfcOutputPicHeight} = \text{Ceil}(\text{CroppedHeight} * (\text{nnpfc_pic_height_num_minus1} + 1) \div (\text{nnpfc_pic_height_denom_minus1} + 1))$$
- [188] $\text{nnpfcOutputPicHeight} \% \text{outSubHeightC}$ 의 값이 0과 동일해야 하는 것은 비트스트림 적합성의 요구사항이다.
- [189] $\text{nnpfc_pic_width_num_minus1}$, $\text{nnpfc_pic_width_denom_minus1}$, $\text{nnpfc_pic_height_num_minus1}$, 및 $\text{nnpfc_pic_height_denom_minus1}$ 이 존재하면 다음의 적어도 하나가 참(true)이어야 한다.
- [190] - $\text{nnpfcOutputPicWidth}$ 의 값은 CroppedWidth 와 동일하지 않음.
- [191] - $\text{nnpfcOutputPicHeight}$ 의 값은 CroppedHeight 와 동일하지 않음.
- [192] $\text{nnpfc_interpolated_pics}[i]$ 는 NNPF의 입력으로 사용되는 i 번째 픽처와 ($i + 1$)번째 픽처 사이에서 NNPF에 의해 생성된 보간 픽처의 수를 나타낼 수 있다.
- [193] $\text{nnpfc_interpolated_pics}[i]$ 의 값은 0 내지 63의 범위에 존재해야 한다. $\text{nnpfc_interpolated_pics}[i]$ 의 값은 0 내지 $\text{nnpfc_num_input_pics_minus1} - 1$ 의

범위에서 i 의 적어도 하나의 값에 대하여 0보다 커야 한다.

- [194] 대응하는 입력 픽처를 가지고 NNPF의 출력 텐서에 존재하는 픽처들의 수를 나타내는 변수 NumInpPicsInOutputTensor, 대응하는 입력 픽처를 가지고 NNPF의 출력 텐서에 존재하는 idx 번째 픽처의 입력 픽처 인덱스를 나타내는 InpIdx[idx] 및 NNPF의 출력 텐서에 존재하는 픽처들의 전체 수를 나타내는 numOutputPics는 다음의 표 6과 같이 유도될 수 있다.

- [195] [표6]

```

for( i = 0, numOutputPics = 0; i < numInputPics; i++ )
    if( nnpfc_input_pic_output_flag[ i ] ) {
        InpIdx[ numOutputPics ] = i
        numOutputPics++
    }
NumInpPicsInOutputTensor = numOutputPics
if( pictureRateUpsamplingFlag )
    for( i = 0; i <= numInputPics - 2; i++ )
        numOutputPics += nnpfc_interpolated_pics[ i ]

```

- [196] nnpfc_component_last_flag의 값 1은 NNPF에 대한 입력 텐서 inputTensor의 마지막 차원과 NNPF의 결과인 출력 텐서 outputTensor가 현재 채널에 사용됨을 나타낼 수 있다. nnpfc_component_last_flag의 값 0은 NNPF에 대한 입력 텐서 inputTensor의 세번째 차원과 NNPF의 결과인 출력 텐서 outputTensor가 현재 채널에 사용됨을 나타낼 수 있다.
- [197] 입력 텐서와 출력 텐서의 첫 번째 차원은 일부 신경망 프레임워크에서 사용되는 배치 인덱스(batch index)에 사용될 수 있다. 이 SEI 메시지의 시멘틱스 내 공식은 0과 같은 배치 인덱스에 해당하는 배치 크기를 사용하지만, 신경망 추론에 대한 입력으로 사용되는 배치 크기를 결정하는 것은 후처리의 구현에 의해 결정될 수 있다.
- [198] 예를 들어, nnpfc_inp_order_idc의 값이 3과 같고 nnpfc_auxiliary_inp_idc의 값이 1과 같을 때, 입력 텐서에는 4개의 루마 행렬, 2개의 크로마 행렬 및 1개의 보조 입력 행렬을 포함하여 7개의 채널들이 존재할 수 있다. 이 경우 DeriveInputTensors() 프로세스는 입력 텐서의 7개 채널 각각을 하나씩 유도할 수 있으며, 이러한 채널 중 특정 채널이 처리될 때 해당 채널은 프로세스 중 현재 채널이라고 지칭될 수 있다.
- [199] nnpfc_inp_format_idc는 입력 픽처의 샘플 값을 NNPF의 입력 값으로 변환하는 방법을 나타낼 수 있다. nnpfc_inp_format_idc가 0인 경우 NNPF에 대한 입력 값은 실수이고 함수들 InpY() 및 InpC()은 다음의 수식 4와 같이 지정될 수 있다.
- [200] [수식 4]

[201]
$$\text{InpY}(x) = x \div ((1 \ll \text{BitDepth}_Y) - 1)$$

$$\text{InpC}(x) = x \div ((1 \ll \text{BitDepth}_C) - 1)$$

[202] nnpfc_inp_format_idc의 값이 1이면 NNPF의 입력 값들은 부호 없는 정수들(unsigned integer numbers)이며, 함수들 InpY() 및 InpC()은 표 7과 같이 유도될 수 있다.

[203] [표7]

```

shiftY = BitDepthY - inpTensorBitDepthY
if( inpTensorBitDepthY >= BitDepthY )
    InpY(x) = x << ( inpTensorBitDepthY - BitDepthY )
else
    InpY(x) = Clip3(0, (1 << inpTensorBitDepthY) - 1, (x + (1 << (shiftY - 1))) >> shiftY)

shiftC = BitDepthC - inpTensorBitDepthC
if( inpTensorBitDepthC >= BitDepthC )
    InpC(x) = x << ( inpTensorBitDepthC - BitDepthC )
else
    InpC(x) = Clip3(0, (1 << inpTensorBitDepthC) - 1, (x + (1 << (shiftC - 1))) >> shiftC)

```

[204] 변수 inpTensorBitDepth_Y는 아래에서 설명되는 선택스 요소 nnpfc_inp_tensor_bitlength_minus8로부터 유도될 수 있다. inpTensorBitDepth_C는 아래에서 설명되는 선택스 요소 nnpfc_inp_tensor_chroma_bitdepth_minus8로부터 유도될 수 있다.

[205] 1보다 큰 nnpfc_inp_format_idc의 값들은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않을 수 있다. 디코더들은 nnpfc_inp_format_idc의 예약된 값들을 포함하는 NNPF SEI 메시지를 무시해야 한다.

[206] 0보다 큰 nnpfc_auxiliary_inp_idc는 NNPF의 입력 텐서에 보조 입력 데이터가 존재함을 나타낼 수 있다. nnpfc_auxiliary_inp_idc의 값 0은 보조 입력 데이터가 입력 텐서에 존재하지 않음을 나타낼 수 있다. nnpfc_auxiliary_inp_idc의 값 1은 하기의 수식 5에 설명된 대로 보조 입력 데이터가 유도됨을 나타낼 수 있다.

[207] nnpfc_auxiliary_inp_idc의 값은 비트스트림에서 0 내지 1의 범위에 존재해야 한다. nnpfc_auxiliary_inp_idc에 대한 2 내지 255의 값들은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않을 수 있다. 디코더들은 2 내지 255의 범위의 nnpfc_auxiliary_inp_idc가 포함된 NNPF SEI 메시지를 무시해야 한다. 255보다 큰 nnpfc_auxiliary_inp_idc의 값들은 비트스트림에 존재하지 않으며, 향후 사용을 위해 예약되지 않는다.

[208] nnpfc_inp_order_idc는 NNPF에 대한 입력 텐서를 형성하기 위해 입력 픽처의 샘플 어레이를 정렬하는 방법을 나타낼 수 있다.

[209] nnpfc_inp_order_idc의 값은 비트스트림에서 0 내지 3의 범위에 존재해야 한다.

nnpfc_inp_order_idc에 대한 4 내지 255의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에 존재하지 않는다. 디코더들은 4 내지 255의 범위에 존재하는 nnpfc_inp_order_idc를 가지는 NNFC SEI 메시지를 무시해야 한다. 255보다 큰 nnpfc_inp_order_idc의 값들은 비트스트림에 존재하지 않으며 향후 사용을 위해 예약되지 않는다.

- [210] ChromaFormatIdc의 값이 1이 아닌 경우, nnpfc_inp_order_idc의 값은 3이 아니어야 한다.
- [211] ChromaFormatIdc이 0인 경우 nnpfc_inp_order_idc은 0과 동일해야 한다.
- [212] chromaUpsamplingFlag이 1인 경우 nnpfc_inp_order_idc는 0이 아니어야 한다.
- [213] 표 8은 nnpfc_inp_order_idc 값에 대한 설명을 포함한다.
- [214] [표8]

nnpfc_inp_order_idc	Description
0	nnpfc_auxiliary_inp_idc의 값이 0이면, 각 입력 픽처에 대한 입력 텐서에 루마 행렬이 1개씩 존재할 수 있으며, 채널 수는 1일 수 있다. 그렇지 않고, nnpfc_auxiliary_inp_idc의 값이 1이면 루마 행렬과 보조 입력 행렬이 하나씩 존재할 수 있으며, 채널 수는 2일 수 있다.
1	nnpfc_auxiliary_inp_idc의 값이 0이면, 입력 텐서에 크로마 행렬이 2개 존재할 수 있으며 채널 개수는 2일 수 있다. 그렇지 않고, nnpfc_auxiliary_inp_idc의 값이 1이면 크로마 행렬 2개와 보조 입력 행렬이 1개 존재할 수 있으며 채널 개수는 3일 수 있다.
2	nnpfc_auxiliary_inp_idc의 값이 0이면, 입력 텐서에는 루마 행렬 1개와 크로마 행렬 2개가 존재할 수 있으며, 채널 개수는 3일 수 있다. 그렇지 않고, nnpfc_auxiliary_inp_idc의 값이 1이면 루마 행렬 1개, 크로마 행렬 2개, 보조 입력 행렬 1개가 존재할 수 있으며, 채널 개수는 4일 수 있다.
3	nnpfc_auxiliary_inp_idc의 값이 0이면, 입력 텐서에는 루마 행렬 4개와 크로마 행렬 2개가 존재할 수 있으며, 채널 수는 6일 수 있다. 그렇지 않고, nnpfc_auxiliary_inp_idc의 값이 1이면 루마 행렬 4개, 크로마 행렬 2개, 보조 입력 행렬 1개 존재할 수 있으며, 채널 수는 7일 수 있다. 루마 채널은 도 4와 같이 인터리브(interleaved) 방식으로 유도될 수 있다. nnpfc_inp_order_idc는 크로마 포맷이 4:2:0인 경우에만 사용될 수 있다.
4..255	예약됨

- [215] nnpfc_inp_tensor_bitlength_minus8 + 8은 입력 정수 텐서에서 루마 샘플 값들의 비트 템스를 나타낼 수 있다. inpTensorBitDepth_Y의 값은 수식 5와 같이 유도될 수 있다.

[216] [수식 5]

[217]

$$\text{inpTensorBitDepth}_Y = \text{nnpfc_inp_tensor_luma_bitdepth_minus8} + 8$$

[218] `nnpfc_inp_tensor_luma_bitdepth_minus8`의 값이 0 내지 24의 범위에 존재해야 하는 것은 비트스트림 적합성의 요구사항이다.

[219] `nnpfc_inp_tensor_chroma_bitdepth_minus8 + 8`은 입력 정수 텐서에서 크로마 샘플 값들의 비트 템스를 나타낼 수 있다. `inpTensorBitDepthC`의 값은 수식 6과 같이 유도될 수 있다.

[220] [수식 6]

[221]

$$\text{inpTensorBitDepth}_C = \text{nnpfc_inp_tensor_chroma_bitdepth_minus8} + 8$$

[222] `nnpfc_inp_tensor_chroma_bitdepth_minus8`의 값이 0 내지 24의 범위에 존재해야 하는 것은 비트스트림 적합성의 요구사항이다.

[223] `nnpfc_auxiliary_inp_idc`이 1이면 변수 `strengthControlScaledVal`는 다음의 표 9와 같이 유도될 수 있다.

[224] [표9]

```

for( i = 0; i < numInputPics; i++ )
  if( nnpfc_inp_format_idc == 1 )
    if( nnpfc_inp_order_idc == 0 || nnpfc_inp_order_idc == 2 ||
        nnpfc_inp_order_idc == 3 )
      strengthControlScaledVal[ i ] =
        Floor ( StrengthControlVal[ i ] * ( ( 1 << inpTensorBitDepthY ) - 1 ) )
    else if( nnpfc_inp_order_idc == 1 )
      strengthControlScaledVal[ i ] =
        Floor ( StrengthControlVal[ i ] * ( ( 1 << inpTensorBitDepthC ) - 1 ) )
    else
      strengthControlScaledVal[ i ] = StrengthControlVal[ i ]

```

[225] 패치는 픽처의 성분(예를 들어, 루마 또는 크로마 성분)으로부터의 샘플들의 직사각형 어레이일 수 있다.

[226] 주어진 수직 샘플 좌표 `cTop` 및 입력 텐서에 포함된 샘플들의 패치에 대한 좌상단 샘플 위치를 명시하는 수평 샘플 좌표 `cLeft`에 대한 입력 텐서 `inputTensor`를 유도하기 위한 프로세스 `DeriveInputTensors()`는 표 10 내지 표 12의 결합과 같이 나타낼 수 있다.

[227] [35.10]

```

for( i = 0; i < numInputPics; i++ ) {
  if( nnpfc_inp_order_idc == 0 )
    for( yP = -nnpfc_overlap; yP < inpPatchHeight + nnpfc_overlap; yP++ )
      for( xP = -nnpfc_overlap; xP < inpPatchWidth + nnpfc_overlap; xP++ ) {
        inpVal = InpY( InpSampleVal( cTop + yP, cLeft + xP, CroppedHeight,
          CroppedWidth, CroppedYPic[ i ], 0 ) )
        yPovlp = yP + nnpfc_overlap
        xPovlp = xP + nnpfc_overlap
        if( !nnpfc_component_last_flag )
          inputTensor[ 0 ][ i ][ 0 ][ yPovlp ][ xPovlp ] = inpVal
        else
          inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 0 ] = inpVal
        if( nnpfc_auxiliary_inp_idc == 1 )
          if( !nnpfc_component_last_flag )
            inputTensor[ 0 ][ i ][ 1 ][ yPovlp ][ xPovlp ] = strengthControlScaledVal[ i ]
          else
            inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 1 ] = strengthControlScaledVal[ i ]
      }
    else if( nnpfc_inp_order_idc == 1 )
      for( yP = -nnpfc_overlap; yP < inpPatchHeight + nnpfc_overlap; yP++ )
        for( xP = -nnpfc_overlap; xP < inpPatchWidth + nnpfc_overlap; xP++ ) {
          inpCbVal = InpC( InpSampleVal( cTop + yP, cLeft + xP, CroppedHeight / SubHeightC,
            CroppedWidth / SubWidthC, CroppedCbPic[ i ], 1 ) )
          inpCrVal = InpC( InpSampleVal( cTop + yP, cLeft + xP, CroppedHeight / SubHeightC,
            CroppedWidth / SubWidthC, CroppedCrPic[ i ], 2 ) )
          yPovlp = yP + nnpfc_overlap
          xPovlp = xP + nnpfc_overlap
          if( !nnpfc_component_last_flag ) {
            inputTensor[ 0 ][ i ][ 0 ][ yPovlp ][ xPovlp ] = inpCbVal
            inputTensor[ 0 ][ i ][ 1 ][ yPovlp ][ xPovlp ] = inpCrVal
          } else {
            inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 0 ] = inpCbVal
            inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 1 ] = inpCrVal
          }
          if( nnpfc_auxiliary_inp_idc == 1 )
            if( !nnpfc_component_last_flag )
              inputTensor[ 0 ][ i ][ 2 ][ yPovlp ][ xPovlp ] = strengthControlScaledVal[ i ]
            else
              inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 2 ] = strengthControlScaledVal[ i ]
        }
      }
}

```

[228] [3511]

```

else if( nnpfc_inp_order_idc == 2 )
  for( yP = -nnpfc_overlap; yP < inpPatchHeight + nnpfc_overlap; yP++)
    for( xP = -nnpfc_overlap; xP < inpPatchWidth + nnpfc_overlap; xP++) {
      yY = cTop + yP
      xY = cLeft + xP
      yC = yY / SubHeightC
      xC = xY / SubWidthC
      inpYVal = InpY( InpSampleVal( yY, xY, CroppedHeight,
        CroppedWidth, CroppedYPic[ i ], 0 ) )
      inpCbVal = InpC( InpSampleVal( yC, xC, CroppedHeight / SubHeightC,
        CroppedWidth / SubWidthC, CroppedCbPic[ i ], 1 ) )
      inpCrVal = InpC( InpSampleVal( yC, xC, CroppedHeight / SubHeightC,
        CroppedWidth / SubWidthC, CroppedCrPic[ i ], 2 ) )
      yPovlp = yP + nnpfc_overlap
      xPovlp = xP + nnpfc_overlap
      if( !nnpfc_component_last_flag ) {
        inputTensor[ 0 ][ i ][ 0 ][ yPovlp ][ xPovlp ] = inpYVal
        inputTensor[ 0 ][ i ][ 1 ][ yPovlp ][ xPovlp ] = inpCbVal
        inputTensor[ 0 ][ i ][ 2 ][ yPovlp ][ xPovlp ] = inpCrVal
      } else {
        inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 0 ] = inpYVal
        inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 1 ] = inpCbVal
        inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 2 ] = inpCrVal
      }
      if( nnpfc_auxiliary_inp_idc == 1 )
        if( !nnpfc_component_last_flag )
          inputTensor[ 0 ][ i ][ 3 ][ yPovlp ][ xPovlp ] = strengthControlScaledVal[ i ]
        else
          inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 3 ] = strengthControlScaledVal[ i ]
    }
}

```

[229] [표 12]

```

else if( nnpfc_inp_order_idc == 3 )
    for( yP = -nnpfc_overlap; yP < inpPatchHeight + nnpfc_overlap; yP++ )
        for( xP = -nnpfc_overlap; xP < inpPatchWidth + nnpfc_overlap; xP++ ) {
            yTL = cTop + yP * 2
            xTL = cLeft + xP * 2
            yBR = yTL + 1
            xBR = xTL + 1
            yC = cTop / 2 + yP
            xC = cLeft / 2 + xP
            inpTLVal = InpY( InpSampleVal( yTL, xTL, CroppedHeight,
                CroppedWidth, CroppedYPic[ i ], 0 ) )
            inpTRVal = InpY( InpSampleVal( yTL, xBR, CroppedHeight,
                CroppedWidth, CroppedYPic[ i ], 0 ) )
            inpBLVal = InpY( InpSampleVal( yBR, xTL, CroppedHeight,
                CroppedWidth, CroppedYPic[ i ], 0 ) )
            inpBRVal = InpY( InpSampleVal( yBR, xBR, CroppedHeight,
                CroppedWidth, CroppedYPic[ i ], 0 ) )
            inpCbVal = InpC( InpSampleVal( yC, xC, CroppedHeight / 2,
                CroppedWidth / 2, CroppedCbPic[ i ], 1 ) )
            inpCrVal = InpC( InpSampleVal( yC, xC, CroppedHeight / 2,
                CroppedWidth / 2, CroppedCrPic[ i ], 2 ) )
            yPovlp = yP + nnpfc_overlap
            xPovlp = xP + nnpfc_overlap
            if( !nnpfc_component_last_flag ) {
                inputTensor[ 0 ][ i ][ 0 ][ yPovlp ][ xPovlp ] = inpTLVal
                inputTensor[ 0 ][ i ][ 1 ][ yPovlp ][ xPovlp ] = inpTRVal
                inputTensor[ 0 ][ i ][ 2 ][ yPovlp ][ xPovlp ] = inpBLVal
                inputTensor[ 0 ][ i ][ 3 ][ yPovlp ][ xPovlp ] = inpBRVal
                inputTensor[ 0 ][ i ][ 4 ][ yPovlp ][ xPovlp ] = inpCbVal
                inputTensor[ 0 ][ i ][ 5 ][ yPovlp ][ xPovlp ] = inpCrVal
            } else {
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 0 ] = inpTLVal
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 1 ] = inpTRVal
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 2 ] = inpBLVal
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 3 ] = inpBRVal
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 4 ] = inpCbVal
                inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 5 ] = inpCrVal
            }
            if( nnpfc_auxiliary_inp_idc == 1 )
                if( !nnpfc_component_last_flag )
                    inputTensor[ 0 ][ i ][ 6 ][ yPovlp ][ xPovlp ] = strengthControlScaledVal[ i ]
                else
                    inputTensor[ 0 ][ i ][ yPovlp ][ xPovlp ][ 6 ] = strengthControlScaledVal[ i ]
        }
    }
}

```

[230] nnpfc_out_format_id의 값 0은 후속하는 후처리 또는 디스플레이를 위한, 원하는 비트 렵스 bitDepth에 대해, NNPF에 출력된 샘플 값이 0 내지 $(1 \ll \text{bitDepth}) - 1$ 의 부호 없는 정수 값 범위에 선형적으로 매핑되는 실수임을 나타낼 수 있다. 여기서, 상기 실수는 0 내지 1 값 범위일 수 있다.

[231] nnpfc_out_format_flag의 값 1은 NNPF에 의해 출력된 루마 샘플 값들이 0 내지 (

$1 \ll \text{outTensorBitDepth}_Y - 1$ 범위의 부호 없는 정수이고, NNPF에 의해 출력된 크로마 샘플 값들이 $(1 \ll \text{outTensorBitDepth}_C) - 1$ 범위의 부호 없는 정수임을 나타낼 수 있다.

[232] 1보다 큰 `nnpfc_out_format_idc`의 값들은 향후 사양을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 `nnpfc_out_format_idc`의 예약된 값들을 포함하는 NNPF SEI 메시지를 무시해야 한다.

[233] `nnpfc_out_order_idc`는 NNPF로 인한 샘플들의 출력 순서를 나타낼 수 있다.

[234] `nnpfc_out_order_idc`의 값은 비트스트림에서 0 내지 3의 범위에 존재해야 한다. `nnpfc_out_order_idc`에 대한 4 내지 255의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 4 내지 255의 범위에 존재하는 `nnpfc_out_order_idc`를 포함하는 NNPF SEI 메시지를 무시해야 한다. 255보다 큰 `nnpfc_out_order_idc`의 값들은 비트스트림에 존재하지 않으며, 향후 사용을 위해 예약되지 않는다.

[235] `chromaUpsamplingFlag`가 1이면 `nnpfc_out_order_idc`는 0 또는 3이 아니어야 한다.

[236] `colourizationFlag`가 1이면 `nnpfc_out_order_idc`는 0이 아니어야 한다.

[237] 표 13은 `nnpfc_out_order_idc` 값들에 대한 설명을 포함한다.

[238] [표13]

<code>nnpfc_out_order_idc</code>	Description
0	루마 행렬만이 출력 텐서에 존재하므로, 채널의 수는 1일 수 있다.
1	크로마 행렬들만이 출력 텐서에 존재하므로, 채널의 수는 2일 수 있다.
2	루마 및 크로마 행렬들이 출력 텐서에 존재하므로, 채널의 수는 3일 수 있다.
3	4개의 루마 행렬들과 2개의 크로마 행렬들이 출력 텐서에 존재하므로, 채널의 수는 6일 수 있다. 크로마 포맷이 4:2:0인 경우에만 <code>nnpfc_out_order_idc</code> 가 이용될 수 있다.
4..255	예약됨

[239] `nnpfc_out_tensor_luma_bitdepth_minus8 + 8`은 출력 정수 텐서의 루마 샘플 값들의 비트 뎁스를 나타낼 수 있다. `nnpfc_out_tensor_luma_bitdepth_minus8`의 값은 0 내지 24의 범위에 존재해야 한다. `outTensorBitDepthY`의 값은 수식 7과 같이 유도될 수 있다.

[240] [수식 7]

[241]

$$\text{outTensorBitDepth}_Y = \text{nnpfc_out_tensor_luma_bitdepth_minus8} + 8$$

[242] `nnpfc_out_tensor_chroma_bitdepth_minus8 + 8`은 출력 정수 텐서의 크로마 샘플 값들의 비트 뎁스를 나타낼 수 있다. `nnpfc_out_tensor_chroma_bitdepth_minus8`의

값은 0 내지 24의 범위에 존재해야 한다. $\text{outTensorBitDepth}_C$ 의 값은 수식 8과 같이 유도될 수 있다.

[243] [수식 8]

[244]

$$\text{outTensorBitDepth}_C = \text{nnpfc_out_tensor_chroma_bitdepth_minus8} + 8$$

[245] $\text{bitDepthUpsamplingFlag}$ 가 1이면 $\text{nnpfc_out_format_idc}$ 의 값은 1과 동일해야 하고, 다음의 조건들 중 적어도 하나가 참이어야 한다.

[246] - $\text{nnpfc_out_tensor_luma_bitdepth_minus8}$ 이 존재하고 $\text{outTensorBitDepth}_Y$ 가 BitDepth_Y 보다 큼.

[247] - $\text{nnpfc_out_tensor_chroma_bitdepth_minus8}$ 이 존재하고 $\text{outTensorBitDepth}_C$ 가 BitDepth_C 보다 큼.

[248] $\text{nnpfc_inp_tensor_luma_bitdepth_minus8}$,
 $\text{nnpfc_inp_tensor_chroma_bitdepth_minus8}$,
 $\text{nnpfc_out_tensor_luma_bitdepth_minus8}$, 및
 $\text{nnpfc_out_tensor_chroma_bitdepth_minus8}$ 이 존재하고 $\text{outTensorBitDepth}_Y$ 가 $\text{inpTensorBitDepth}_Y$ 보다 크면 $\text{outTensorBitDepth}_C$ 는 $\text{inpTensorBitDepth}_C$ 보다 작아야 한다.

[249] $\text{nnpfc_inp_tensor_luma_bitdepth_minus8}$,
 $\text{nnpfc_inp_tensor_chroma_bitdepth_minus8}$,
 $\text{nnpfc_out_tensor_luma_bitdepth_minus8}$, 및
 $\text{nnpfc_out_tensor_chroma_bitdepth_minus8}$ 이 존재하고 $\text{outTensorBitDepth}_C$ 가 $\text{inpTensorBitDepth}_C$ 보다 크면 $\text{outTensorBitDepth}_Y$ 는 $\text{inpTensorBitDepth}_Y$ 보다 작아야 한다.

[250] 주어진 수직 샘플 좌표 $cTop$ 및 입력 텐서에 포함된 샘플들의 패치에 대한 좌상단 샘플 위치를 나타내는 수평 샘플 좌표 $cLeft$ 에 대한 출력 텐서 outputTensor 로부터 필터링된 출력 샘플 어레이 FilteredYPic , FilteredCbPic 및 FilteredCrPic 내 샘플 값들을 유도하기 위한 프로세스 $\text{StoreOutputTensors}()$ 는 표 14 및 표 15의 결합과 같이 표현될 수 있다.

[251] [圖14]

```

for( i = 0; i < numOutputPics; i++ ) {
  if( nnpfc_out_order_idc == 0 )
    for( yP = 0; yP < outPatchHeight; yP++ )
      for( xP = 0; xP < outPatchWidth; xP++ ) {
        yY = cTop * outPatchHeight / inpPatchHeight + yP
        xY = cLeft * outPatchWidth / inpPatchWidth + xP
        if( yY < nnpfcOutputPicHeight && xY < nnpfcOutputPicWidth )
          if( !nnpfc_component_last_flag )
            FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ 0 ][ yP ][ xP ]
          else
            FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 0 ]
        }
      }
  else if( nnpfc_out_order_idc == 1 )
    for( yP = 0; yP < outPatchCHeight; yP++ )
      for( xP = 0; xP < outPatchCWidth; xP++ ) {
        xSrc = cLeft * horCScaling + xP
        ySrc = cTop * verCScaling + yP
        if( ySrc < nnpfcOutputPicHeight / outSubHeightC &&
            xSrc < nnpfcOutputPicWidth / outSubWidthC )
          if( !nnpfc_component_last_flag ) {
            FilteredCbPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ 0 ][ yP ][ xP ]
            FilteredCrPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ 1 ][ yP ][ xP ]
          } else {
            FilteredCbPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 0 ]
            FilteredCrPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 1 ]
          }
        }
      }
  else if( nnpfc_out_order_idc == 2 )
    for( yP = 0; yP < outPatchHeight; yP++ )
      for( xP = 0; xP < outPatchWidth; xP++ ) {
        yY = cTop * outPatchHeight / inpPatchHeight + yP
        xY = cLeft * outPatchWidth / inpPatchWidth + xP
        yC = yY / outSubHeightC
        xC = xY / outSubWidthC
        yPc = ( yP / outSubHeightC ) * outSubHeightC
        xPc = ( xP / outSubWidthC ) * outSubWidthC
      }
}

```

[252] [표 15]

```

if (yY < nnpfcOutputPicHeight && xY < nnpfcOutputPicWidth)
  if( !nnpfc_component_last_flag ) {
    FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ 0 ][ yP ][ xP ]
    FilteredCbPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ 1 ][ yPc ][ xPc ]
    FilteredCrPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ 2 ][ yPc ][ xPc ]
  } else {
    FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 0 ]
    FilteredCbPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ yPc ][ xPc ][ 1 ]
    FilteredCrPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ yPc ][ xPc ][ 2 ]
  }
}
else if( nnpfc_out_order_idc == 3 )
  for( yP = 0; yP < outPatchHeight; yP++ )
    for( xP = 0; xP < outPatchWidth; xP++ ) {
      ySrc = cTop / 2 * outPatchHeight / inpPatchHeight + yP
      xSrc = cLeft / 2 * outPatchWidth / inpPatchWidth + xP
      if ( ySrc < nnpfcOutputPicHeight / 2 &&
          xSrc < nnpfcOutputPicWidth / 2 )
        if( !nnpfc_component_last_flag ) {
          FilteredYPic[ i ][ xSrc * 2 ][ ySrc * 2 ] = outputTensor[ 0 ][ i ][ 0 ][ yP ][ xP ]
          FilteredYPic[ i ][ xSrc * 2 + 1 ][ ySrc * 2 ] = outputTensor[ 0 ][ i ][ 1 ][ yP ][ xP ]
          FilteredYPic[ i ][ xSrc * 2 ][ ySrc * 2 + 1 ] = outputTensor[ 0 ][ i ][ 2 ][ yP ][ xP ]
          FilteredYPic[ i ][ xSrc * 2 + 1 ][ ySrc * 2 + 1 ] = outputTensor[ 0 ][ i ][ 3 ][ yP ][ xP ]
          FilteredCbPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ 4 ][ yP ][ xP ]
          FilteredCrPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ 5 ][ yP ][ xP ]
        } else {
          FilteredYPic[ i ][ xSrc * 2 ][ ySrc * 2 ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 0 ]
          FilteredYPic[ i ][ xSrc * 2 + 1 ][ ySrc * 2 ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 1 ]
          FilteredYPic[ i ][ xSrc * 2 ][ ySrc * 2 + 1 ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 2 ]
          FilteredYPic[ i ][ xSrc * 2 + 1 ][ ySrc * 2 + 1 ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 3 ]
          FilteredCbPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 4 ]
          FilteredCrPic[ i ][ xSrc ][ ySrc ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 5 ]
        }
      }
    }
}

```

[253] nnpfc_separate_colour_description_present_flag의 값 1은 NNPF로 인한 픽처에 대한 행렬 계수들에 관련하여 적용된 기본 색상(colour primaries), 변환 특성들, 행렬 계수들, 및 스케일링 및 오프셋 값들의 고유한 조합이 SEI 메시지 선택스 구조에 지정됨을 나타낼 수 있다.

nnpfc_separate_colour_description_present_flag의 값 0은 후처리 필터로 인한 픽처에 대한 행렬 계수들에 관련하여 적용된 기본 색상, 변환 특성들, 행렬 계수들, 및 스케일링 및 오프셋 값들의 조합이 CLVS의 VUI 파라미터들에 표시된 것과 동일함을 나타낼 수 있다.

[254] nnpfc_colour_primaries는 다음을 제외하고 vui_colour_primaries 선택스 요소에 대해 정의된 것과 동일한 시멘틱스를 가질 수 있다.

[255] - nnpfc_colour_primaries는 CLVS에 사용되는 기본 색상이 아닌, SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처의 기본 색상을 나타낼 수 있다.

[256] - NNPF SEI 메시지에 nnpfc_colour_primaries가 존재하지 않으면,

- nnpfc_colour_primaries의 값은 vui_colour_primaries의 값과 동일한 것으로 추론될 수 있다.
- [257] nnpfc_transfer_characteristics는 다음을 제외하고 vui_transfer_characteristics 선택스 요소에 대해 정의된 것과 동일한 시멘틱스를 가질 수 있다.
- [258] - nnpfc_transfer_characteristics는 CLVS에 사용되는 변환 특성이 아닌, SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처의 변환 특성을 나타낼 수 있다.
- [259] - NNPF SEI 메시지에 nnpfc_transfer_characteristics가 존재하지 않으면, nnpfc_transfer_characteristics의 값은 vui_transfer_characteristics의 값과 동일한 것으로 추론될 수 있다.
- [260] nnpfc_matrix_coefs는 녹색, 파란색, 빨간색 또는 Y, Z, X 원색에서 루마 및 크로마 신호들을 유도하는데 사용되는 수식을 설명할 수 있다. 그것의 시멘틱스들은 상기 SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처에 적용되며, outTensorBitDepth_Y 및 outTensorBitDepth_C 각각과 동일한 outTensorBitDepth_C 및 BitDepth_C를 포함하는 MatrixCoefficients에 대하여 지정된 것과 동일할 수 있다.
- [261] NNPF SEI 메시지에 nnpfc_matrix_coefs가 존재하지 않으면, nnpfc_matrix_coefs의 값은 vui_matrix_coefs의 값과 동일한 것으로 추론될 수 있다.
- [262] 다음의 조건들이 모두 참이 아닌 한 nnpfc_matrix_coefs는 0과 동일할 수 없다.
- [263] - nnpfc_out_tensor_chroma_bitdepth_minus8는 nnpfc_out_tensor_luma_bitdepth_minus8과 동일함.
- [264] - nnpfc_out_order_idc가 2과 동일하고, outSubHeightC가 1과 동일하며, outSubWidthC는 1과 동일함.
- [265] 다음의 조건들 중 하나가 참이 아닌 한 nnpfc_matrix_coefs는 8과 동일할 수 없다.
- [266] - nnpfc_out_tensor_chroma_bitdepth_minus8는 nnpfc_out_tensor_luma_bitdepth_minus8과 동일함.
- [267] - nnpfc_out_tensor_chroma_bitdepth_minus8이 nnpfc_out_tensor_luma_bitdepth_minus8 + 1과 동일하고, nnpfc_out_order_idc가 2와 동일하고, outSubHeightC가 1과 동일하며, outSubWidthC가 1과 동일함.
- [268] nnpfc_full_range_flag는 nnpfc_matrix_coefs에 의해 지정된 대로 행렬 계수와 관련하여 적용된 스케일링 및 오프셋 값들을 나타낼 수 있다. 그것의 시멘틱스들은 VideoFullRangeFlag 파라미터에 대하여 지정된 것과 동일할 수 있다. 그것이 존재하지 않으면 nnpfc_full_range_flag의 값은 0과 동일한 것으로 추론될 수 있다.
- [269] nnpfc_chroma_loc_info_present_flag의 값 1은 NNPF SEI 메시지의 nnpfc_chroma_sample_loc_type_frame 선택스 요소의 존재를 나타낼 수 있다.

`nnpfc_chroma_loc_info_present_flag`의 값 0은 NNPF SEI 메시지의 `nnpfc_chroma_sample_loc_type_frame` 선택스 요소의 부재를 나타낼 수 있다. `colourizationFlag`가 0이거나 `nnpfc_out_colour_format_idc`가 1이 아닌 경우 `nnpfc_chroma_loc_info_present_flag`의 값은 0과 동일해야 한다.

[270] `nnpfc_chroma_sample_loc_type_frame`은 `nnpfc_chroma_sample_loc_type_frame`이 6이 아니고, `nnpfc_out_colour_format_idc`가 1과 동일하면 출력 픽처들의 크로마 샘플들의 위치를 나타낼 수 있다. `nnpfc_chroma_sample_loc_type_frame`이 6이고 `nnpfc_out_colour_format_idc`가 1과 동일하면 크로마 샘플들의 위치가 알려지지 않거나 명시되지 않거나 다른 수단에 의해 명시됨을 나타낼 수 있다.

`nnpfc_chroma_sample_loc_type_frame`의 값은 0 내지 6의 범위에 존재해야 한다.

[271] `nnpfc_overlap`은 NNPF의 인접한 입력 텐서의 오버랩핑(overlapping) 수평 및 수직 샘플 수를 나타낼 수 있다. `nnpfc_overlap`의 값은 0 내지 16383의 범위에 존재해야 한다. `nnpfc_constant_patch_size_flag`의 값 1은 NNPF가 `nnpfc_patch_width_minus1` 및 `nnpfc_patch_height_minus1`에 의해 표시된 패치 크기를 입력으로 정확히 허용함을 나타낼 수 있다. `nnpfc_constant_patch_size_flag`의 값 0은 NNPF가, 너비가 `inpPatchWidth`이고 높이가 `inpPatchHeight`인 모든 패치 크기를 입력으로 허용함을 나타낼 수 있다. 이는 `inpPatchWidth + 2 * nnpfc_overlap`과 동일한 확장된 패치(즉, 패치 + 오버랩핑 영역)의 너비가 `nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2 * nnpfc_overlap`의 양의 정수배가 되고, `inpPatchHeight + 2 * nnpfc_overlap`과 동일한 확장된 패치의 높이가 `nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2 * nnpfc_overlap`의 양의 정수배가 되도록 할 수 있다.

[272] `nnpfc_patch_width_minus1 + 1`은 `nnpfc_constant_patch_size_flag`의 값이 1일 때 NNPF의 입력에 필요한 패치 크기의 수평 샘플 수를 나타낼 수 있다. `nnpfc_patch_width_minus1`의 값은 0 내지 $\text{Min}(32766, \text{CroppedWidth} - 1)$ 의 범위에 존재해야 한다.

[273] `nnpfc_patch_height_minus1 + 1`은 `nnpfc_constant_patch_size_flag`의 값이 1일 때 NNPF의 입력에 필요한 패치 크기의 수직 샘플 수를 나타낼 수 있다. `nnpfc_patch_height_minus1`의 값은 0 내지 $\text{Min}(32766, \text{CroppedHeight} - 1)$ 의 범위에 존재해야 한다.

[274] `nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2 * nnpfc_overlap`은 `nnpfc_constant_patch_size_flag`가 0일 때 NNPF에 대한 입력에 필요한 확장된 패치의 너비의 허용된 모든 값들의 공약수를 나타낼 수 있다. `nnpfc_extended_patch_width_cd_delta_minus1`의 값은 0 내지 $\text{Min}(32766, \text{CroppedWidth} - 1)$ 의 범위에 존재해야 한다.

[275] `nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2 * nnpfc_overlap`은 `nnpfc_constant_patch_size_flag`가 0일 때 NNPF에 대한 입력에 필요한 확장된 패치의 높이의 허용된 모든 값들의 공약수를 나타낼 수 있다.

nnpfc_extended_patch_height_cd_delta_minus1의 값은 0 내지 $\text{Min}(32766, \text{CroppedHeight} - 1)$ 의 범위에 존재해야 한다.

[276] 변수들 inpPatchWidth 및 inpPatchHeight는 각각 패치 크기 너비와 패치 크기 높이로 설정될 수 있다.

[277] nnpfc_constant_patch_size_flag의 값이 0이면 다음이 적용될 수 있다.

[278] - inpPatchWidth 및 inpPatchHeight의 값들은 외부 수단에 의해 제공되거나 사후 프로세서(post-processor) 자체에 의해 설정될 수 있다.

[279] - inpPatchWidth + 2 * nnpfc_overlap의 값은 ofnnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2 * nnpfc_overlap의 양의 정수배여야 하며 inpPatchWidth는 CroppedWidth보다 작거나 같아야 한다. inpPatchHeight + 2 * nnpfc_overlap의 값은

ofnnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2 * nnpfc_overlap의 양의 정수배여야 하며 inpPatchHeight는 CroppedHeight보다 작거나 같아야 한다.

[280] 그렇지 않으면(nnpfc_constant_patch_size_flag의 값이 1과 동일함), inpPatchWidth의 값은 nnpfc_patch_width_minus1 + 1과 동일하게 설정되고 inpPatchHeight의 값은 nnpfc_patch_height_minus1 + 1과 동일하게 설정될 수 있다.

[281] 변수들 outPatchWidth, outPatchHeight, horCScaling, verCScaling, outPatchCWidth, outPatchCHeight 및 overlapSize는 표 16과 같이 유도될 수 있다.

[282] [표16]

$$\text{outPatchWidth} = (\text{nnpfcOutputPicWidth} * \text{inpPatchWidth}) / \text{CroppedWidth}$$

$$\text{outPatchHeight} = (\text{nnpfcOutputPicHeight} * \text{inpPatchHeight}) / \text{CroppedHeight}$$

$$\text{horCScaling} = \text{SubWidthC} / \text{outSubWidthC}$$

$$\text{verCScaling} = \text{SubHeightC} / \text{outSubHeightC}$$

$$\text{outPatchCWidth} = \text{outPatchWidth} * \text{horCScaling}$$

$$\text{outPatchCHeight} = \text{outPatchHeight} * \text{verCScaling}$$

[283] outPatchWidth * CroppedWidth가 nnpfc_pic_width_in_luma_samples * inpPatchWidth와 같아야 하며, outPatchHeight * CroppedHeight가 nnpfc_pic_height_in_luma_samples * inpPatchHeight와 같아야 하는 것은 비트스트림 적합성의 요구 사항이다.

[284] nnpfc_padding_type은 표 17에 설명된 대로 입력 픽처의 경계 외부에 있는 샘플 위치를 참조할 때 패딩하는 프로세스를 나타낼 수 있다. nnpfc_padding_type의 값은 비트스트림에서 0 내지 4의 범위에 존재해야 한다. nnpfc_padding_type에

대한 5 내지 15의 값들은 향후 사용을 위해 예약될 수 있으며 비트스트림에는 존재하지 않는다. 디코더들은 5 내지 15의 범위에 있는 `nnpfc_padding_type`을 포함하는 NNPFC SEI 메시지를 무시해야 한다. 15보다 큰 `nnpfc_padding_type`의 값들은 비트스트림에 존재하지 않으며, 향후 사용을 위해 예약되지 않는다.

[285] [표17]

<code>nnpfc_padding_type</code>	Description
0	제로 패딩 (Zero padding)
1	복제 패딩 (Replication padding)
2	반사 패딩 (Reflection padding)
3	랩-어라운드 패딩 (Wrap-around padding)
4	고정 패딩 (Fixed padding)
5..15	예약됨

- [286] `nnpfc_luma_padding_val`은 `nnpfc_padding_type`의 값이 4일 때 패딩에 사용할 루마 값을 나타낼 수 있다. `nnpfc_luma_padding_val`의 값은 0 내지 $(1 \ll \text{BitDepth}_y) - 1$ 의 범위에 존재해야 한다.
- [287] `nnpfc_cb_padding_val`은 `nnpfc_padding_type`의 값이 4일 때 패딩에 사용할 Cb 값을 나타낼 수 있다. `nnpfc_cb_padding_val`의 값은 0 내지 $(1 \ll \text{BitDepth}_c) - 1$ 의 범위에 존재해야 한다.
- [288] `nnpfc_cr_padding_val`은 `nnpfc_padding_type`의 값이 4일 때 패딩에 사용할 Cr 값을 나타낼 수 있다. `nnpfc_cr_padding_val`의 값은 0 내지 $(1 \ll \text{BitDepth}_C) - 1$ 의 범위에 존재해야 한다.
- [289] 수직 샘플 위치 y , 수평 샘플 위치 x , 픽처 높이 `picHeight`, 픽처 너비 `picWidth`, 샘플 어레이 `CroppedPic`, 및 성분 인덱스 `cIdx` (루마에 대해 0과 같고, Cb에 대해 1과 같고, Cr에 대해 2와 같음)인 입력들을 가지는 함수 `InpSampleVal(y, x, picHeight, picWidth, CroppedPic)`는 표 18과 같이 유도된 `SampleVal`의 값을 반환할 수 있다.
- [290] 함수 `InpSampleVal()`에 대한 입력에 대하여, 일부 추론 엔진의 입력 텐서 규칙과의 호환성을 위해 수직 위치가 수평 위치 앞에 나열될 수 있다.

[291] [표18]

```

if( nnpfc_padding_type == 0 )
  if( y < 0 || x < 0 || y >= picHeight || x >= picWidth )
    sampleVal = 0
  else
    sampleVal = croppedPic[ x ][ y ]
else if( nnpfc_padding_type == 1 )
  sampleVal = croppedPic[ Clip3( 0, picWidth - 1, x ) ][ Clip3( 0, picHeight - 1, y ) ]
else if( nnpfc_padding_type == 2 )
  sampleVal = croppedPic[ Reflect( picWidth - 1, x ) ][ Reflect( picHeight - 1, y ) ]
else if( nnpfc_padding_type == 3 )
  if( y >= 0 && y < picHeight )
    sampleVal = croppedPic[ Wrap( picWidth - 1, x ) ][ y ]
else if( nnpfc_padding_type == 4 )
  if( y < 0 || x < 0 || y >= picHeight || x >= picWidth )
    sampleVal = ( cIdx == 0 ? nnpfc_luma_padding_val :
      ( cIdx == 1 ? nnpfc_cb_padding_val : nnpfc_cr_padding_val ) )
  else
    sampleVal = croppedPic[ x ][ y ]

```

[292] NNPF PostProcessingFilter()는 NNPF SEI 메시지의 시맨틱스들에서 유도된 대상 NNPF일 수 있다. 다음의 예제 프로세스는, 패치-와이즈 방식(patch-wise manner)으로 필터링된 및/또는 보간된 픽처(들)을 생성하기 위해 사용될 수 있다. 필터링된 및/또는 보간된 픽처(들)은 nnpfc_out_order_idc에 의해 지정된 바와 같이 Y 샘플 어레이 FilterdYPic, Cb 샘플 어레이 FilterdCbPic 및 Cr 샘플 어레이 FilterdCrPic을 포함할 수 있다.

[293] [표19]

```

if( nnpfc_inp_order_idc == 0 || nnpfc_inp_order_idc == 2 )
    for( cTop = 0; cTop < CroppedHeight; cTop += inpPatchHeight )
        for( cLeft = 0; cLeft < CroppedWidth; cLeft += inpPatchWidth ) {
            DeriveInputTensors( )
            outputTensor = PostProcessingFilter( inputTensor )
            StoreOutputTensors( )
        }
else if( nnpfc_inp_order_idc == 1 )
    for( cTop = 0; cTop < CroppedHeight / SubHeightC; cTop += inpPatchHeight )
        for( cLeft = 0; cLeft < CroppedWidth / SubWidthC; cLeft += inpPatchWidth ) {
            DeriveInputTensors( )
            outputTensor = PostProcessingFilter( inputTensor )
            StoreOutputTensors( )
        }
else if( nnpfc_inp_order_idc == 3 )
    for( cTop = 0; cTop < CroppedHeight; cTop += inpPatchHeight * 2 )
        for( cLeft = 0; cLeft < CroppedWidth; cLeft += inpPatchWidth * 2 ) {
            DeriveInputTensors( )
            outputTensor = PostProcessingFilter( inputTensor )
            StoreOutputTensors( )
        }

```

- [294] 인덱스 i 를 포함하는 NNPF-생성 픽처는, 존재하면 표 19에 의해 유도되는 샘플 어레이들 $FilteredYPic[i]$, $FilteredCbPic[i]$, 및 $FilteredCrPic[i]$ 를 포함할 수 있다. NNPF-생성 픽처는 오버랩 영역들을 포함하지 않을 수 있다.
- [295] NNPF 프로세스는 그들의 증가하는 인덱스 순서로 NNPF-생성 픽처를 출력하여 표 19에 의해 정의된 프로세스를 구성할 수 있다. 여기서, NNPF에 의해 보간된 모든 NNPF-생성 픽처들은 출력되고 NNPF에 대한 모든 입력 픽처들에 대응하는 NNPF-생성 픽처들은 NNPF SEI 메시지의 시멘틱스들에 지정된 대로 출력될 수 있다.
- [296] $nnpfc_complexity_info_present_flag$ 의 값 1은 $nnpfc_id$ 와 연관된 NNPF의 복잡도를 나타내는 하나 이상의 신택스 요소가 존재함을 나타낼 수 있다. $nnpfc_complexity_info_present_flag$ 의 값 0은 $nnpfc_id$ 와 연관된 NNPF의 복잡도를 나타내는 신택스 요소가 존재하지 않음을 나타낼 수 있다.
- [297] $nnpfc_parameter_type_idc$ 의 값 0은 신경망이 정수 파라미터들만을 사용함을 나타낼 수 있다. $nnpfc_parameter_type_flag$ 의 값 1은 신경망이 부동 소수점 또는 정수 파라미터들을 사용할 수 있음을 나타낼 수 있다. $nnpfc_parameter_type_idc$ 의 값 2는 신경망이 이진 파라미터들만을 사용함을 나타낼 수 있다. $nnpfc_parameter_type_idc$ 의 값 3은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 $nnpfc_parameter_type_idc$ 의 값이 3인 NNPF SEI 메시지를 무시해야 한다.
- [298] $nnpfc_log2_parameter_bit_length_minus3$ 의 값 0, 1, 2 및 3은 신경망이 각각 8, 16,

32 및 64보다 큰 비트 길이의 파라미터들을 사용하지 않음을 나타낼 수 있다. `nnpfc_parameter_type_idc`가 존재하고 `nnpfc_log2_parameter_bit_length_minus3`이 존재하지 않으면 신경망은 1보다 큰 비트 길이의 파라미터들을 사용하지 않을 수 있다.

- [299] `nnpfc_num_parameters_idc`는 NNPF에 대한 신경망 파라미터들의 최대 수를 2048의 거듭제곱 단위로 나타낼 수 있다. `nnpfc_num_parameters_idc`의 값 0은 신경망 파라미터들의 최대 수를 알 수 없음을 나타낼 수 있다. `nnpfc_num_parameters_idc`의 값은 0 내지 52 범위에 존재해야 한다. 52보다 큰 `nnpfc_num_parameters_idc`의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 52보다 큰 `nnpfc_num_parameters_idc`를 가진 NNPF SEI 메시지를 무시해야 한다.
- [300] `nnpfc_num_parameters_idc`의 값이 0보다 큰 경우, 변수 `maxNumParameters`는 수식 9와 같이 유도될 수 있다.
- [301] [수식 9]
- [302]
- $$\text{maxNumParameters} = (2048 \ll \text{nnpfc_num_parameters_idc}) - 1$$
- [303] NNPF의 신경망 파라미터들의 수가 `maxNumParameters` 보다 작거나 같아야 하는 것은 비트스트림 적합성의 요구사항이다.
- [304] 0보다 큰 `nnpfc_num_kmac_operations_idc`는 NNPF의 샘플 당 곱누산 작업의 최대 수(the maximum number of multiply-accumulate operations per sample)가 `nnpfc_num_kmac_operations_idc * 1000`보다 작거나 같음을 나타낼 수 있다. `nnpfc_num_kmac_operations_idc`의 값 0은 네트워크의 곱누산 작업의 최대 수를 알 수 없음을 나타낼 수 있다. `nnpfc_num_kmac_operations_idc`의 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [305] 0보다 큰 `nnpfc_total_kilobyte_size`는 신경망의 압축되지 않은 파라미터들을 저장하는 데 필요한 총 크기(kilobytes)를 나타낼 수 있다. 비트 단위의 총 크기는 각 파라미터를 저장하는 데 사용된 비트의 합보다 크거나 같은 숫자일 수 있다. `nnpfc_total_kilobyte_size`는 총 크기(비트)를 8000으로 나누고 반올림한 것일 수 있다. `nnpfc_total_kilobyte_size`의 값 0은 신경망에 대한 파라미터들을 저장하는 데 필요한 전체 크기를 알 수 없음을 나타낼 수 있다. `nnpfc_total_kilobyte_size`의 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [306] `nnpfc_metadata_extension_num_bits`의 값 0은 `nnpfc_reserved_metadata_extension`이 존재하지 않음을 나타낼 수 있다. 0보다 큰 `nnpfc_metadata_extension_num_bits`는 `nnpfc_reserved_metadata_extension`의 길이를 비트 단위로 나타낼 수 있다. `nnpfc_metadata_extension_num_bits`는 0과 동일해야 한다. `nnpfc_metadata_extension_num_bits`에 대한 1 내지 2048의 범위 값들은 향후 사용을 위해 예약될 수 있으며 비트스트림에서는 존재하지 않는다.

디코더들은 0 내지 2048의 범위에서 `nnpfc_metadata_extension_num_bits`의 모든 값을 허용해야 한다. 2048보다 큰 `nnpfc_metadata_extension_num_bits`의 값들은 비트스트림에 존재하지 않으며 향후 사용을 위해 예약되지 않는다.

- [307] `nnpfc_reserved_metadata_extension`은 비트스트림에서 존재하지 않는다. 그러나, 디코더들은 `nnpfc_reserved_metadata_extension`의 존재 및 값들을 무시해야 한다. 그것이 존재하는 경우 `nnpfc_reserved_metadata_extension`의 비트 단위의 길이는 `nnpfc_metadata_extension_num_bits`과 동일할 수 있다.
- [308] `nnpfc_reserved_zero_bit_b`는 비트스트림에서 0과 동일해야 한다. 디코더들은 `nnpfc_reserved_zero_bit_b`가 0이 아닌 NNPFC SEI 메시지를 무시해야 한다.
- [309] `nnpfc_payload_byte[i]`는 비트스트림의 i 번째 바이트를 포함할 수 있다. i 의 모든 존재하는 값에 대한 바이트 시퀀스 `nnpfc_payload_byte[i]`는 ISO/IEC 15938-17을 준수하는 완전한 비트스트림이어야 한다.
- [310] 신경망 포스트-필터 활성화(Neural-network post-filter activation, NNPFA)
- [311] NNPFA에 대한 선택스 구조가 표 20에 나타나 있다.
- [312] [표20]

<code>nn_post_filter_activation(payloadSize) {</code>	Descriptor
<code> nnpfa_target_id</code>	<code>ue(v)</code>
<code> nnpfa_cancel_flag</code>	<code>u(1)</code>
<code> if(!nnpfa_cancel_flag) {</code>	
<code> nnpfa_target_base_flag</code>	<code>u(1)</code>
<code> nnpfa_persistence_flag</code>	<code>u(1)</code>
<code> nnpfa_num_output_entries</code>	<code>ue(v)</code>
<code> for(i = 0; i < nnpfa_num_output_entries; i++)</code>	
<code> nnpfa_output_flag[i]</code>	<code>u(1)</code>
<code> }</code>	
<code> }</code>	

- [313] 표 20의 NNPFA 선택스 구조는 SEI 메시지 형태로 시그널링될 수 있다. 표 20의 NNPFA 선택스 구조를 시그널링하는 SEI 메시지는 NNPFA SEI 메시지라고 지칭될 수 있다.
- [314] NNPFA SEI 메시지는 픽처 세트의 후처리 필터링을 위해 `nnpfa_target_id`와 `nnpfc_base_flag`로 식별되는 대상 신경망 후처리 필터의 가능한 사용을 활성화하거나 비활성화할 수 있다. NNPFA가 활성화된 특정 픽처에 대해, 대상 NNPFA는 다음과 같이 도출될 수 있다.

- [315] - nnpfa_target_base_flag가 1이면, 대상 NNPf는 nnpfa_target_id와 동일한 nnpfc_id를 가지는 기본 NNPf이다.
- [316] - 그렇지 않은 경우(nnpfa_target_base_flag가 0인 경우), 대상 NNPf는 디코딩 순서 상 현재 픽처의 첫 번째 VCL NAL 유닛에 선행하고 기본 NNPf를 포함하는 NNPFC SEI 메시지의 반복이 아니면서, nnpfa_target_id와 동일한 nnpfc_id를 갖는 마지막 NNPFC SEI 메시지에 의해 특정되는 NNPf이다.
- [317] 후처리 필터는 다른 목적을 위해 사용되거나 다른 색상 성분을 필터링하는 경우에, 동일한 픽처에 대해 여러 개의 NNPFA SEI 메시지가 존재할 수 있다.
- [318] nnpfa_target_id는 현재 픽처와 관련되고 nnpfa_target_id와 동일한 nnpfc_id를 갖는 하나 이상의 NNPFC SEI 메시지에 의해 지정되는 대상 NNPf를 나타낼 수 있다.
- [319] nnpfa_target_id의 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다. 256 내지 511의 범위 및 2^{31} 내지 $2^{32} - 2$ 의 범위 내 nnpfa_target_id 값은 향후 사용을 위해 예약될 수 있다. 디코더들은 256 내지 511 또는 2^{31} 내지 $2^{32} - 2$ 범위의 nnpfa_target_id를 가진 NNPFA SEI 메시지를 무시해야 한다.
- [320] nnpfa_target_id의 특정 값을 가진 NNPFA SEI 메시지는 다음 조건 중 하나 또는 둘 모두가 참이 아닌 한 현재 PU(Picture Unit)에 존재하지 않아야 한다.
- [321] - 현재 CLVS 내에는 디코딩 순서에서 현재 PU 앞에 있는 PU에 존재하는 nnpfa_target_id의 특정 값과 동일한 nnpfc_id를 갖는 NNPFC SEI 메시지가 존재함
- [322] - 현재 PU에 nnpfa_target_id의 특정 값과 동일한 nnpfc_id를 갖는 NNPFC SEI 메시지가 존재함
- [323] PU가 특정 값의 nnpfc_id를 갖는 NNPFC SEI 메시지와, 특정 값의 nnpfc_id와 동일한 nnpfa_target_id를 갖는 NNPFA SEI 메시지를 모두 포함하는 경우, NNPFC SEI 메시지는 디코딩 순서에서 NNPFA SEI 메시지보다 선행해야 한다.
- [324] nnpfa_cancel_flag의 값 1은 현재 SEI 메시지와 동일한 nnpfa_target_id를 갖는 임의의 이전 NNPFA SEI 메시지에 의해 설정된 대상 신경망 후처리 필터의 지속성이 취소됨을 나타낼 수 있다. 즉, 대상 신경망 후처리 필터는 현재 SEI 메시지와 동일한 nnpfa_target_id 및 0과 동일한 nnpfa_cancel_flag를 갖는 다른 NNPFA SEI 메시지에 의해 활성화되지 않는 한, 더 이상 사용되지 않는다. nnpfa_cancel_flag의 값 0은 nnpfa_target_base_flag, nnpfa_persistence_flag 및 nnpfa_num_output_entries가 뒤따른다는 것을 나타낼 수 있다.
- [325] nnpfa_target_base_flag의 값 1은 대상 NNPf가 nnpfa_target_id와 동일한 nnpfc_id를 갖는 기본 NNPf임을 나타낼 수 있다. nnpfa_target_base_flag의 값 0은 대상 NNPf가 디코딩 순서 상 현재 픽처의 첫 번째 VCL NAL 유닛에 선행하는 nnpfa_target_id와 동일한 nnpfc_id를 갖는 마지막 NNPFC SEI 메시지에 의해 특정되는 NNPf임을 나타낼 수 있다.
- [326] nnpfa_persistence_flag는 현재 레이어에 대한 대상 신경망 후처리 필터의 지속성을 나타낼 수 있다. nnpfa_persistence_flag의 값 0은 대상 신경망 후처리

필터가 현재 픽처에 대한 후처리 필터링에만 사용될 수 있음을 나타낼 수 있다. `nnpfa_persistence_flag`의 값 1은 대상 신경망 후처리 필터가 다음 조건 중 하나 이상이 참일 때까지 출력 순서대로 현재 픽처와 현재 레이어 내 모든 후속 픽처들에 대한 후처리 필터링에 사용될 수 있음을 나타낼 수 있다.

- [327] - 현재 레이어의 새로운 CLVS가 시작됨
- [328] - 비트스트림이 종료됨
- [329] - 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 1과 동일한 `nnpfa_cancel_flag`를 갖는 NNPF SEI 메시지와 연관된 현재 레이어 내 픽처가 출력 순서에서 현재 픽처 다음에 출력됨
- [330] 대상 신경망 후처리 필터는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 1과 동일한 `nnpfa_cancel_flag`를 갖는 NNPF SEI 메시지와 연관된 현재 레이어 내 후속 픽처에는 적용되지 않는다.
- [331] `nnpfcTargetPictures`가 디코딩 순서 상 현재 NNPF SEI 메시지에 선행하면서 `nnpfa_target_id`와 동일한 `nnpfc_id`를 갖는 마지막 NNPF SEI 메시지와 관련된 픽처들의 세트인 것으로 한다. `nnpfaTargetPictures`는 현재 NNPF SEI 메시지에 의해 대상 NNPF가 활성화되는 픽처들의 세트인 것으로 한다. `nnpfaTargetPictures`에 포함되는 픽처는 `nnpfcTargetPictures`에도 포함되어야 한다는 것이 비트스트림 적합성 요건이다.
- [332] `nnpfa_num_output_entries`는 NNPF SEI 메시지에 존재하는 `nnpfa_output_flag[i]` 신텍스 요소들의 개수를 나타낼 수 있다. `nnpfa_num_output_entries`의 값은 0부터 `NumInpPicsInOutTensor`까지의 범위에 포함된다.
- [333] `nnpfa_output_flag[i]`의 값 1은 인덱스 `InpIdx[i]`를 갖는 입력 픽처에 대응되는 NNPF 생성 픽처가 이 NNPF SEI 메시지에 의해 활성화된 NNPF 프로세스에 의해 출력됨을 나타내고, 여기서 NNPF 프로세스는 NNPF SEI 메시지의 시맨틱스 내에서 특정될 수 있다. `nnpfa_output_flag[i]`의 값 0은 인덱스 `InpIdx[i]`를 갖는 입력 픽처에 대응되는 NNPF 생성 픽처가 이 NNPF SEI 메시지에 의해 활성화된 NNPF 프로세스에 의해 출력되지 않음을 나타낼 수 있다. `nnpfa_num_output_entries`가 `NumInpPicsInOutTensor`보다 작으면, `nnpfa_output_flag[i]`는 `nnpfa_num_output_entries`에서 `NumInpPicsInOutTensor - 1`까지의 범위에 있는 각 `i`의 값에 대해 1로 추론될 수 있다.
- [334] **포스트-필터 힌트(Post-filter hint)**
- [335] 포스트-필터 힌트에 대한 신텍스 구조가 표 21에 나타나 있다.

[336] [표21]

	Descriptor
post_filter_hint(payloadSize) {	
filter_hint_cancel_flag	u(1)
if(!filter_hint_cancel_flag) {	
filter_hint_persistence_flag	u(1)
filter_hint_size_y	ue(v)
filter_hint_size_x	ue(v)
filter_hint_type	u(2)
filter_hint_chroma_coeff_present_flag	u(1)
for(cIdx = 0; cIdx < (filter_hint_chroma_coeff_present_flag ? 3 : 1); cIdx++)	
for(cy = 0; cy < filter_hint_size_y; cy++)	
for(cx = 0; cx < filter_hint_size_x; cx++)	
filter_hint_value[cIdx][cy][cx]	se(v)
}	
}	

- [337] 표 21의 포스트-필터 힌트 신택스 구조는 SEI 메시지 형태로 시그널링될 수 있다. 표 21의 포스트-필터 힌트 신택스 구조를 시그널링하는 SEI 메시지는 포스트-필터 힌트 SEI 메시지라고 지칭될 수 있다.
- [338] 포스트-필터 힌트 SEI 메시지는 향상된 디스플레이 품질을 얻기 위해 디코딩 및 출력된 픽처 세트를 후처리에 잠재적으로 사용할 수 있도록 포스트-필터의 설계에 대한 포스트-필터 계수 또는 상관 정보를 제공할 수 있다.
- [339] filter_hint_cancel_flag의 값 1은 SEI 메시지가 현재 레이어에 적용되는 출력 순서 상 이전의 포스트-필터 힌트 SEI 메시지의 지속성을 취소함을 나타낼 수 있다. filter_hint_cancel_flag의 값 0은 포스트-필터 힌트 정보가 후속함을 나타낼 수 있다.
- [340] filter_hint_persistence_flag는 현재 레이어에 대한 포스트-필터 힌트 SEI 메시지의 지속성을 나타낼 수 있다. filter_hint_persistence_flag의 값 0은 포스트-필터 힌트가 현재 디코딩된 픽처에만 적용됨을 나타낼 수 있다. filter_hint_persistence_flag의 값 1은 포스트-필터 힌트 SEI 메시지가 현재 디코딩된 픽처에 적용되고 다음 조건 중 하나 이상이 참일 때까지 출력 순서에 따라 현재 레이어 내 모든 후속 픽처들에 대해 지속됨을 나타낼 수 있다.
- [341] - 현재 레이어의 새로운 CLVS가 시작됨
- [342] - 비트스트림이 종료됨
- [343] - 포스트-필터 힌트 SEI 메시지와 연관된 AU의 현재 레이어 내 픽처는 출력 순서에서 현재 픽처 다음에 출력됨

- [344] `filter_hint_size_y`는 필터 계수 또는 상관 어레이의 수직 크기를 나타낼 수 있다. `filter_hint_size_y`의 값은 1 내지 15의 범위에 존재해야 한다.
- [345] `filter_hint_size_x`는 필터 계수 또는 상관 어레이의 수평 크기를 나타낼 수 있다. `filter_hint_size_x`의 값은 1 내지 15의 범위에 존재해야 한다.
- [346] `filter_hint_type`은 표 19에 나타낸 바와 같이 전송된 필터 힌트의 유형을 나타낼 수 있다. `filter_hint_type`의 값은 0 내지 2의 범위에 존재해야 한다. 3과 동일한 `filter_hint_type` 값은 비트스트림에는 존재하지 않는다. 디코더들은 `filter_hint_type`이 3인 포스트-필터 힌트 SEI 메시지를 무시해야 한다.
- [347] [표22]

Value	Description
0	2D-FIR 필터의 계수들
1	1D-FIR 필터들의 계수들
2	Cross-correlation matrix

- [348] `filter_hint_chroma_coeff_present_flag`의 값 1은 크로마에 대한 필터 계수들이 존재함을 나타낼 수 있다. `filter_hint_chroma_coeff_present_flag`의 값 0은 크로마에 대한 필터 계수들이 존재하지 않음을 나타낼 수 있다.
- [349] `filter_hint_value[cIdx][cy][cx]`는 필터 계수, 또는 원본 신호와 디코딩된 신호 간의 상호 상관 행렬 요소를 16비트 정밀도로 나타낼 수 있다. `filter_hint_value[cIdx][cy][cx]`의 값은 $-2^{31} + 1$ 내지 $2^{31} - 1$ 범위에 존재해야 한다. `cIdx`는 관련 색상 요소를 나타내며, `cy`는 수직 방향의 카운터를 나타내고, `cx`는 수평 방향의 카운터를 나타낼 수 있다. `filter_hint_type`의 값에 따라 다음이 적용될 수 있다.
- [350] - `filter_hint_type`의 값이 0이면, `filter_hint_size_y * filter_hint_size_x` 크기의 2차원 FIR(Finite Impulse Response) 필터의 계수들이 전송될 수 있다.
- [351] - 그렇지 않고, `filter_hint_type`의 값이 1이면, 2개의 1차원 FIR 필터의 필터 계수들이 전송될 수 있다. 이 경우, `filter_hint_size_y`의 값은 2가 되어야 한다. 0인 인덱스 `cy`는 수평 필터의 필터 계수들을 나타내며, 1인 `cy`는 수직 필터의 필터 계수를 나타낼 수 있다. 필터링 과정에서는 수평 필터가 먼저 적용되고 그 결과는 수직 필터에 의해 필터링될 수 있다.
- [352] - 그렇지 않으면(`filter_hint_type`의 값이 2이면), 전송된 힌트는 원래 신호 `s`와 디코딩된 신호 `s'`사이의 상호 상관 행렬을 나타낼 수 있다.
- [353] `filter_hint_size_y * filter_hint_size_x` 크기의 `cIdx`로 식별된 관련 색상 성분에 대한 정규화된 상호 상관 행렬은 수식 10과 같이 정의될 수 있다.
- [354] [수식 10]

[355]

$$\text{filter_hint_value}(cIdx, cy, cx) = \frac{1}{(2^{8+\text{bitDepth}} - 1)^2 * h * w} \sum_{m=0}^{h-1} \sum_{n=0}^{w-1} s(m, n) * s'(m + cy - \text{OffsetY}, n + cx - \text{OffsetX})$$

[356]

수식 10에서, s는 원본 픽처의 색상 성분 cIdx의 샘플 어레이를 나타내며, s'는 이에 대응하는 디코딩된 픽처의 어레이를 나타내고, h는 관련 색상 성분의 수직 높이는 나타내며, w는 관련 색상 성분의 수평 너비를 나타내고, bitDepth는 색상 성분의 비트 뎁스를 나타낸다. 또한, OffsetY는 (filter_hint_size_y >> 1)와 동일하고, OffsetX는 (filter_hint_size_x >> 1)와 동일하고, cy의 범위는 0 <= cy < filter_hint_size_y이고, cx의 범위는 0 <= cx < filter_hint_size_x이다.

[357]

디코더는 원본 신호와 디코딩된 신호의 cross-correlation matrix 및 디코딩된 신호의 auto-cross-correlation matrix로부터 Wiener 포스트-필터를 유도할 수 있다.

[358]

종래 기술의 문제점

[359]

NNPFC 및 NNPFA SEI 메시지들의 현재 구성에서, NNPFA SEI 메시지는 nnpfa_target_base_flag의 값에 기반하여 기본 필터(즉, 1과 동일한 nnpfc_base_flag를 포함하는 NNPFA SEI 메시지) 또는 업데이트 필터(예를 들어, 0과 동일한 nnpfc_base_flag를 포함하는 NNPFA SEI 메시지)를 활성화할 수 있다.

[360]

nnpfa_target_base_flag는 nnpfa_cancel_flag의 값이 0일 때만 시그널링될 수 있다. 이는 nnpfa_target_id은 동일하지만 nnpfa_target_base_flag의 값이 서로 다른, 2개의 NNPF 활성화가 있는 경우 어떤 NNPF 활성화가 취소되어야 하는지를 식별하는데 문제점이 발생될 수 있다.

[361]

다시 말해서, 동일한 nnpfc_id를 가지는 기본 NNPF 및 업데이트 NNPF가 모두 동일한 픽처에 대해 활성화되면 이러한 경우가 허용되지 않는 한 1과 동일한 nnpfa_cancel_flag를 가지는 동일한 nnpfa_target_id를 포함하는 새로운 NNPFA SEI 메시지가 나중에 존재할 때 이 두가지 중 어떠한 지속성이 취소될지 구별하는 방법이 명확하지 않다.

[362]

실시예

[363]

본원은 위에서 설명한 문제점에 대한 해결방안을 제공한다. 각 실시예는 개별적으로 수행되거나 둘 이상이 서로 조합하여 수행될 수 있다.

[364]

본원에 의해 제안되는 실시예들을 요약하면 아래와 같다.

[365]

1. nnpfa_cancel_flag 값의 조건 없이 nnpfa_target_base_flag를 시그널링할 수 있다.

[366]

2. nnpfa_cancel_flag 값이 1과 동일하면, 동일한 값의 nnpfa_target_id 및 nnpfa_target_base_flag를 포함하는 이전에 활성화된 대상 NNPF의 지속성을 취소할 수 있다.

[367]

3. 대안적으로, 특정 nnpfa_target_id 및 nnpfa_target_base_flag를 포함하는 NNPFA가 존재하면, 출력 순서에 따라 동일한 값의 nnpfa_target_id 및 nnpfa_target_base_flag를 가지는 이전 NNPFA SEI 메시지의 지속성을 취소할 수 있다.

- [368] 4. 또 다른 대안으로, `nnpfa_target_base_flag`의 값과 관계없이 동일한 픽처에 대해 활성화된 2개(또는 그 이상)의 NNPF SEI 메시지(즉, 0과 동일한 `nnpfa_cancel_flag`를 포함함)가 없도록 제약할 수 있다.
- [369] 다시 말해서, `nnpfa_target_base_flag`의 값과 관계없이, 동일한 값의 `nnpfa_target_id`를 가지는 `nnpfa_cancel_flag`가 0인, 두 개의 NNPF SEI 메시지가 활성화되지 않는다.
- [370] 실시예 1
- [371] 실시예 1은 위에서 설명된 요약 1 및 요약 2에 대한 설명을 제공한다. 다음과 같이 표가 업데이트될 수 있다.
- [372] [표23]

<code>nn_post_filter_activation(payloadSize) {</code>	Descriptor
<code> nnpfa_target_id</code>	<code>ue(v)</code>
<code> nnpfa_target_base_flag</code>	<code>u(1)</code>
<code> nnpfa_cancel_flag</code>	<code>u(1)</code>
<code> if(!nnpfa_cancel_flag) {</code>	
<code> nnpfa_persistence_flag</code>	<code>u(1)</code>
<code> nnpfa_num_output_entries</code>	<code>ue(v)</code>
<code> for(i = 0; i < nnpfa_num_output_entries; i++)</code>	
<code> nnpfa_output_flag[i]</code>	<code>u(1)</code>
<code> }</code>	
<code> }</code>	
<code>}</code>	

- [373] NNPF SEI 메시지는 픽처들의 세트의 후처리 필터링을 위해 `nnpfa_target_id` 및 `nnpfa_target_base_flag`에 의해 식별되는 대상 NNPF의 가능한 사용을 활성화하거나 비활성화할 수 있다. `nnpfa_target_base_flag`는 표 23과 같이 `nnpfa_cancel_flag` 값의 조건 없이 파싱될 수 있다. 따라서, NNPF가 활성화되는 특정 픽처에 대하여 대상 NNPF는 다음과 같이 유도될 수 있다.
- [374] - `nnpfa_target_base_flag`가 1과 동일하면 대상 NNPF는 `nnpfc_id`가 `nnpfa_target_id`와 동일한 기본 NNPF일 수 있다.
- [375] - 그렇지 않으면 (`nnpfa_target_base_flag`가 0과 동일함) 대상 NNPF는 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는, `nnpfa_target_id`와 동일한 `nnpfc_id`를 갖는 마지막 NNPF SEI 메시지에 의해서 지정되는 NNPF이고, 기본 NNPF를 포함하는 NNPF SEI 메시지의 반복이 아닐 수 있다.
- [376] 예를 들어, NNPF가 다른 목적들 또는 다른 색상 성분들의 필터링을 위한 것을

의미하는 경우 동일한 픽처에 대하여 존재하는 몇몇의 NNPFA SEI 메시지가 존재할 수 있다.

- [377] `nnpfa_target_id`는 현재 픽처에 관련되고 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 하나 이상의 NNPFC SEI 메시지에 의해서 명시되는 대상 NNPFA를 나타낼 수 있다. `nnpfa_target_id` 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [378] `nnpfa_target_id`의 특정 값을 가지는 NNPFA SEI 메시지는 다음 조건 중 하나 또는 둘 모두가 참인 경우를 제외하고는 현재 PU에 존재하지 않아야 한다.
- [379] - 현재 CLVS 내에는 디코딩 순서에서 현재 PU에 선행하는 PU에 존재하는 `nnpfa_target_id`의 특정 값과 동일한 `nnpfc_id`를 가지는 NNPFC SEI 메시지가 존재할 수 있다.
- [380] - 현재 PU에서 `nnpfa_target_id`의 특정 값과 동일한 `nnpfc_id`를 가지는 NNPFC SEI 메시지가 존재할 수 있다.
- [381] PU가 `nnpfc_id`의 특정 값을 가지는 NNPFC SEI 메시지 및 `nnpfc_id`의 상기 특정 값과 동일한 `nnpfa_target_id`를 가지는 NNPFA SEI 메시지를 포함하면 NNPFC SEI 메시지는 디코딩 순서상 NNPFA SEI 메시지보다 선행해야 한다.
- [382] `nnpfa_target_base_flag`의 값 1은 대상 NNPFA가 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 기본 NNPFA인 것을 나타낼 수 있다. `nnpfa_target_base_flag`의 값 0은 대상 NNPFA가 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 마지막 NNPFC SEI 메시지에 의해 지정된 NNPFA이며 기본 NNPFA를 포함하는 NNPFC SEI 메시지의 반복이 아닌 것을 나타낼 수 있다.
- [383] `nnpfa_cancel_flag`의 값 1은 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 포함하는 임의의 이전 NNPFA SEI 메시지에 의해 설정된 대상 NNPFA의 지속성이 취소됨을 나타낼 수 있다. 즉, 대상 NNPFA는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 0인 또 다른 NNPFA SEI 메시지에 의해 활성화되지 않는 한 더 이상 사용되지 않을 수 있다. `nnpfa_cancel_flag`의 값 0은 `nnpfa_persistence_flag` 및 `nnpfa_num_output_entries`가 뒤따른다는 것을 나타낼 수 있다.
- [384] `nnpfa_persistence_flag`는 현재 레이어에 대한 대상 NNPFA의 지속성을 나타낼 수 있다.
- [385] `nnpfa_persistence_flag`의 값 0은 대상 NNPFA가 현재 픽처만을 후처리 필터링하기 위해 사용될 수 있음을 나타낼 수 있다.
- [386] `nnpfa_persistence_flag`의 값 1은 다음의 조건 중 하나 이상이 참일 때까지 대상 NNPFA가 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들에 대한 후처리 필터링을 위해 사용될 수 있음을 나타낼 수 있다.
- [387] - 현재 레이어의 새로운 CLVS가 시작됨
- [388] - 비트스트림이 종료됨
- [389] - 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고

nnpfa_cancel_flag가 1인 NNPF SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력됨.

- [390] 대상 NNPF는 현재 NNPF SEI 메시지와 동일한 nnpfa_target_id 및 nnpfa_target_base_flag를 가지고 nnpfa_cancel_flag가 1인 NNPF SEI 메시지에 관련된 현재 레이어의 후속 픽처에 적용되지 않을 수 있다. 다시 말해서, nnpfa_persistence_flag 값이 1인 경우 대상 NNPF는 현재 SEI 메시지와 동일한 nnpfa_target_id 및 nnpfa_target_base_flag를 가지는 NNPF SEI 메시지에 포함된 nnpfa_cancel_flag 값이 1인 것에 기반하여 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력되기 전까지 사용될 수 있다.
- [391] nnpfcTargetPictures는 디코딩 순서상 현재 NNPF SEI 메시지에 선행하는 nnpfa_target_id와 동일한 nnpfc_id를 가지는 마지막 NNPF SEI 메시지와 관련되는 픽처들의 세트로 설정될 수 있다. nnpfaTargetPictures는 대상 NNPF가 현재 NNPF SEI 메시지에 의해 활성화되는 픽처들의 세트로 설정될 수 있다. nnpfaTargetPictures에 포함된 모든 픽처들이 nnpfcTargetPictures에도 포함되어야 한다는 것이 비트스트림의 적합성의 요구사항이다.
- [392] 실시예 2
- [393] 실시예 2는 위에서 설명된 요약 1 및 3에 대한 설명을 제공한다. 다음과 같이 표가 업데이트될 수 있다.
- [394] [표24]

	Descriptor
nm_post_filter_activation(payloadSize) {	
nnpfa_target_id	ue(v)
nnpfa_target_base_flag	u(1)
nnpfa_cancel_flag	u(1)
if(!nnpfa_cancel_flag) {	
nnpfa_persistence_flag	u(1)
nnpfa_num_output_entries	ue(v)
for(i = 0; i < nnpfa_num_output_entries; i++)	
nnpfa_output_flag[i]	u(1)
}	
}	

- [395] NNPF SEI 메시지는 픽처들의 세트의 후처리 필터링을 위해 nnpfa_target_id 및 nnpfa_target_base_flag에 의해 식별되는 대상 NNPF의 가능한 사용을 활성화하거나 비활성화할 수 있다. 실시예 1과 유사하게 nnpfa_target_base_flag는

- 표 24와 같이 `nnpfa_cancel_flag` 값의 조건 없이 파싱될 수 있다. 따라서, NNPF가 활성화된 특정 픽처에 대하여 대상 NNPF는 다음과 같이 유도될 수 있다.
- [396] - `nnpfa_target_base_flag`가 1과 동일하면 대상 NNPF는 `nnpfc_id`가 `nnpfa_target_id`와 동일한 기본 NNPF일 수 있다.
- [397] - 그렇지 않으면 (`nnpfa_target_base_flag`가 0과 동일함) 대상 NNPF는 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는 `nnpfa_target_id`와 동일한 `nnpfc_id`를 갖는 마지막 NNPF SEI 메시지에 의해서 지정되는 NNPF이고, 기본 NNPF를 포함하는 NNPF SEI 메시지의 반복이 아닐 수 있다.
- [398] 예를 들어, NNPF가 다른 목적들 또는 다른 색상 성분들의 필터링을 위한 것으로 의미하는 경우 동일한 픽처에 대하여 존재하는 몇몇의 NNPF SEI 메시지가 존재할 수 있다.
- [399] `nnpfa_target_id`는 현재 픽처에 연관되고 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 하나 이상의 NNPF SEI 메시지에 의해서 지시되는 대상 NNPF를 나타낼 수 있다. `nnpfa_target_id` 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [400] `nnpfa_target_id`의 특정 값을 가지는 NNPF SEI 메시지는 다음 조건 중 하나 또는 둘 모두가 참인 경우를 제외하고는 현재 PU에 존재하지 않아야 한다.
- [401] - 현재 CLVS 내에는 디코딩 순서에서 현재 PU에 선행하는 PU에 존재하는 `nnpfa_target_id`의 특정 값과 동일한 `nnpfc_id`를 가지는 NNPF SEI 메시지가 존재할 수 있다.
- [402] - 현재 PU에서 `nnpfa_target_id`의 특정 값과 동일한 `nnpfc_id`를 가지는 NNPF SEI 메시지가 존재할 수 있다.
- [403] PU가 `nnpfc_id`의 특정 값을 가지는 NNPF SEI 메시지 및 `nnpfc_id`의 상기 특정 값과 동일한 `nnpfa_target_id`를 가지는 NNPF SEI 메시지를 포함하면 NNPF SEI 메시지는 디코딩 순서상 NNPF SEI 메시지보다 선행해야 한다.
- [404] `nnpfa_target_base_flag`의 값 1은 대상 NNPF가 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 기본 NNPF인 것을 나타낼 수 있다. `nnpfa_target_base_flag`의 값 0은 대상 NNPF가 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 마지막 NNPF SEI 메시지에 의해 지정된 NNPF이며 기본 NNPF를 포함하는 NNPF SEI 메시지의 반복이 아닌 것을 나타낼 수 있다.
- [405] `nnpfa_cancel_flag`의 값 1은 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 포함하는 임의의 이전 NNPF SEI 메시지에 의해 설정된 대상 NNPF의 지속성이 취소됨을 나타낼 수 있다. 즉, 대상 NNPF는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 0인 또 다른 NNPF SEI 메시지에 의해 활성화되지 않는 한 더 이상 사용되지 않을 수 있다. `nnpfa_cancel_flag`의 값 0은 `nnpfa_persistence_flag` 및 `nnpfa_num_output_entries`가 뒤따른다는 것을 나타낼 수 있다.
- [406] `nnpfa_persistence_flag`는 현재 레이어에 대한 대상 NNPF의 지속성을 나타낼 수

있다.

- [407] nnpfa_persistence_flag의 값 0은 대상 NNPF가 현재 픽처만을 후처리 필터링하기 위해 사용될 수 있음을 나타낼 수 있다.
- [408] nnpfa_persistence_flag의 값 1은 다음의 조건 중 하나 이상이 참일 때까지 대상 NNPF가 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들에 대한 후처리 필터링을 위해 사용될 수 있음을 나타낼 수 있다.
- [409] - 현재 레이어의 새로운 CLVS가 시작됨
- [410] - 비트스트림이 종료됨
- [411] - 현재 SEI 메시지와 동일한 nnpfa_target_id 및 nnpfa_target_base_flag를 가지는 NNPF SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력됨.
- [412] 대상 NNPF는 현재 NNPF SEI 메시지와 동일한 nnpfa_target_id 및 nnpfa_target_base_flag를 가지는 NNPF SEI 메시지에 관련된 현재 레이어의 후속 픽처에 적용되지 않을 수 있다. 다시 말해서, nnpfa_persistence_flag 값이 1인 경우 대상 NNPF는 현재 SEI 메시지와 동일한 nnpfa_target_id 및 nnpfa_target_base_flag를 가지는 NNPF SEI 메시지에 포함된 nnpfa_cancel_flag의 값과 상관없이 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력될 때까지 사용될 수 있다.
- [413] nnpfcTargetPictures를 디코딩 순서상 현재 NNPF SEI 메시지에 선행하는 nnpfa_target_id와 동일한 nnpfc_id를 가지는 마지막 NNPF SEI 메시지와 관련되는 픽처들의 세트가 되도록 할 수 있다. nnpfaTargetPictures를 대상 NNPF가 현재 NNPF SEI 메시지에 의해 활성화되는 픽처들의 세트가 되도록 할 수 있다. nnpfaTargetPictures에 포함된 모든 픽처들이 nnpfcTargetPictures에도 포함되어야 한다는 것이 비트스트림의 적합성의 요구사항이다.
- [414] **실시예 3**
- [415] 실시예 3은 위에서 설명된 요약 4에 대한 설명을 제공한다. 다음과 같이 표가 업데이트될 수 있다.

[416] [표25]

	Descriptor
nm_post_filter_activation(payloadSize) {	
npfa_target_id	ue(v)
npfa_cancel_flag	u(1)
if(!npfa_cancel_flag) {	u(1)
npfa_target_base_flag	
npfa_persistence_flag	u(1)
npfa_num_output_entries	u(1)
for(i = 0; i < npfa_num_output_entries; i++)	ue(v)
npfa_output_flag[i]	
}	u(1)
}	

[417] NNPFA SEI 메시지는 픽처들의 세트의 후처리 필터링을 위해 `npfa_target_id` 및 `npfa_target_base_flag`에 의해 식별되는 대상 NNPFC의 가능한 사용을 활성화하거나 비활성화할 수 있다. NNPFC가 활성화된 특정 픽처에 대하여 대상 NNPFC는 다음과 같이 유도될 수 있다.

[418] - `npfa_target_base_flag`가 1과 동일하면 대상 NNPFC는 `npfc_id`가 `npfa_target_id`와 동일한 기본 NNPFC일 수 있다.

[419] - 그렇지 않으면 (`npfa_target_base_flag`가 0과 동일함) 대상 NNPFC는 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는 `npfa_target_id`와 동일한 `npfc_id`를 갖는 마지막 NNPFC SEI 메시지에 의해서 지정되는 NNPFC이고, 기본 NNPFC를 포함하는 NNPFC SEI 메시지의 반복이 아닐 수 있다.

[420] 예를 들어, NNPFC가 다른 목적들 또는 다른 색상 성분들의 필터링을 위한 것을 의미하는 경우 동일한 픽처에 대하여 존재하는 몇몇의 NNPFA SEI 메시지가 존재할 수 있다.

[421] `npfa_target_id`는 현재 픽처에 관련되고 `npfa_target_id`와 동일한 `npfc_id`를 가지는 하나 이상의 NNPFC SEI 메시지에 의해서 명시되는 대상 NNPFC를 나타낼 수 있다. `npfa_target_id` 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.

[422] `npfa_target_id`의 특정 값을 가지는 NNPFA SEI 메시지는 다음 조건 중 하나 또는 둘 모두가 참인 경우를 제외하고는 현재 PU에 존재하지 않아야 한다.

[423] - 현재 CLVS 내에는 디코딩 순서에서 현재 PU에 선행하는 PU에 존재하는 `npfa_target_id`의 특정 값과 동일한 `npfc_id`를 가지는 NNPFC SEI 메시지가 있을 수 있다.

- [424] - 현재 PU에서 `nnpfa_target_id`의 특정 값과 동일한 `nnpfc_id`를 가지는 NNPF SEI 메시지가 있을 수 있다.
- [425] PU가 `nnpfc_id`의 특정 값을 가지는 NNPF SEI 메시지 및 `nnpfc_id`의 상기 특정 값과 동일한 `nnpfa_target_id`를 가지는 NNPF SEI 메시지를 포함하면 NNPF SEI 메시지는 디코딩 순서상 NNPF SEI 메시지보다 선행해야 한다.
- [426] `nnpfa_cancel_flag`의 값 1은 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 포함하는 임의의 이전 NNPF SEI 메시지에 의해 설정된 대상 NNPF의 지속성이 취소됨을 나타낼 수 있다. 즉, 대상 NNPF는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 0인 또 다른 NNPF SEI 메시지에 의해 활성화되지 않는 한 더 이상 사용되지 않을 수 있다. `nnpfa_cancel_flag`의 값 0은 `nnpfa_persistence_flag`, `nnpfa_persistence_flag` 및 `nnpfa_num_output_entries`가 뒤따른다는 것을 나타낼 수 있다.
- [427] `nnpfa_target_base_flag`의 값 1은 대상 NNPF가 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 기본 NNPF인 것을 나타낼 수 있다. `nnpfa_target_base_flag`의 값 0은 대상 NNPF가 디코딩 순서에서 현재 픽처의 첫 번째 VCL NAL 유닛에 선행하는 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 마지막 NNPF SEI 메시지에 의해 지정된 NNPF이며 기본 NNPF를 포함하는 NNPF SEI 메시지의 반복이 아닌 것을 나타낼 수 있다.
- [428] `nnpfa_target_base_flag`의 값에 관계없이 동일한 픽처에 대해서 두 개의 NNPF SEI 메시지 활성화(즉, 0과 동일한 `nnpfa_cancel_flag`를 포함함)가 없어야 한다는 제약이 있다.
- [429] `nnpfa_persistence_flag`는 현재 레이어에 대한 대상 NNPF의 지속성을 나타낼 수 있다.
- [430] `nnpfa_persistence_flag`의 값 0은 대상 NNPF가 현재 픽처만을 후처리 필터링하기 위해 사용될 수 있음을 나타낼 수 있다.
- [431] `nnpfa_persistence_flag`의 값 1은 다음의 조건 중 하나 이상이 참일 때까지 대상 NNPF가 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들에 대한 후처리 필터링을 위해 사용될 수 있음을 나타낼 수 있다.
- [432] - 현재 레이어의 새로운 CLVS가 시작됨
- [433] - 비트스트림이 종료됨
- [434] - 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 1인 NNPF SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력됨.
- [435] 대상 NNPF는 현재 NNPF SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 1인 NNPF SEI 메시지에 관련된 현재 레이어의 후속 픽처에 적용되지 않을 수 있다.
- [436] `nnpfcTargetPictures`는 디코딩 순서상 현재 NNPF SEI 메시지에 선행하는

nnpfa_target_id와 동일한 nnpfc_id를 가지는 마지막 NNPFC SEI 메시지와 관련되는 픽처들의 세트로 설정될 수 있다. nnpfaTargetPictures는 대상 NNPF가 현재 NNPF SEI 메시지에 의해 활성화되는 픽처들의 세트로 설정될 수 있다. nnpfaTargetPictures에 포함된 모든 픽처들이 nnpfcTargetPictures에도 포함되어야 한다는 것이 비트스트림의 적합성의 요구사항이다.

[437] 영상 부호화 방법 및 영상 복호화 방법

[438] 이하에서는, 본원의 다양한 실시예들에 따른 영상 부호화 방법과 영상 복호화 방법을 설명하도록 한다.

[439] 도 6은 영상 부호화 장치의 영상 부호화 방법을 설명하기 위한 도면이고, 도 7은 영상 복호화 장치의 영상 복호화 방법을 설명하기 위한 도면이다.

[440] 도 6을 참조하면, 영상 부호화 장치(200)는 후처리 필터로 이용될 수 있는 적어도 하나의 신경망에 대한 정보를 적어도 하나의 NNPFC SEI 메시지로 부호화할 수 있다(S610). 다시 말해서, 영상 부호화 장치(200)는 후처리 필터로 이용될 수 있는 적어도 하나의 신경망을 결정하고, 결정된 적어도 하나의 신경망에 대한 정보를 적어도 하나의 NNPFC SEI 메시지로 부호화할 수 있다.

[441] 영상 부호화 장치(200)는 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부가 결정된 대상 NNPF에 대한 정보들을 NNPFA SEI 메시지로 부호화할 수 있다(S620). 다시 말해서, 영상 부호화 장치(200)는 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 결정하고, 결정된 대상 NNPF에 대한 정보를 NNPFA SEI 메시지로 부호화할 수 있다.

[442] 대상 NNPF의 활성화 여부를 결정하는 과정은 현재 픽처에 적용될 수 있는 대상 NNPF를 결정하는 과정과, 대상 NNPF의 지속성을 취소할 지 여부를 결정하는 과정, 및 대상 NNPF의 지속성을 결정하는 과정 등을 포함할 수 있다. 예를 들어, 영상 부호화 장치(200)는 현재 픽처에 적용될 수 있는 대상 NNPF를 결정하고, 동일한 대상 NNPF 식별 정보를 가지는 이전 NNPFA SEI 메시지의 존재 여부에 따라 대상 NNPF의 지속성 취소 여부를 결정할 수 있다. 동일한 대상 NNPF 식별 정보를 가지는 이전 NNPFA SEI 메시지가 존재하면 영상 부호화 장치(200)는 해당 NNPFA SEI 메시지에 의해서 지정된 대상 NNPF의 지속성을 취소하도록 결정할 수 있다. 대상 NNPF의 지속성을 취소하지 않는 것으로 결정되면 영상 부호화 장치(200)는 대상 NNPF의 지속성을 결정할 수 있다. 이와 같이 결정된 대상 NNPF에 대한 정보는 NNPFA SEI 메시지로 부호화될 수 있다.

[443] 영상 부호화 장치(200)는 포스트-필터의 설계에 대한 포스트-필터 계수 또는 상관 정보 등을 포스트-필터 힌트 SEI 메시지로 부호화할 수 있다(S630). NNPFC SEI 메시지, NNPFA SEI 메시지 및/또는 포스트-필터 힌트 SEI 메시지는 NNPF SEI 메시지에 포함될 수 있다.

[444] 도 7을 참조하면, 영상 복호화 장치(300)는 현재 픽처에 적용될 NNPF에 대한 SEI 메시지를 비트스트림으로부터 획득될 수 있다. NNPF에 대한 SEI 메시지는 NNPFC SEI 메시지, NNPFA SEI 메시지 및/또는 포스트-필터 힌트 SEI 메시지를

포함할 수 있다.

- [445] 영상 복호화 장치(300)는 NNPF에 대한 SEI 메시지가 현재 픽처에 적용되는 경우에, NNPF에 대한 SEI 메시지에 포함된 적어도 하나의 NNPF SEI 메시지에 기반하여, 후처리 필터로 이용될 수 있는 적어도 하나의 신경망을 결정할 수 있다(S710).
- [446] 영상 복호화 장치(300)는 비트스트림으로부터 획득된 적어도 하나의 NNPF SEI 메시지에 기반하여, 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 결정할 수 있다(S720).
- [447] 대상 NNPF의 활성화 여부를 결정하는 과정은, 대상 NNPF를 결정하는 과정, 대상 NNPF의 지속성 취소 여부를 결정하는 과정 및 대상 NNPF의 지속성을 결정하는 과정 등을 포함할 수 있다. 예를 들어, 대상 NNPF의 활성화 여부를 결정하는 과정은, NNPF SEI 메시지에 기반하여 대상 NNPF를 결정하는 과정, 대상 NNPF의 지속성 취소 여부를 결정하는 과정, 및 대상 NNPF의 지속성을 결정하는 과정 등을 포함할 수 있다. 여기서, 대상 NNPF 식별 정보는 `nnpfa_target_id`를 포함하거나, 또는 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 포함할 수 있다.
- [448] 대상 NNPF는, 대상 NNPF를 나타내는 식별자인 `nnpfa_target_id`에 기반하여 결정될 수 있다. 다양한 실시예에서 대상 NNPF는 `nnpfa_target_id` 및 `nnpfa_target_base_flag`에 기반하여 결정될 수도 있다.
- [449] `nnpfa_target_id`는 현재 픽처에 관련되고 대상 NNPF 식별 정보와 동일한 필터 식별 정보(즉, `nnpfc_id`)를 가지는 하나 이상의 NNPF SEI 메시지에 의해서 명시되는 대상 NNPF를 나타낼 수 있다.
- [450] `nnpfa_target_base_flag`는 대상 NNPF가 대상 NNPF 식별 정보와 동일한 필터 식별 정보를 가지는 기본 NNPF 또는 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하고 동일한 식별 정보를 가지는 마지막 NNPF SEI 메시지에 의해서 지정된 NNPF인지 여부를 나타낼 수 있다. 다시 말해서, 대상 NNPF는 대상 필터 플래그에 기반하여 대상 NNPF 식별 정보와 동일한 필터 식별 정보를 가지는 기본 NNPF로 결정될 수 있다. 또한, 대상 NNPF는 대상 필터 플래그에 기반하여 디코딩 순서에서 현재 픽처의 첫번째 VCL NAL 유닛에 선행하는, 대상 NNPF 식별 정보와 동일한 필터 식별 정보를 가지는, 마지막 NNPF SEI 메시지에 의해 지정된 NNPF로 결정될 수 있다. 이때, 대상 NNPF는 기본 NNPF를 포함하는 NNPF SEI 메시지의 반복이 아닐 수 있다.
- [451] 영상 복호화 장치(300)는 현재 NNPF SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 이전 NNPF SEI 메시지가 존재하는지 여부를 결정하고, 현재 NNPF SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 이전 NNPF SEI 메시지가 존재하면 대상 NNPF의 지속성 취소 여부를 결정할 수 있다. 이 경우, 영상 복호화 장치(300)는 대상 NNPF의 지속성에 대한 취소 여부를 나타내는 지속성 취소 플래그인 `nnpfa_cancel_flag`에 기반하여 대상 NNPF의 지속성에 대한

취소 여부를 결정할 수 있다.

- [452] `nnpfa_cancel_flag`는 현재 SEI 메시지(예를 들어, 현재 NNPFA SEI 메시지)와 동일한 대상 NNPf 식별 정보를 가지는 이전 NNPFA SEI 메시지에 의해서 설정되는 대상 NNPf의 지속성에 대한 취소 여부를 나타낼 수 있다. 예를 들어, `nnpfa_cancel_flag` 값이 1이면 현재 SEI 메시지와 동일한 대상 NNPf 식별 정보 및 대상 필터 플래그를 포함하는 이전 NNPFA SEI 메시지에 의해서 설정되는 대상 NNPf의 지속성이 취소될 수 있다. 이때, 대상 NNPf는 현재 SEI 메시지와 동일한 대상 NNPf 식별 정보 및 대상 필터 플래그를 가지고, `nnpfa_cancel_flag` 값이 0인 또 다른 NNPFA SEI 메시지에 의해 활성화되지 않는 한 더 이상 사용되지 않을 수 있다. 예를 들어, `nnpfa_cancel_flag` 값이 0이면 현재 레이어에 대한 대상 NNPf의 지속성을 나타내는 플래그인 `nnpfa_persistence_flag` 및 `nnpfa_target_base_flag`가 유도될 수 있다.
- [453] 다양한 실시예에 따라 `nnpfa_cancel_flag` 값이 0이면 `nnpfa_target_base_flag`, `nnpfa_persistence_flag` 및 `nnpfa_num_output_entries`가 유도될 수도 있다.
- [454] 영상 복호화 장치(300)는 `nnpfa_persistence_flag` 값에 기반하여 대상 NNPf의 지속성을 결정할 수 있다. 예를 들어, `nnpfa_persistence_flag`가 0이면 대상 NNPf은 현재 픽처를 위해서만 사용될 수 있다. 예를 들어, `nnpfa_persistence_flag`가 1이면 대상 NNPf은 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들을 위해서 사용될 수 있다. 이때, 대상 NNPf은 현재 레이어의 새로운 CLVS가 시작되거나, 비트스트림이 종료되거나, 또는 현재 NNPFA SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 1인 NNPFA SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력되는 것 중 하나 이상이 충족될 때까지 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들을 위해서 사용될 수 있다. 다시 말해서, 대상 NNPf의 지속성은 현재 NNPFA SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지고 `nnpfa_cancel_flag`가 1인 NNPFA SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력되는지에 기반하여 결정될 수 있다.
- [455] 다양한 실시예에 따라 `nnpfa_persistence_flag`가 1이면 대상 NNPf은 현재 NNPFA SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지는 NNPFA SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력될 때까지 출력 순서상 현재 픽처 및 현재 레이어의 모든 후속 픽처들을 위해서 사용될 수 있다. 다시 말해서, 대상 NNPf의 지속성은 현재 레이어의 픽처에 관련된 NNPFA SEI 메시지에 포함된 `nnpfa_cancel_flag` 값과 상관없이 현재 NNPFA SEI 메시지와 동일한 `nnpfa_target_id` 및 `nnpfa_target_base_flag`를 가지는 NNPFA SEI 메시지에 관련된 현재 레이어의 픽처가 출력 순서상 현재 픽처 다음에 출력되는지에 기반하여 결정될 수 있다.
- [456] 다양한 실시예에 따르면, `nnpfa_target_base_flag` 값과 상관없이 동일 픽처에

- 대한 둘 이상의 NNPFA SEI 메시지들은 활성화되지 않을 수 있다.
- [457] 다양한 실시예에 따르면, 대상 NNPFA는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 1과 동일한 `nnpfa_cancel_flag`를 가지는 NNPFA SEI 메시지와 연관된 현재 레이어의 후속 픽처에 대하여 적용되지 않을 수 있다.
- [458] 도 8은 대상 NNPFA의 활성화 여부를 결정하는 실시예에 대한 영상 복호화 방법을 설명하기 위한 도면이다.
- [459] 도 8을 참조하면, 영상 복호화 장치(300)는 NNPFA SEI 메시지에 기반하여 대상 NNPFA를 결정할 수 있다(S800). 예를 들어, 영상 복호화 장치(300)는 NNPFA SEI 메시지의 대상 NNPFA 식별 정보(즉, `nnpfa_target_id`)에 기반하여 대상 NNPFA를 결정할 수 있다. 예를 들어, 영상 복호화 장치(300)는 현재 픽처와 관련되고 `nnpfa_target_id` 값과 동일한 `nnpfc_id` 값을 갖는 하나 이상의 NNPFA SEI 메시지에 의해 지정되는 대상 NNPFA를 결정할 수 있다.
- [460] 영상 복호화 장치(300)는 대상 NNPFA의 지속성 취소 여부를 결정할 수 있다(S810). 예를 들어, 영상 복호화 장치(300)는 대상 NNPFA의 지속성에 대한 취소 여부를 나타내는 지속성 취소 플래그(즉, `nnpfa_cancel_flag`)에 기반하여 대상 NNPFA의 지속성에 대한 취소 여부를 결정할 수 있다.
- [461] 예를 들어, 지속성 취소 플래그 값이 1이면 영상 복호화 장치(300)는 현재 SEI 메시지와 동일한 대상 NNPFA 식별 정보를 가지는 임의의 이전 NNPFA SEI 메시지에 의해서 설정된 대상 NNPFA의 지속성을 취소한다고 결정할 수 있다. 다시 말해서, 대상 NNPFA는 현재 SEI 메시지와 동일한 대상 NNPFA 식별 정보를 가지고 지속성 취소 플래그 값이 0인 또 다른 NNPFA SEI 메시지에 의해서 활성화되지 않는 한 더 이상 사용되지 않을 수 있다. 예를 들어, 지속성 취소 플래그 값이 0이면 영상 복호화 장치(300)는 대상 NNPFA의 지속성을 취소하지 않는다고 결정할 수 있다.
- [462] 영상 복호화 장치(300)는 대상 NNPFA의 지속성을 취소하지 않는 것에 기반하여 대상 NNPFA의 지속성을 결정할 수 있다(S830). 예를 들어, 대상 NNPFA의 지속성을 취소하지 않으면 영상 복호화 장치(300)는 대상 NNPFA를 현재 픽처에 대한 후처리 필터링에만 사용하는 것으로 결정하거나, 또는 상술한 조건이 충족될 때까지 대상 NNPFA를 출력 순서대로 현재 픽처와 현재 레이어 내 모든 후속 픽처들에 대한 후처리 필터링에 사용하는 것으로 결정할 수 있다.
- [463] 이처럼, 대상 NNPFA의 지속성이 결정되면 영상 복호화 장치(300)는 대상 NNPFA의 활성화를 결정 또는 지속할 수 있다.
- [464] 도 9 내지 도 11은 대상 NNPFA의 지속성 취소 여부에 대한 다양한 실시예들을 설명하기 위한 도면이다.
- [465] 도 9에 나타낸 바와 같이, 현재 NNPFA SEI 메시지의 `nnpfa_cancel_flag` 값이 1이면 현재 NNPFA SEI 메시지와 동일한 대상 NNPFA 식별 정보(즉, `nnpfa_target_id` 값이 1이고, `nnpfa_target_base_flag` 값이 1임)를 가지는 이전 NNPFA SEI 메시지가 존재하는 경우 `nnpfc_id` 값이 1이고, `nnpfc_base_flag` 값이

- 1인 NNPF SEI 메시지에 의해서 지정된 대상 NNPF에 대한 지속성이 취소될 수 있다.
- [466] 도 10에 나타낸 바와 같이, 현재 NNPF SEI 메시지와 동일한 대상 NNPF 식별 정보(즉, `nnpfa_target_id` 값이 1이고, `nnpfa_target_base_flag` 값이 1임)를 가지는 이전 NNPF SEI 메시지가 존재하는 경우 현재 NNPF SEI 메시지에 포함된 `nnpfa_cancel_flag` 값과 상관없이 대상 NNPF에 대한 지속성이 취소될 수 있다.
- [467] 도 11에 나타낸 바와 같이, 현재 NNPF SEI 메시지와 동일한 대상 NNPF 식별 정보(즉, `nnpfa_target_id` 값이 1임)를 가지는 이전 NNPF SEI 메시지가 존재하는 경우 대상 NNPF에 대한 지속성이 취소될 수 있다.
- [468] 도 12는 본 개시에 따른 실시예가 적용될 수 있는 콘텐츠 스트리밍 시스템을 예시적으로 나타낸 도면이다.
- [469] 도 12에 도시된 바와 같이, 본 개시의 실시예가 적용된 콘텐츠 스트리밍 시스템은 크게 인코딩 서버, 스트리밍 서버, 웹 서버, 미디어 저장소, 사용자 장치 및 멀티미디어 입력 장치를 포함할 수 있다.
- [470] 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.
- [471] 상기 비트스트림은 본 개시의 실시예가 적용된 영상 부호화 방법 및/또는 영상 부호화 장치에 의해 생성될 수 있고, 상기 스트리밍 서버는 상기 비트스트림을 전송 또는 수신하는 과정에서 일시적으로 상기 비트스트림을 저장할 수 있다.
- [472] 상기 스트리밍 서버는 웹 서버를 통한 사용자 요청에 기반하여 멀티미디어 데이터를 사용자 장치에 전송하고, 상기 웹 서버는 사용자에게 어떠한 서비스가 있는지를 알려주는 매개체 역할을 할 수 있다. 사용자가 상기 웹 서버에 원하는 서비스를 요청하면, 상기 웹 서버는 이를 스트리밍 서버에 전달하고, 상기 스트리밍 서버는 사용자에게 멀티미디어 데이터를 전송할 수 있다. 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 수행할 수 있다.
- [473] 상기 스트리밍 서버는 미디어 저장소 및/또는 인코딩 서버로부터 콘텐츠를 수신할 수 있다. 예를 들어, 상기 인코딩 서버로부터 콘텐츠를 수신하는 경우, 상기 콘텐츠를 실시간으로 수신할 수 있다. 이 경우, 원활한 스트리밍 서비스를 제공하기 위하여 상기 스트리밍 서버는 상기 비트스트림을 일정 시간동안 저장할 수 있다.
- [474] 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.

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

[476] 본 개시의 범위는 다양한 실시예의 방법에 따른 동작이 장치 또는 컴퓨터 상에서 실행되도록 하는 소프트웨어 또는 머신-실행가능한 명령들(예를 들어, 운영체제, 애플리케이션, 펌웨어(firmware), 프로그램 등), 및 이러한 소프트웨어 또는 명령 등이 저장되어 장치 또는 컴퓨터 상에서 실행 가능한 비-일시적 컴퓨터-판독가능 매체(non-transitory computer-readable medium)를 포함한다.

산업상 이용가능성

[477] 본 개시에 따른 실시예는 영상을 부호화/복호화하는데 이용될 수 있다.

청구범위

- [청구항 1] 영상 복호화 장치에 의해 수행되는 영상 복호화 방법에 있어서, NNPF(neural-network post-filter) SEI(supplemental enhancement information) 메시지를 획득하되, 상기 NNPF SEI 메시지는 NNPF(neural-network post-filter characteristics) SEI 메시지 및 NNPFA(neural-network post-filter activation) SEI 메시지를 포함하고, 상기 NNPF SEI 메시지에 기반하여 후처리(post-processing) 필터로 이용될 수 있는 신경망을 결정하는 단계; 및 상기 NNPFA SEI 메시지에 기반하여, 상기 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를 결정하는 단계를 포함하고, 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 NNPFA SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어의 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 영상 복호화 방법.
- [청구항 2] 제1항에 있어서, 상기 대상 NNPF의 활성화 여부를 결정하는 단계는, 상기 대상 NNPF를 결정하는 단계; 상기 대상 NNPF의 지속성 취소 여부를 결정하는 단계; 및 상기 대상 NNPF의 지속성을 취소하지 않는 것에 기반하여 상기 대상 NNPF의 지속성을 결정하는 단계를 포함하는, 영상 복호화 방법.
- [청구항 3] 제1항에 있어서, 상기 대상 NNPF는, 상기 현재 픽처와 관련되고 상기 NNPFA SEI 메시지와 동일한 NNPF 식별 정보를 갖는 하나 이상의 NNPF SEI 메시지에 의해 지정되는 대상 NNPF를 나타내는 대상 NNPF 식별 정보에 기반하여 결정되는, 영상 복호화 방법.
- [청구항 4] 제2항에 있어서, 상기 대상 NNPF의 지속성 취소 여부는, 상기 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 이전 NNPFA SEI 메시지에 의해서 설정되는 대상 NNPF의 지속성 취소 여부를 나타내는 지속성 취소 플래그에 기반하여 결정되는, 영상 복호화 방법.
- [청구항 5] 제2항에 있어서, 상기 대상 NNPF의 지속성은, 상기 현재 레이어의 픽처와 관련된 NNPFA SEI 메시지가 가지는 지속성 취소 플래그의 값과 상관없이 상기 현재 레이어의 픽처가 상기 출력 순서상 상기 현재 픽처 다음에 출력되는지에 기반하여 결정되는, 영상 복호화 방법.
- [청구항 6] 영상 부호화 장치에 의해 수행되는 영상 부호화 방법으로서, 후처리(post-processing) 필터로 이용될 수 있는 신경망을 NNPF(neural-network post-filter characteristics) SEI(supplemental

enhancement information) 메시지로 부호화하는 단계; 및
 현재 픽처에 적용될 수 있는 대상 NNPF의 활성화 여부를
 NNPFA(neural-network post-filter activation) SEI 메시지로 부호화하는
 단계를 포함하고,

상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를
 가지는 NNPFA SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상
 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처
 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해
 사용되는, 영상 부호화 방법.

[청구항 7] 제6항에 있어서, 상기 대상 NNPF의 활성화 여부는,
 상기 대상 NNPF를 결정하고, 상기 대상 NNPF의 지속성 취소 여부를
 결정하고, 상기 대상 NNPF의 지속성을 취소하지 않는 것에 기반하여
 상기 대상 NNPF의 지속성을 결정하는 것에 의해 결정되는, 영상 부호화
 방법.

[청구항 8] 제6항에 있어서, 상기 대상 NNPF는,
 상기 현재 픽처와 관련되고 상기 NNPFA SEI 메시지와 동일한 NNPF 식별
 정보를 갖는 하나 이상의 NNPF SEI 메시지에 의해 지정되는 대상
 NNPF를 나타내는 대상 NNPF 식별 정보에 기반하여 결정되는, 영상
 부호화 방법.

[청구항 9] 제7항에 있어서, 상기 대상 NNPF의 지속성 취소 여부는,
 상기 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지는 이전
 NNPFA SEI 메시지에 의해서 설정되는 상기 대상 NNPF의 지속성 취소
 여부를 나타내는 지속성 취소 플래그에 기반하여 결정되는, 영상 부호화
 방법.

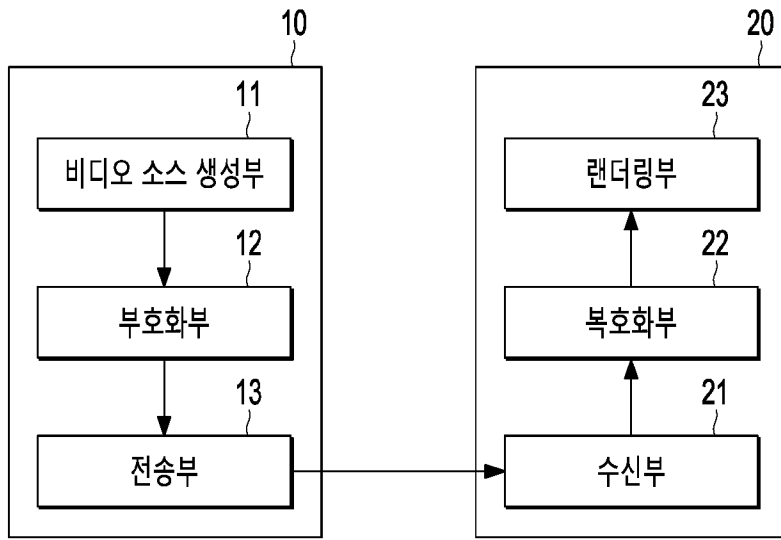
[청구항 10] 제7항에 있어서, 상기 대상 NNPF의 지속성은,
 상기 현재 레이어의 픽처와 관련된 NNPFA SEI 메시지가 가지는 지속성
 취소 플래그의 값과 상관없이 상기 현재 레이어의 픽처가 상기 출력
 순서상 상기 현재 픽처 다음에 출력되는지에 기반하여 결정되는, 영상
 부호화 방법.

[청구항 11] 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 컴퓨터 판독
 가능한 기록 매체로서, 상기 비트스트림은,
 후처리(post-processing) 필터로 이용될 수 있는 신경망을 나타내는 적어도
 하나의 NNPF(neural-network post-filter characteristics) SEI(supplemental
 enhancement information) 메시지 및
 현재 픽처에 적용될 수 있는 대상 신경망 후처리 필터의 활성화 여부를
 나타내는 NNPFA(neural-network post-filter activation) SEI 메시지를
 포함하고,
 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를

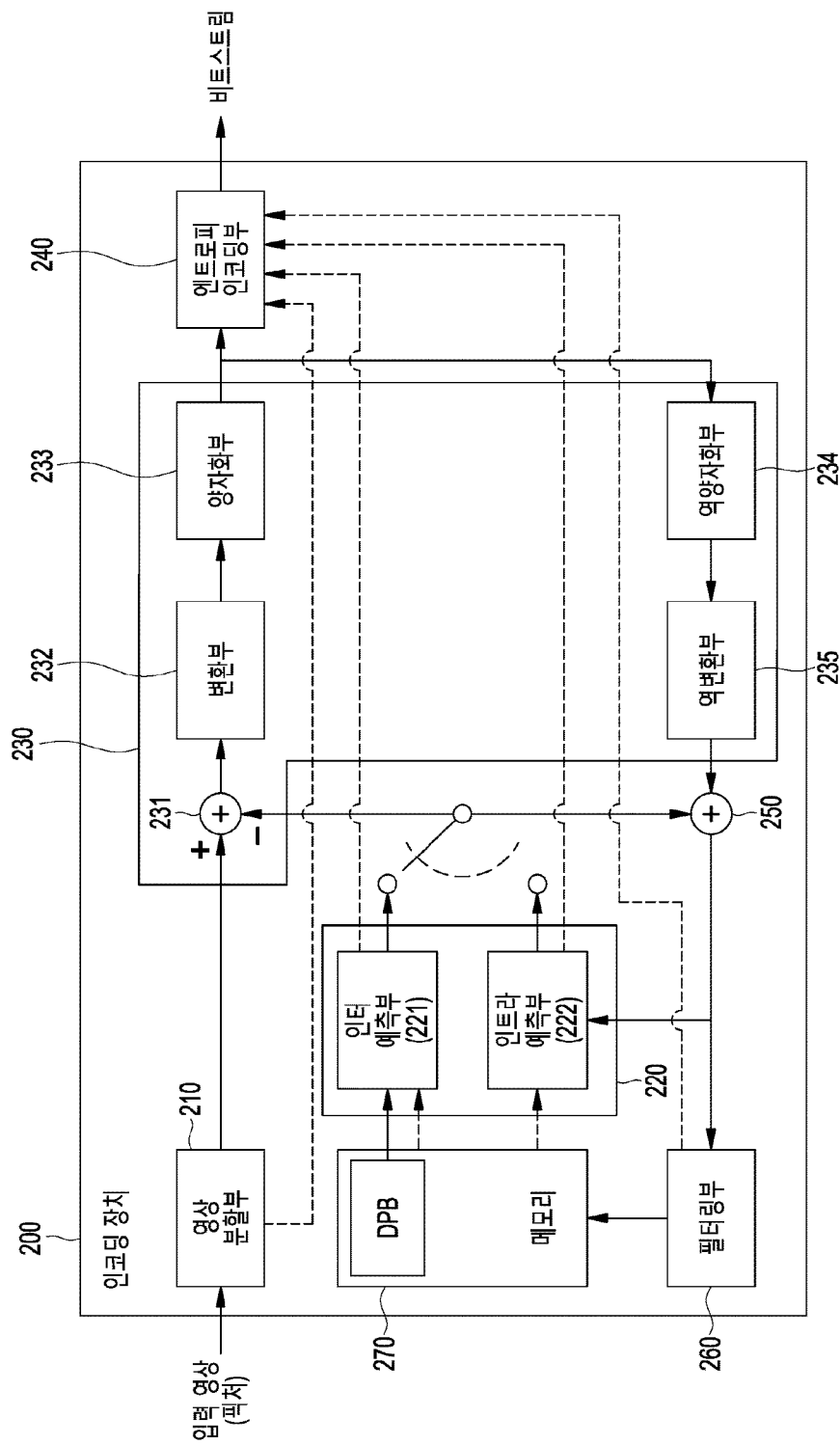
가지는 NNPF SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 컴퓨터 판독 가능한 기록 매체.

- [청구항 12] 영상에 대한 데이터를 전송하는 전송 방법에 있어서, 상기 전송 방법은, 상기 영상에 대한 비트스트림을 생성하되, 상기 비트스트림은, 후처리(post-processing) 필터로 이용될 수 있는 신경망을 NNPFC(neural-network post-filter characteristics) SEI(supplemental enhancement information) 메시지로 부호화하는 단계; 및 현재 픽처에 적용될 수 있는 대상 신경망 후처리 필터의 활성화 여부를 NNPFA(neural-network post-filter activation) SEI 메시지로 부호화하는 단계를 기반으로 생성되고, 상기 비트스트림을 포함하는 상기 데이터를 전송하는 단계를 포함하고, 상기 대상 NNPF은, 현재 SEI 메시지와 동일한 대상 NNPF 식별 정보를 가지며 NNPF SEI 메시지와 관련된 현재 레이어의 픽처가 출력 순서상 상기 현재 픽처 다음에 출력될 때까지, 상기 출력 순서상 상기 현재 픽처 및 상기 현재 레이어 내 후속 픽처들에 대한 후처리 필터링하기 위해 사용되는, 전송 방법.

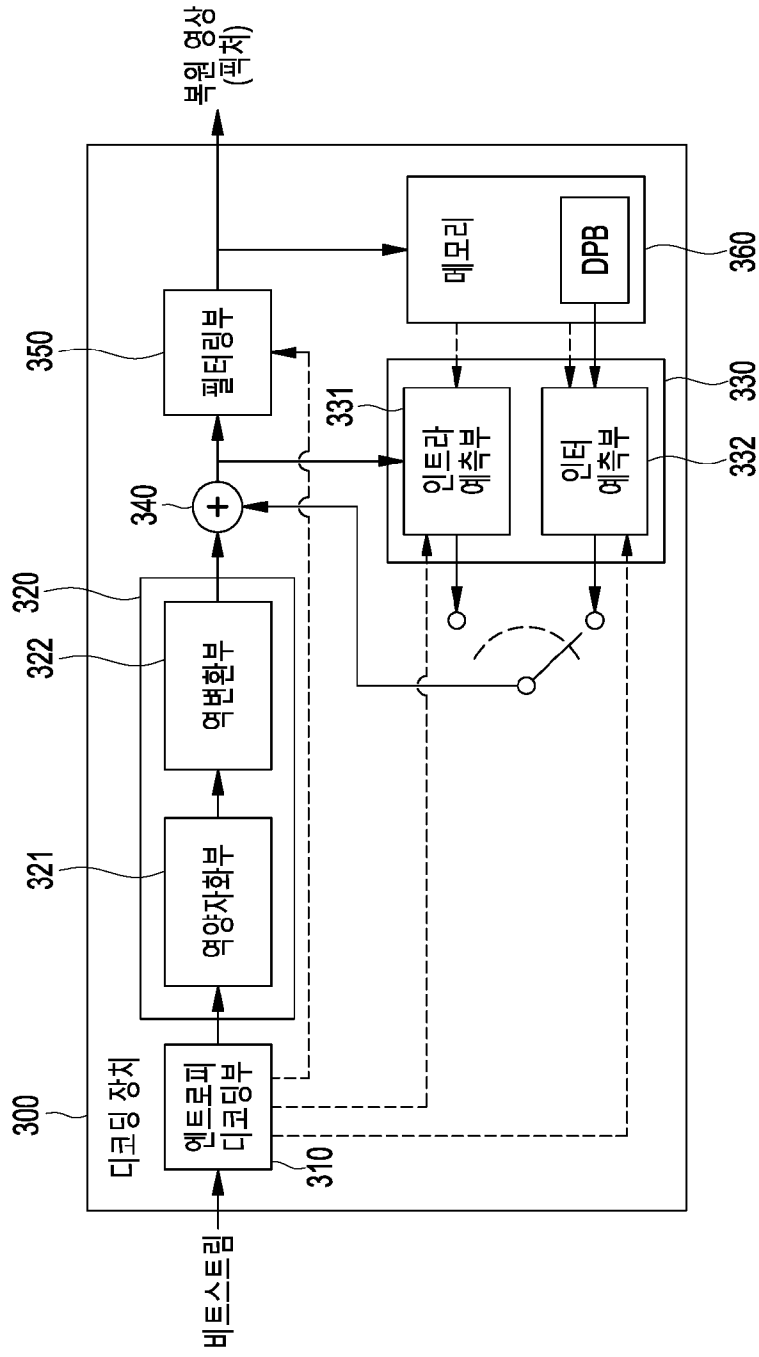
[도 1]



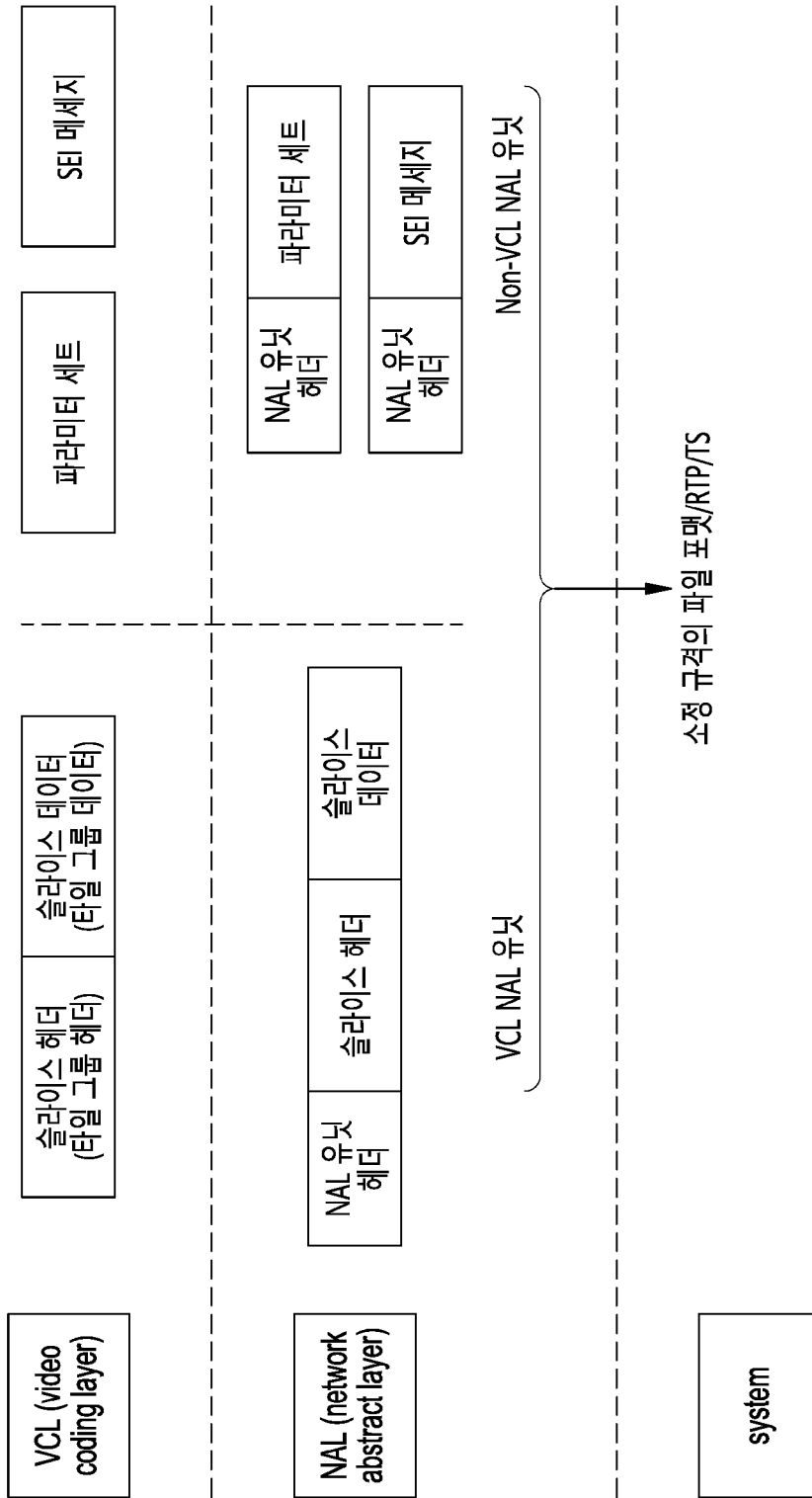
[도2]



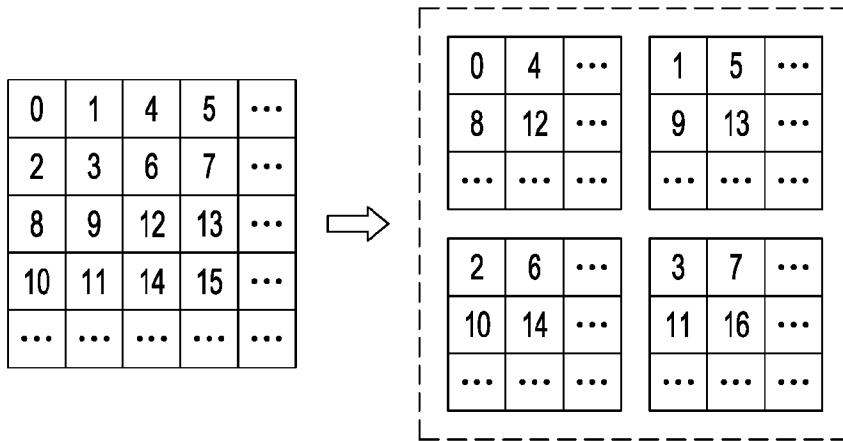
[도3]



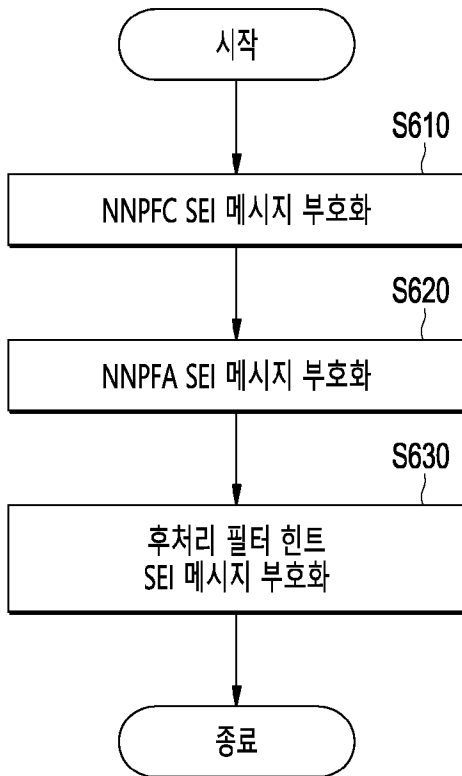
[도4]



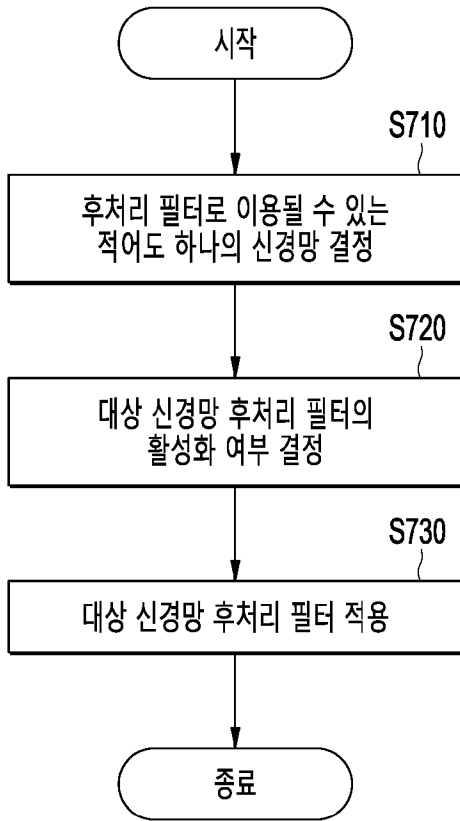
[도5]



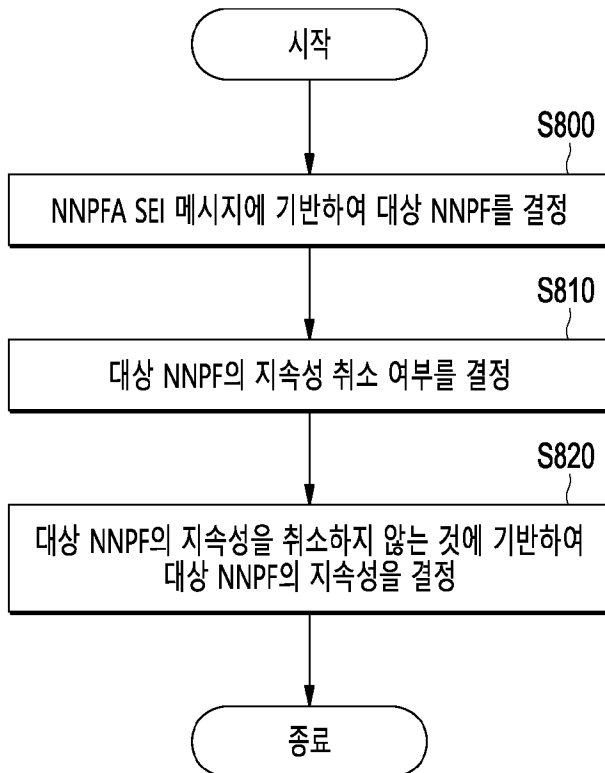
[도6]



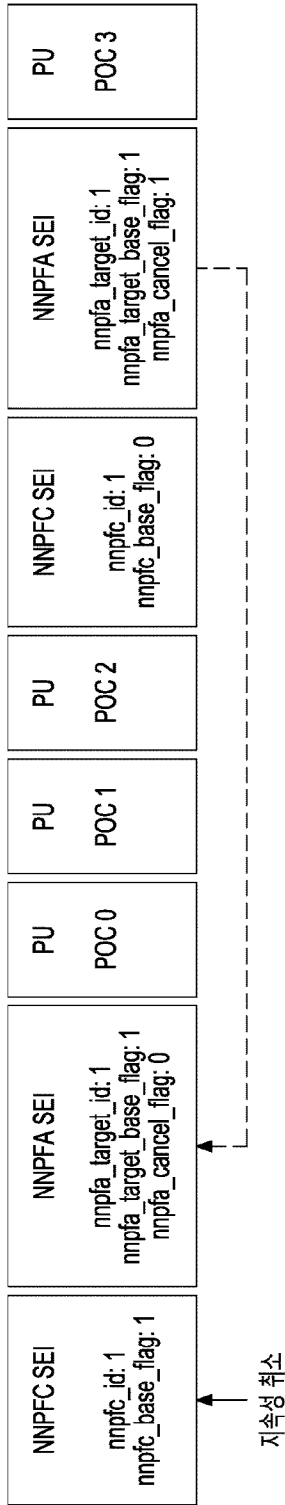
[도7]



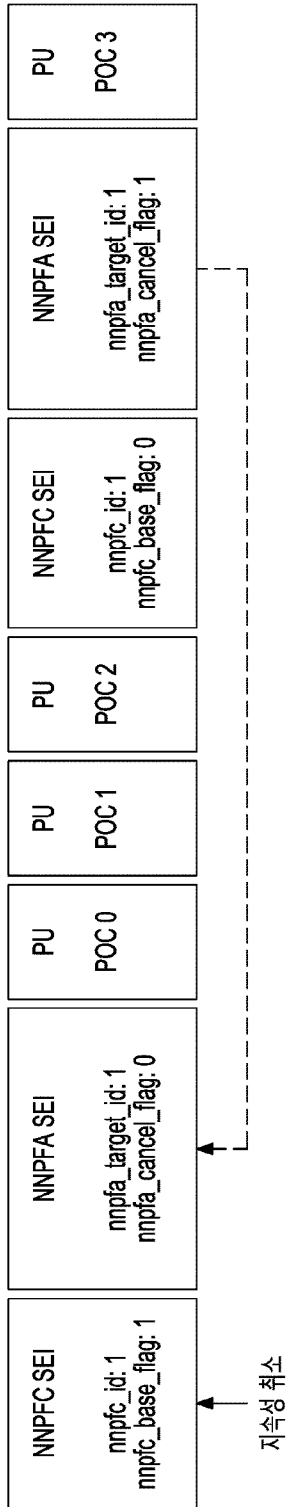
[도8]



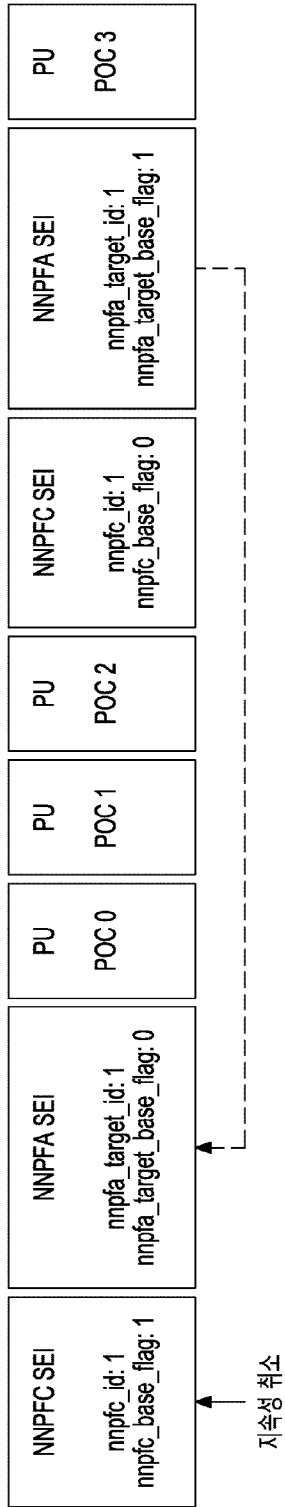
[도9]



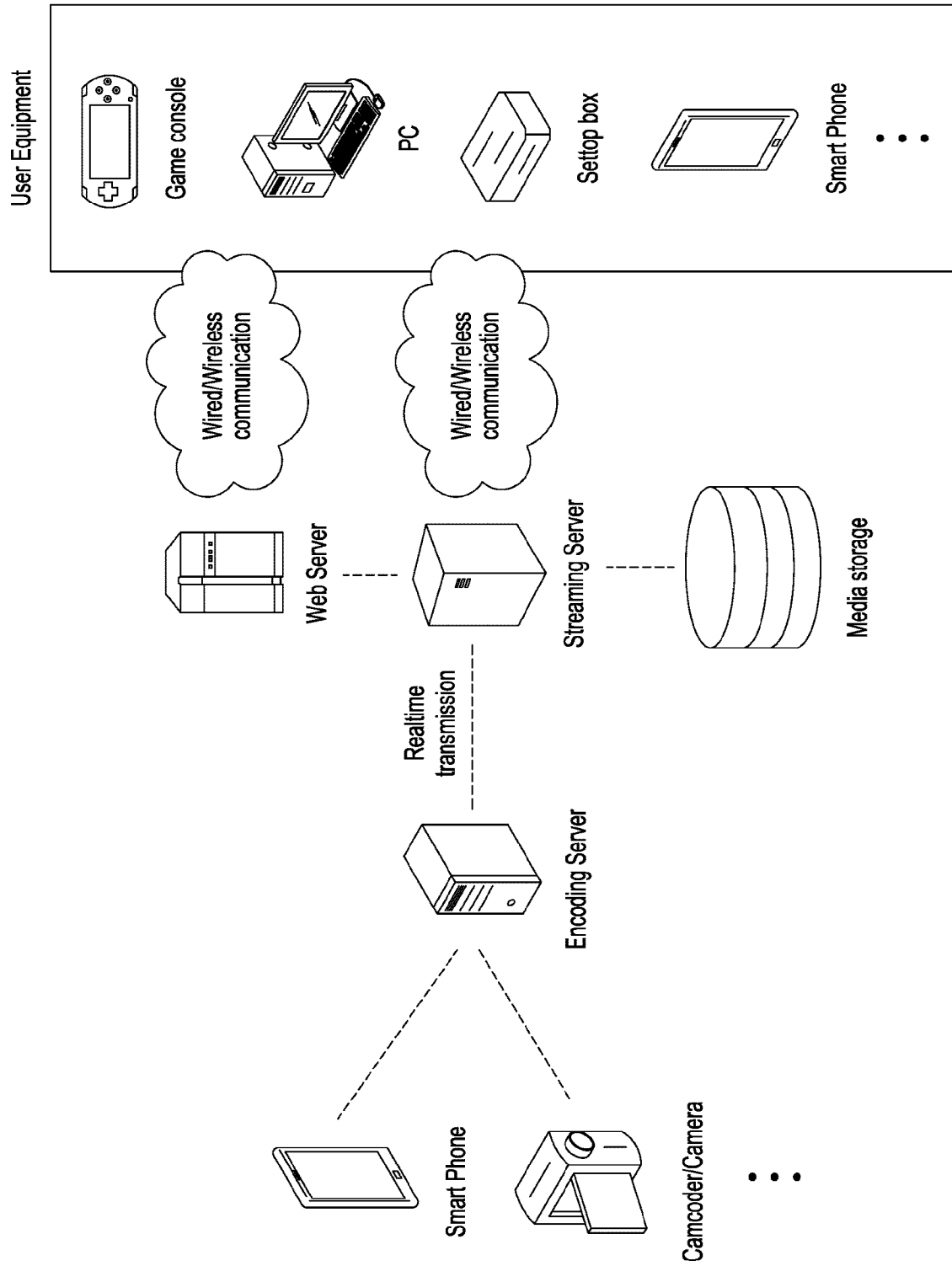
[도 10]



[圖 11]



[도12]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2024/009604

A. CLASSIFICATION OF SUBJECT MATTER		
H04N 19/117(2014.01)i; H04N 19/70(2014.01)i; H04N 19/85(2014.01)i; G06N 3/02(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04N 19/117(2014.01); G06N 3/04(2006.01); G06N 3/0464(2023.01); G06N 3/048(2023.01); H04N 19/119(2014.01); H04N 19/436(2014.01); H04N 19/70(2014.01)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models: IPC as above Japanese utility models and applications for utility models: IPC as above		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS (KIPO internal) & keywords: NNPF(neural-network post-filter), 식별(identify), 활성화(activation), SEI(supplemental enhancement information)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WANG, Ye-Kui et al. AHG9: NNPF activation parameters. JVET-AD0089-v1, Joint Video Experts Team (JVET) of ITU-T SG WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 30th Meeting: Antalya, TR. 21 April 2023. See pages 1-13.	1-12
A	KR 10-2023-0078658 A (QUALCOMM INCORPORATED) 02 June 2023 (2023-06-02) See claims 1-15.	1-12
A	KR 10-2023-0079360 A (QUALCOMM INCORPORATED) 07 June 2023 (2023-06-07) See claims 1-17.	1-12
A	US 2022-0141496 A1 (TENCENT AMERICA LLC) 05 May 2022 (2022-05-05) See paragraphs [0045]-[0050]; and figure 3.	1-12
A	US 2022-0329837 A1 (LEMON INC.) 13 October 2022 (2022-10-13) See paragraphs [0371]-[0372]; and figure 21.	1-12
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 22 October 2024		Date of mailing of the international search report 22 October 2024
Name and mailing address of the ISA/KR Korean Intellectual Property Office Government Complex-Daejeon Building 4, 189 Cheongsaro, Seo-gu, Daejeon 35208 Facsimile No. +82-42-481-8578		Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2024/009604

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
KR	10-2023-0078658	A	02 June 2023	CN	116325729	A	23 June 2023
				EP	4222954	A1	09 August 2023
				JP	2023-543762	A	18 October 2023
				US	11647212	B2	09 May 2023
				US	11778213	B2	03 October 2023
				US	2022-0103845	A1	31 March 2022
				US	2023-0012661	A1	19 January 2023
				WO	2022-072684	A1	07 April 2022

KR	10-2023-0079360	A	07 June 2023	CN	116349226	A	27 June 2023
				EP	4222960	A1	09 August 2023
				JP	2023-542841	A	12 October 2023
				US	11930215	B2	12 March 2024
				US	2022-0103864	A1	31 March 2022
				US	2024-0244265	A1	18 July 2024
				WO	2022-072245	A1	07 April 2022

US	2022-0141496	A1	05 May 2022	CN	111726637	A	29 September 2020
				CN	111726637	B	16 April 2024
				US	11265580	B2	01 March 2022
				US	11765391	B2	19 September 2023
				US	2020-0304836	A1	24 September 2020

US	2022-0329837	A1	13 October 2022	CN	115209143	A	18 October 2022
				US	12113995	B2	08 October 2024

A. 발명이 속하는 기술분류(국제특허분류(IPC)) H04N 19/117(2014.01)i; H04N 19/70(2014.01)i; H04N 19/85(2014.01)i; G06N 3/02(2006.01)i		
B. 조사된 분야 조사된 최소문헌(국제특허분류를 기재) H04N 19/117(2014.01); G06N 3/04(2006.01); G06N 3/0464(2023.01); G06N 3/048(2023.01); H04N 19/119(2014.01); H04N 19/436(2014.01); H04N 19/70(2014.01) 조사된 기술분야에 속하는 최소문헌 이외의 문헌 한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC 일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC 국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우)) eKOMPASS(특허청 내부 검색시스템) & 키워드: NNPF(neural-network post-filter), 식별(identify), 활성화(activation), SEI(supplemental enhancement information)		
C. 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X	YE-KUI WANG 등, "AHG9: NNPF activation parameters", JVET-AD0089-v1, Joint Video Experts Team (JVET) of ITU-T SG WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 30th Meeting: Antalya, TR, 2023.04.21 페이지 1-13	1-12
A	KR 10-2023-0078658 A (헬컴 인코포레이티드) 2023.06.02 청구항 1-15	1-12
A	KR 10-2023-0079360 A (헬컴 인코포레이티드) 2023.06.07 청구항 1-17	1-12
A	US 2022-0141496 A1 (TENCENT AMERICA LLC) 2022.05.05 단락 [0045]-[0050]; 및 도면 3	1-12
A	US 2022-0329837 A1 (LEMON INC.) 2022.10.13 단락 [0371]-[0372]; 및 도면 21	1-12
<input type="checkbox"/> 추가 문헌이 C(계속)에 기재되어 있습니다. <input checked="" type="checkbox"/> 대응특허에 관한 별지를 참조하십시오.		
* 인용된 문헌의 특별 카테고리: "A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌 "D" 본 국제출원에서 출원인이 인용한 문헌 "E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌 "L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌 "O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌 "P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌 "T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌 "X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다. "Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다. "&" 동일한 대응특허문헌에 속하는 문헌		
국제조사의 실제 완료일 2024년 10월 22일 (22.10.2024)		국제조사보고서 발송일 2024년 10월 22일 (22.10.2024)
ISA/KR의 명칭 및 우편주소 대한민국 특허청 (35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사) 팩스 번호 +82-42-481-8578		심사관 양정록 전화번호 +82-42-481-5709

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2023-0078658 A	2023/06/02	CN 116325729 A	2023/06/23
		EP 4222954 A1	2023/08/09
		JP 2023-543762 A	2023/10/18
		US 11647212 B2	2023/05/09
		US 11778213 B2	2023/10/03
		US 2022-0103845 A1	2022/03/31
		US 2023-0012661 A1	2023/01/19
		WO 2022-072684 A1	2022/04/07
KR 10-2023-0079360 A	2023/06/07	CN 116349226 A	2023/06/27
		EP 4222960 A1	2023/08/09
		JP 2023-542841 A	2023/10/12
		US 11930215 B2	2024/03/12
		US 2022-0103864 A1	2022/03/31
		US 2024-0244265 A1	2024/07/18
US 2022-0141496 A1	2022/05/05	WO 2022-072245 A1	2022/04/07
		CN 111726637 A	2020/09/29
		CN 111726637 B	2024/04/16
		US 11265580 B2	2022/03/01
		US 11765391 B2	2023/09/19
US 2022-0329837 A1	2022/10/13	US 2020-0304836 A1	2020/09/24
		CN 115209143 A	2022/10/18
		US 12113995 B2	2024/10/08