

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일

2025년 1월 2일 (02.01.2025)



(10) 국제공개번호

WO 2025/005681 A1

(51) 국제특허분류:

H04N 19/85 (2014.01)

H04N 19/117 (2014.01)

H04N 19/70 (2014.01)

H04N 19/172 (2014.01)

G06N 3/02 (2006.01)

(21) 국제출원번호:

PCT/KR2024/008958

(22) 국제출원일:

2024년 6월 27일 (27.06.2024)

(25) 출원언어:

한국어

(26) 공개언어:

한국어

(30) 우선권정보:

63/523,373

2023년 6월 27일 (27.06.2023) US

(71) 출원인: 엘지전자 주식회사 (LG ELECTRONICS INC.) [KR/KR]; 07336 서울특별시 영등포구 여의대로 128, Seoul (KR).

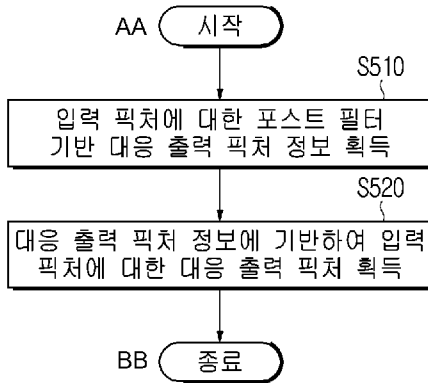
(72) 발명자: 탄헨드리 (TAN, Hendry); 06772 서울특별시 서초구 양재대로11길 19 LG전자 특허센터, Seoul (KR).

(74) 대리인: 성병기 (SUNG, Byung Kee); 06651 서울특별시 서초구 사임당로 32 12층 마루특허법률사무소, 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.

(54) Title: IMAGE ENCODING/DECODING METHOD, METHOD FOR TRANSMITTING BITSTREAM, AND RECORDING MEDIUM HAVING STORED BITSTREAM THEREIN

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



S510 ... Acquire post-filter-based corresponding output picture information for input picture

S520 ... Acquire corresponding output picture for input picture on basis of corresponding output picture information

AA ... Start

BB ... End

(57) Abstract: Provided are an image encoding/decoding method, a bitstream transmission method, and a computer-readable recording medium storing a bitstream. The image decoding method according to the present disclosure comprises the steps of: acquiring post-filter-based output picture information for an input picture from a neural-network post-filter (NNPF)-related supplemental enhancement information (SEI) message; and on the basis of the output picture information, acquiring an output picture for the input picture, wherein the output picture information may include output picture output information which is information indicating whether or not the output picture for the input picture is output.

[다음 쪽 계속]

(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, 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: 영상 부호화/복호화 방법, 비트스트림 전송 방법 및 비트스트림을 저장하는 컴퓨터 판독 가능한 기록 매체가 제공된다. 본 개시에 따른 영상 복호화 방법은, 입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 NNPF (neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지로부터 획득하는 단계 및 상기 출력 픽처 정보에 기반하여, 상기 입력 픽처에 대한 출력 픽처를 획득하는 단계를 포함하되, 상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함할 수 있다.

명세서

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

기술분야

- [1] 본 개시는 영상 부호화/복호화 방법, 비트스트림을 전송하는 방법 및 비트스트림을 저장한 기록 매체에 관한 것으로서, 신경망 후처리 필터(neural-network post-filter)와 관련된 영상 부호화/복호화 방법, 비트스트림을 전송하는 방법 및 비트스트림을 저장한 기록 매체에 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 NNPF 관련 SEI 메시지(NNPF SEI 및 NNPF SEI)를 처리하는 방법을 제공하는 것을 목적으로 한다.
- [6] 또한, 본 개시는 NNPF 관련 SEI 메시지에 의해 NNPF의 출력 픽처를 보다 명확히 특정하는 것을 목적으로 한다.
- [7] 또한, 본 개시는 NNPF의 출력 픽처와 관련된 정보의 의미를 명확히 하는 것을 목적으로 한다.
- [8] 또한, 본 개시는 NNPF의 출력 픽처와 관련된 정보의 의미를 명확히 함으로써 디코더의 오차를 줄이는 것을 목적으로 한다.
- [9] 또한, 본 개시는 NNPF가 출력 픽처와 관련된 정보의 의미를 명확히 함으로써 코딩 품질 및 효율을 향상시키는 것을 목적으로 한다.
- [10] 또한, 본 개시는 입력 픽처가 존재하지 않는 경우의 출력 픽처의 출력 여부를 결정하여 코딩 효율을 향상시키는 것을 목적으로 한다.
- [11] 또한, 본 개시는 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체를 제공하는 것을 목적으로 한다.

- [12] 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.
- [13] 또한, 본 개시는 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법을 제공하는 것을 목적으로 한다.
- [14] 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

- [15] 본 개시의 일 실시예에 따른 영상 복호화 장치에 의해 수행되는 영상 복호화 방법은 입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 NNPF (neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지에서 획득하는 단계 및 상기 출력 픽처 정보에 기반하여, 상기 입력 픽처에 대한 출력 픽처를 획득하는 단계 포함하되, 상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함할 수 있다.
- [16] 한편, 일 실시예로서, 상기 출력 픽처 출력 정보의 값은 상기 입력 픽처의 존재 여부에 기반하여 결정될 수 있다.
- [17] 한편, 일 실시예로서, 상기 출력 픽처 출력 정보의 값은 상기 입력 픽처에 대한 상기 출력 픽처의 존재 여부를 나타내는 정보인 출력 픽처 존재 정보에 더 기반하여 결정될 수 있다.
- [18] 한편, 일 실시예로서, 상기 입력 픽처가 존재하지 않고 상기 출력 픽처 존재 정보가 상기 출력 픽처가 존재함을 나타냄에 기반하여 상기 출력 픽처 출력 정보는 상기 출력 픽처가 출력되지 않음을 나타낼 수 있다.
- [19] 한편, 일 실시예로서, 상기 출력 픽처 출력 정보의 개수는 특정 범위 내의 값으로 결정될 수 있다.
- [20] 한편, 일 실시예로서, 상기 특정 범위는 특정 입력 픽처의 인덱스 값에 기반하여 결정될 수 있다.
- [21] 한편, 일 실시예로서, 상기 출력 픽처 출력 정보의 개수는 특정 값보다 항상 큰 값으로 결정될 수 있다.
- [22] 한편, 일 실시예로서, 상기 NNPF에 기반한 필터링 프로세스에서 상기 존재하지 않는 입력 픽처에 대한 상기 출력 픽처의 출력이 제외될 수 있다.
- [23] 한편, 일 실시예로서, 상기 출력 픽처 존재 정보는 NNPF SEI 메시지에서 획득될 수 있다.
- [24] 한편, 일 실시예로서, 상기 출력 픽처 출력 정보는 NNPF SEI 메시지에서 획득될 수 있다.

- [25] 본 개시의 일 실시예에 따른 영상 부호화 장치에 의해 수행되는 영상 부호화 방법은 입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 결정하는 단계 및 상기 출력 픽처 정보를 NNPF(neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지로 시그널링하는 단계를 포함하되, 상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함할 수 있다.
- [26] 또한, 본 개시에 따르면, 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [27] 또한, 본 개시에 따르면, 본 개시에 따른 영상 복호화 장치에 의해 수신되고 복호화되어 영상의 복원에 이용되는 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [28] 또한, 본 개시에 따르면, 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법이 제공될 수 있다.
- [29] 본 개시에 대하여 위에서 간략하게 요약된 특징들은 후술하는 본 개시의 상세한 설명의 예시적인 양상일 뿐이며, 본 개시의 범위를 제한하는 것은 아니다.
- 발명의 효과**
- [30] 본 개시에 따르면, 부호화/복호화 효율이 향상된 영상 부호화/복호화 방법 및 장치가 제공될 수 있다.
- [31] 또한, 본 개시에 따르면, NNPF 관련 SEI 메시지 내의 정보의 시맨틱스를 수정하여 보다 명확한 의미 전달이 가능해질 수 있다.
- [32] 또한, 본 개시에 따르면, NNPF 관련 SEI 메시지 내의 정보의 시맨틱스를 수정하여 디코더의 오차를 줄일 수 있다.
- [33] 또한, 본 개시에 따르면, NNPF 관련 SEI 메시지로 출력 픽처의 출력을 명확히 함으로써 효율을 향상시킬 수 있다.
- [34] 또한, 본 개시에 따르면, NNPF 관련 SEI 메시지에 의해 NNPF의 출력 픽처에 대한 정보를 보다 명확히 특정함으로써 효율을 향상시킬 수 있다.
- [35] 또한, 본 개시에 따르면, NNPF가 출력 픽처와 관련된 정보의 의미를 명확히 함으로써 코딩 품질 및 효율을 향상시킬 수 있다.
- [36] 또한, 본 개시에 따르면, 본 개시에 따른 영상 부호화 방법에 의해 생성된 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [37] 또한, 본 개시에 따르면, 본 개시에 따른 영상 복호화 장치에 의해 수신되고 복호화되어 영상의 복원에 이용되는 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능한 기록 매체가 제공될 수 있다.
- [38] 또한, 본 개시에 따르면, 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법이 제공될 수 있다.

- [39] 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

- [40] 도 1은 본 개시에 따른 실시예가 적용될 수 있는 비디오 코딩 시스템을 개략적으로 나타내는 도면이다.
- [41] 도 2는 본 개시에 따른 실시예가 적용될 수 있는 영상 부호화 장치를 개략적으로 나타내는 도면이다.
- [42] 도 3은 본 개시에 따른 실시예가 적용될 수 있는 영상 복호화 장치를 개략적으로 나타내는 도면이다.
- [43] 도 4는 루마 채널의 유도를 위한 인터리브(interleaved) 방식을 설명하기 위한 도면이다.
- [44] 도 5는 본 개시에 따른 실시예가 적용될 수 있는 영상 복호화 방법을 설명하기 위한 순서도이다.
- [45] 도 6은 본 개시에 따른 실시예가 적용될 수 있는 영상 부호화 방법을 설명하기 위한 순서도이다.
- [46] 도 7은 본 개시에 따른 실시예가 적용될 수 있는 콘텐츠 스트리밍 시스템을 예시적으로 나타내는 도면이다.

발명의 실시를 위한 형태

- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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, 마찬

가지로 일 실시예에서의 제2 구성요소를 다른 실시예에서 제1 구성요소라고 칭할 수도 있다.

- [51] 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.
- [52] 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.
- [53] 본 개시는 영상의 부호화 및 복호화에 관한 것으로서, 본 개시에서 사용되는 용어는, 본 개시에서 새롭게 정의되지 않는 한 본 개시가 속한 기술 분야에서 통용되는 통상의 의미를 가질 수 있다.
- [54] 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.
- [55] 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.
- [56] 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. 유닛은 경우에 따라서 "샘플 어레이", "블록(block)" 또는 "영역(area)" 등의 용어와 혼용하여 사용될 수 있다. 일반적인 경우, $M \times N$ 블록은 M개의 열과 N개의 행으로 이루어진 샘플들(또는 샘플 어레이) 또는 변환 계수(transform coefficient)들의 집합(또는 어레이)을 포함할 수 있다.
- [57] 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.

- [58] 본 개시에서 "현재 블록"은 크로마 블록이라는 명시적인 기재가 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 기재를 포함하여 표현될 수 있다. 또한, 현재 블록의 크로마 성분 블록은 명시적으로 "크로마 블록" 또는 "현재 크로마 블록"과 같이 크로마 성분 블록이라는 명시적인 기재를 포함하여 표현될 수 있다.
- [59] 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.
- [60] 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.
- [61] 비디오 코딩 시스템 개요
- [62] 도 1은 본 개시에 따른 실시예가 적용될 수 있는 비디오 코딩 시스템을 개략적으로 나타내는 도면이다.
- [63] 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.
- [64] 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.
- [65] 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.

- [66] 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.
- [67] 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.
- [68] 복호화부(22)는 부호화부(12)의 동작에 대응하는 역양자화, 역변환, 예측 등 일련의 절차를 수행하여 비디오/영상을 복호화할 수 있다.
- [69] 렌더링부(23)는 복호화된 비디오/영상을 렌더링할 수 있다. 렌더링된 비디오/영상은 디스플레이부를 통하여 디스플레이될 수 있다.
- [70] 영상 부호화 장치 개요
- [71] 도 2는 본 개시에 따른 실시예가 적용될 수 있는 영상 부호화 장치를 개략적으로 도시한 도면이다.
- [72] 도 2에 도시된 바와 같이, 영상 부호화 장치(100)는 영상 분할부(110), 감산부(115), 변환부(120), 양자화부(130), 역양자화부(140), 역변환부(150), 가산부(155), 필터링부(160), 메모리(170), 인터 예측부(180), 인트라 예측부(185) 및 엔트로피 인코딩부(190)를 포함할 수 있다. 인터 예측부(180) 및 인트라 예측부(185)는 합쳐서 "예측부"라고 지칭될 수 있다. 변환부(120), 양자화부(130), 역양자화부(140), 역변환부(150)는 레지듀얼(residual) 처리부에 포함될 수 있다. 레지듀얼 처리부는 감산부(115)를 더 포함할 수도 있다.
- [73] 영상 부호화 장치(100)를 구성하는 복수의 구성부들의 전부 또는 적어도 일부는 실시예에 따라 하나의 하드웨어 컴포넌트(예를 들어, 인코더 또는 프로세서)로 구현될 수 있다. 또한 메모리(170)는 DPB(decoded picture buffer)를 포함할 수 있고, 디지털 저장 매체에 의하여 구현될 수 있다.
- [74] 영상 분할부(110)는 영상 부호화 장치(100)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) 맵스의 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. 최대 코딩 유닛이 바로 최종 코딩 유닛으로 사용될 수 있고, 최대 코딩 유닛을 분할하여 획득한 하위 맵스의 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.

[75] 예측부(인터 예측부(180) 또는 인트라 예측부(185))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. 예측부는 현재 블록 또는 CU 단위로 인트라 예측이 적용되는지 또는 인터 예측이 적용되는지 결정할 수 있다. 예측부는 현재 블록의 예측에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(190)로 전달할 수 있다. 예측에 관한 정보는 엔트로피 인코딩부(190)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.

[76] 인트라 예측부(185)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(185)는 주변 블록에 적용된 예측 모드를 이용하여, 현재 블록에 적용되는 예측 모드를 결정할 수도 있다.

[77] 인터 예측부(180)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, 서브 블록 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(180)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(180)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.

- [78] 예측부는 후술하는 다양한 예측 방법 및/또는 예측 기법을 기반으로 예측 신호를 생성할 수 있다. 예를 들어, 예측부는 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, 예측부는 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 기복원된 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.

- [79] 예측부를 통해 생성된 예측 신호는 복원 신호를 생성하기 위해 이용되거나 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 감산부(115)는 입력 영상 신호(원본 블록, 원본 샘플 어레이)로부터 예측부에서 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(120)로 전송될 수 있다.

- [80] 변환부(120)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.
- [81] 양자화부(130)는 변환 계수들을 양자화하여 엔트로피 인코딩부(190)로 전송할 수 있다. 엔트로피 인코딩부(190)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(130)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.
- [82] 엔트로피 인코딩부(190)는 예를 들어 지수 골롬(exponential Golomb), CAVLC(context-adaptive variable length coding), CABAC(context-adaptive binary arithmetic coding) 등과 같은 다양한 인코딩 방법을 수행할 수 있다. 엔트로피 인코딩부(190)는 양자화된 변환 계수들 외 비디오/이미지 복원에 필요한 정보들(예를 들어 신텍스 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 시그널링 정보, 전송되는 정보 및/또는 신텍스 요소들은 상술한 인코딩 절차를 통하여 인코딩되어 상기 비트스트림에 포함될 수 있다.
- [83] 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(190)로부터 출력된 신호를 전송하는 전송부(미도시) 및/또는 저장하는 저장부(미도시)가 영상 부호화 장치(100)의 내/

외부 엘리먼트로서 구비될 수 있고, 또는 전송부는 엔트로피 인코딩부(190)의 구성요소로서 구비될 수도 있다.

- [84] 양자화부(130)로부터 출력된 양자화된 변환 계수들은 레지듀얼 신호를 생성하기 위해 이용될 수 있다. 예를 들어, 양자화된 변환 계수들에 역양자화부(140) 및 역변환부(150)를 통해 역양자화 및 역변환을 적용함으로써 레지듀얼 신호(레지듀얼 블록 or 레지듀얼 샘플들)를 복원할 수 있다.
- [85] 가산부(155)는 복원된 레지듀얼 신호를 인터 예측부(180) 또는 인트라 예측부(185)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(155)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.
- [86] 필터링부(160)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(160)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(170), 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(160)는 각 필터링 방법에 대한 설명에서 후술하는 바와 같이 필터링에 관한 다양한 정보를 생성하여 엔트로피 인코딩부(190)로 전달할 수 있다. 필터링에 관한 정보는 엔트로피 인코딩부(190)에서 인코딩되어 비트스트림 형태로 출력될 수 있다.
- [87] 메모리(170)에 전송된 수정된 복원 픽처는 인터 예측부(180)에서 참조 픽처로 사용될 수 있다. 영상 부호화 장치(100)는 이를 통하여 인터 예측이 적용되는 경우, 영상 부호화 장치(100)와 영상 복호화 장치에서의 예측 미스매치를 피할 수 있고, 부호화 효율도 향상시킬 수 있다.
- [88] 메모리(170) 내 DPB는 인터 예측부(180)에서의 참조 픽처로 사용하기 위해 수정된 복원 픽처를 저장할 수 있다. 메모리(170)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(180)에 전달될 수 있다. 메모리(170)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(185)에 전달할 수 있다.
- [89] 영상 복호화 장치 개요
- [90] 도 3은 본 개시에 따른 실시예가 적용될 수 있는 영상 복호화 장치를 개략적으로 도시한 도면이다.

- [91] 도 3에 도시된 바와 같이, 영상 복호화 장치(200)는 엔트로피 디코딩부(210), 역양자화부(220), 역변환부(230), 가산부(235), 필터링부(240), 메모리(250), 인터 예측부(260) 및 인트라 예측부(265)를 포함하여 구성될 수 있다. 인터 예측부(260) 및 인트라 예측부(265)를 합쳐서 "예측부"라고 지칭될 수 있다. 역양자화부(220), 역변환부(230)는 레지듀얼 처리부에 포함될 수 있다.
- [92] 영상 복호화 장치(200)를 구성하는 복수의 구성부들의 전부 또는 적어도 일부는 실시예에 따라 하나의 하드웨어 컴포넌트(예를 들어 디코더 또는 프로세서)로 구현될 수 있다. 또한 메모리(170)는 DPB를 포함할 수 있고, 디지털 저장 매체에 의하여 구현될 수 있다.
- [93] 비디오/영상 정보를 포함하는 비트스트림을 수신한 영상 복호화 장치(200)는 도 2의 영상 부호화 장치(100)에서 수행된 프로세스에 대응하는 프로세스를 수행하여 영상을 복원할 수 있다. 예를 들어, 영상 복호화 장치(200)는 영상 부호화 장치에서 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 분할하여 획득될 수 있다. 그리고, 영상 복호화 장치(200)를 통해 디코딩 및 출력된 복원 영상 신호는 재생 장치(미도시)를 통해 재생될 수 있다.
- [94] 영상 복호화 장치(200)는 도 2의 영상 부호화 장치로부터 출력된 신호를 비트스트림 형태로 수신할 수 있다. 수신된 신호는 엔트로피 디코딩부(210)를 통해 디코딩될 수 있다. 예를 들어, 엔트로피 디코딩부(210)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 장치는 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(210)는 지수 골롬 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 다

음 심볼/빈의 문맥 모델을 위해 디코딩된 심볼/빈의 정보를 이용하여 문맥 모델을 업데이트할 수 있다. 엔트로피 디코딩부(210)에서 디코딩된 정보 중 예측에 관한 정보는 예측부(인터 예측부(260) 및 인트라 예측부(265))로 제공되고, 엔트로피 디코딩부(210)에서 엔트로피 디코딩이 수행된 레지듀얼 값, 즉 양자화된 변환 계수들 및 관련 파라미터 정보는 역양자화부(220)로 입력될 수 있다. 또한, 엔트로피 디코딩부(210)에서 디코딩된 정보 중 필터링에 관한 정보는 필터링부(240)로 제공될 수 있다. 한편, 영상 부호화 장치로부터 출력된 신호를 수신하는 수신부(미도시)가 영상 복호화 장치(200)의 내/외부 엘리먼트로서 추가적으로 구비될 수 있고, 또는 수신부는 엔트로피 디코딩부(210)의 구성요소로서 구비될 수도 있다.

- [95] 한편, 본 개시에 따른 영상 복호화 장치는 비디오/영상/픽처 복호화 장치라고 불릴 수 있다. 상기 영상 복호화 장치는 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(210)를 포함할 수 있고, 상기 샘플 디코더는 역양자화부(220), 역변환부(230), 가산부(235), 필터링부(240), 메모리(250), 인터 예측부(260) 및 인트라 예측부(265) 중 적어도 하나를 포함할 수 있다.
- [96] 역양자화부(220)에서는 양자화된 변환 계수들을 역양자화하여 변환 계수들을 출력할 수 있다. 역양자화부(220)는 양자화된 변환 계수들을 2차원의 블록 형태로 재정렬할 수 있다. 이 경우 상기 재정렬은 영상 부호화 장치에서 수행된 계수 스캔 순서에 기반하여 수행될 수 있다. 역양자화부(220)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.
- [97] 역변환부(230)에서는 변환 계수들을 역변환하여 레지듀얼 신호(레지듀얼 블록, 레지듀얼 샘플 어레이)를 획득할 수 있다.
- [98] 예측부는 현재 블록에 대한 예측을 수행하고, 상기 현재 블록에 대한 예측 샘플들을 포함하는 예측된 블록(predicted block)을 생성할 수 있다. 예측부는 엔트로피 디코딩부(210)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.
- [99] 예측부가 후술하는 다양한 예측 방법(기법)을 기반으로 예측 신호를 생성할 수 있음은 영상 부호화 장치(100)의 예측부에 대한 설명에서 언급된 바와 동일하다.
- [100] 인트라 예측부(265)는 현재 픽처 내의 샘플들을 참조하여 현재 블록을 예측할 수 있다. 인트라 예측부(185)에 대한 설명은 인트라 예측부(265)에 대해서도 동일하게 적용될 수 있다.
- [101] 인터 예측부(260)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, 서브

블록 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(260)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.

[102] 가산부(235)는 획득된 레지듀얼 신호를 예측부(인터 예측부(260) 및/또는 인트라 예측부(265) 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(155)에 대한 설명은 가산부(235)에 대해서도 동일하게 적용될 수 있다. 가산부(235)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.

[103] 필터링부(240)는 복원 신호에 필터링을 적용하여 주관적/객관적 화질을 향상시킬 수 있다. 예를 들어 필터링부(240)는 복원 픽처에 다양한 필터링 방법을 적용하여 수정된(modified) 복원 픽처를 생성할 수 있고, 상기 수정된 복원 픽처를 메모리(250), 구체적으로 메모리(250)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.

[104] 메모리(250)의 DPB에 저장된 (수정된) 복원 픽처는 인터 예측부(260)에서 참조 픽처로 사용될 수 있다. 메모리(250)는 현재 픽처 내 움직임 정보가 도출된(또는 디코딩된) 블록의 움직임 정보 및/또는 이미 복원된 픽처 내 블록들의 움직임 정보를 저장할 수 있다. 상기 저장된 움직임 정보는 공간적 주변 블록의 움직임 정보 또는 시간적 주변 블록의 움직임 정보로 활용하기 위하여 인터 예측부(260)에 전달할 수 있다. 메모리(250)는 현재 픽처 내 복원된 블록들의 복원 샘플들을 저장할 수 있고, 인트라 예측부(265)에 전달할 수 있다.

[105] 본 명세서에서, 영상 부호화 장치(100)의 필터링부(160), 인터 예측부(180) 및 인트라 예측부(185)에서 설명된 실시예들은 각각 영상 복호화 장치(200)의 필터링부(240), 인터 예측부(260) 및 인트라 예측부(265)에도 동일 또는 대응되도록 적용될 수 있다.

[106] 신경망 포스트 필터 특성(Neural-network post-filter characteristics, NNPFC)

[107] 표 1 내지 표 3의 결합은 NNPFCS 선택스 구조를 나타낸다.

[108] [표1]

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)
nnpfc_inp_format_idc	ue(v)
nnpfc_auxiliary_inp_idc	ue(v)

[109] [표2]

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)
} else {	
nnpfc_extended_patch_width_cd_delta_minus1	ue(v)
nnpfc_extended_patch_height_cd_delta_minus1	ue(v)
}	

[110] [표3]

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)
}	
}	

- [111] 표 1 내지 표 3의 NNPFCS 선택스 구조는 SEI(supplemental enhancement information) 메시지 형태로 시그널링될 수 있다. 표 1 내지 표 3의 NNPFCS 선택스 구조를 시그널링하는 SEI 메시지는 NNPFCS SEI 메시지라고 지칭될 수 있다.
- [112] NNPFCS SEI 메시지는 후처리(post-processing) 필터로 이용될 수 있는 신경망을 특정할 수 있다. 특정 픽처들에 대해 특정된 후처리 필터들의 사용은 신경망 포스트 필터 활성화(neural-network post-filter activation, NNPFSA) SEI 메시지들을 이용하여 나타낼 수 있다. 여기서, '후처리 필터'와 '포스트 필터'는 동일한 의미를 가질 수 있다.
- [113] 이러한 SEI 메시지들을 사용하려면 아래와 같은 변수들의 정의가 필요할 수 있다.

- [114] - 입력 픽처의 너비와 높이는 루마 샘플 단위로 잘릴 수 있으며, 이 너비와 높이는 각각 `CroppedWidth` 및 `CroppedHeight`로 나타낼 수 있다.
- [115] - 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 있으며, 인덱스인 `idx`는 0 내지 `numInputPics-1`의 범위를 가질 수 있다.
- [116] - `BitDepthY`는 입력 픽처들의 루마 샘플 어레이에 대한 비트 맵스를 나타낼 수 있다.
- [117] - `BitDepthC`는 이들이 존재하는 경우에 입력 픽처들의 크로마 샘플 어레이들의 비트 맵스를 나타낼 수 있다.
- [118] - `ChromaFormatIdc`는 크로마 포맷 식별자를 나타낼 수 있다.
- [119] - `nnpfc_auxiliary_inp_idc`의 값이 1인 경우, 입력 픽처들의 필터링 강도 제어 값 어레이인 `StrengthControlVal[idx]`는 0 내지 1 범위의 실수들을 포함해야 하며, 인덱스인 `idx`는 0 내지 `numInputPics-1`의 범위를 가질 수 있다.
- [120] 인덱스 0을 가지는 입력 픽처는 NNPF SEI 메시지에 의해 정의된 NNPF가 NNPF SEI 메시지에 의해 활성화된 픽처에 해당할 수 있다. 인덱스 `i`가 1 내지 `numInputPics - 1` 범위 내에 존재하는 입력 픽처는 출력 순서에서 인덱스 `i-1`을 가지는 입력 픽처에 우선할 수 있다.
- [121] `nnpfc_purpose`는 표 4에 나타낸 NNPF의 목적을 나타낼 수 있다. 표 4에서, 0이 아닌 (`nnpfc_purpose & bitMask`)은 표 4의 NNPF가 `bitMask` 값에 연관된 목적을 가짐을 나타낼 수 있다. `nnpfc_purpose`가 0보다 크고 (`nnpfc_purpose & bitMask`) 이 0과 동일한 경우, `bitMask` 값에 연관된 목적은 NNPF에 적용 가능하지 않을 수 있다. `nnpfc_purpose`가 0과 동일한 경우, NNPF는 애플리케이션에 의한 결정에 따라 사용될 수 있다. `nnpfc_purpose`의 값은 비트스트림에서 0 내지 63의 범위 내에 존재하도록 제한될 수 있다. `nnpfc_purpose`에 대한 64 내지 65 535 범위의 값은 향후 사용을 위해 예약될 수 있다. 디코더들은 64 내지 65 535 범위의 `nnpfc_purpose`를 가진 NNPF SEI 메시지를 무시해야 한다.

[122] [표4]

bitMask	Interpretation
0x01	General visual quality improvement
0x02	Chroma upsampling (from the 4:2:0 chroma format to the 4:2:2 or 4:4:4 chroma format, or from the 4:2:2 chroma format to the 4:4:4 chroma format)
0x04	Resolution resampling (increasing or decreasing the width or height)
0x08	Picture rate upsampling
0x10	Bit depth upsampling (increasing the luma bit depth or the chroma bit depth)
0x20	Colourization

- [123] `nnpfc_purpose`가 나타내는 NNPF의 목적이 크로마 업샘플링인지 여부를 나타내는 변수 `chromaUpsamplingFlag`, `nnpfc_purpose`가 나타내는 NNPF의 목적이 해상

도 리샘플링인지 여부를 나타내는 변수 resolutionResamplingFlag, nnpfc_purpose가 나타내는 NNPF의 목적이 픽처 레이트(rate) 업샘플링인지 여부를 나타내는 변수 pictureRateUpsamplingFlag, nnpfc_purpose가 나타내는 NNPF의 목적이 비트맵 업샘플링인지 여부를 나타내는 변수 bitDepthUpsamplingFlag, 및 nnpfc_purpose가 나타내는 NNPF의 목적이 컬러라이제이션(colourization)인지 여부를 나타내는 변수 colourizationFlag는 아래의 표 5와 같이 유도될 수 있다.

[124] [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
```

[125] ChromaFormatIdc가 3과 동일한 경우, chromaUpsamplingFlag는 0과 동일하도록 제한될 수 있다. ChromaFormatIdc 또는 chromaUpsamplingFlag가 0과 동일하지 않은 경우, colourizationFlag는 0과 동일하도록 제한될 수 있다. pictureRateUpsamplingFlag가 1과 동일하고 인덱스 0을 가지는 입력 픽처가 fp_arrangement_type의 값이 5인 프레임 패킹 배열 SEI 메시지와 연관된 경우, 모든 입력픽처들은 fp_arrangement_type의 값이 5이고 및 fp_current_frame_is_frame0_flag의 값이 동일한 프레임 패킹 배열 SEI 메시지와 연관될 수 있다.

[126] nnpfc_id는 NNPF를 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_id를 가진 NNPF SEI 메시지를 무시해야 한다.

[127] NNPF SEI 메시지가 현재 CLVS 내에서 특정 nnpfc_id 값을 가지는 디코딩 순서 상 첫 번째 NNPF SEI 메시지인 경우에는 다음이 적용될 수 있다.

[128] - 상기 SEI 메시지는 기본(base) NNPF를 나타낼 수 있다.

[129] - 상기 SEI 메시지는, 출력 순서에서, 현재 CLVS가 종료될 때까지 현재 디코딩된 픽처 및 현재 레이어의 모든 후속하는 디코딩된 픽처들과 관련될 수 있다.

[130] NNPF SEI 메시지는 디코딩 순서에서 현재 CLVS 내 이전 NNPF SEI 메시지의 반복일 수 있으며, 후속하는 시퀀스들은 이 SEI 메시지가 현재 CLVS 내에서 동일한 내용을 가진 유일한 NNPF SEI 메시지인 것처럼 적용될 수 있다.

[131] nnpfc_mode_idc의 값 0은 SEI 메시지가 기본 NNPF를 나타내는 비트스트림을 포함하거나 또는 동일한 nnpfc_id 값을 가진 기본 NNPF에 관련된 업데이트를 나타낼 수 있다.

- [132] NNPFC SEI 메시지가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지인 경우, `nnpfc_mode_idc`의 값 1은 `nnpfc_id` 값에 관련된 기본 NNPFC가 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.
- [133] NNPFC SEI 메시지가 현재 CLVS 내에서 특정 `nnpfc_id` 값을 가지는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지가 아니고 첫 번째 NNPFC SEI 메시지의 반복도 아닌 경우, `nnpfc_mode_idc`의 값 1은 동일한 `nnpfc_id` 값을 가진 기본 NNPFC에 관련된 업데이트가 태그 URI `nnpfc_tag_uri`로 식별되는 형식을 사용하여 `nnpfc_uri`로 표시된 URI로 정의됨을 나타낼 수 있다.
- [134] `nnpfc_base_flag`의 값 1은 SEI 메시지가 기본 NNPFC를 나타냄을 나타낼 수 있다. `nnpfc_base_flag`의 값 0은 SEI 메시지가 기본 NNPFC에 관련된 업데이트를 나타냄을 나타낼 수 있다.
- [135] 아래와 같은 제한들이 `nnpfc_base_flag`의 값에 적용될 수 있다.
- [136] - NNPFC SEI 메시지가 디코딩 순서상 현재 CLVS 내에서 특정 `nnpfc_id` 값을 갖는 첫 번째 NNPFC SEI 메시지인 경우, `nnpfc_base_flag`의 값은 1과 같아야 할 수 있다.
- [137] - NNPFC SEI 메시지 `nnpfcB`가 디코딩 순서상 현재 CLVS 내에서 특정 `nnpfc_id` 값을 갖는 첫 번째 NNPFC SEI 메시지가 아니고 `nnpfc_base_flag`의 값이 1과 같은 경우, NNPFC SEI 메시지는 동일한 `nnpfc_id`를 가지는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지 `nnpfcA`의 반복에 해당할 수 있다. 즉, `nnpfcB`의 페이로드 콘텐츠는 `nnpfcA`의 페이로드 콘텐츠와 동일해야 할 수 있다.
- [138] `nnpfc_base_flag`의 값이 0인 경우, 다음의 제한들이 적용될 수 있다.
- [139] - 이 SEI 메시지는 동일한 `nnpfc_id` 값을 사용하여 디코딩 순서 상 선행하는 기본 NNPFC와 관련된 업데이트를 정의할 수 있다. 업데이트는 누적되지 않지만 각 업데이트는 현재 CLVS 내에서 특정 `nnpfc_id` 값을 갖는 디코딩 순서 상 첫 번째 NNPFC SEI 메시지에 지정된 NNPFC인 기본 NNPFC에 적용될 수 있다. 이 SEI 메시지에 의해 정의된 NNPFC는 동일한 `nnpfc_id` 값을 가진 기본 NNPFC에 관련된 SEI 메시지에 의해 정의된 업데이트를 적용함으로써 획득될 수 있다.
- [140] - 이 SEI 메시지는 현재 디코딩된 픽처와 현재 레이어의 출력 순서 상 현재 CLVS의 끝까지의 모든 후속하는 디코딩된 픽처들에 관련될 수 있다. 여기서, 현재 CLVS 내에서 출력 순서 상 현재 디코딩된 픽처를 따르는 디코딩된 픽처는 제외될 수 있다. 이 SEI 메시지는 `nnpfc_base_flag`가 0이고 현재 CLVS 내의 특정 `nnpfc_id` 값 중 더 빠른 값을 갖는 디코딩 순서 상 후속하는 NNPFC SEI 메시지에 연관될 수 있다.
- [141] `nnpfc_mode_idc`의 값 0은 SEI 메시지가 기본 NNPFC를 나타내는 비트스트림을 포함하거나 (`nnpfc_base_flag`의 값이 1인 경우), 동일한 `nnpfc_id` 값을 가지는 기본 NNPFC에 대한 업데이트 (`nnpfc_base_flag`의 값이 0인 경우)임을 나타낼 수 있

다. `nnpfc_base_flag`의 값이 1인 경우, 1과 동일한 `nnpfc_mode_idc`는 `nnpfc_id`의 값에 연관된 기본 NNPF가 URI에 의해 식별되는 신경망에 연관됨을 나타낼 수 있다. 여기서, URI는 태그 URI `nnpfc_tag_uri`에 의해 지시되는 포맷의 `nnpfc_uri`에 의해 지시될 수 있다. `nnpfc_base_flag`의 값이 0인 경우, 1과 동일한 `nnpfc_mode_idc`는 동일한 `nnpfc_id` 값의 기본 NNPF에 연관된 업데이트가 URI에 의해 정의됨을 나타낼 수 있다. 여기서, URI는 태그 URI `nnpfc_tag_uri`에 의해 지시되는 포맷의 `nnpfc_uri`에 의해 지시될 수 있다.

- [142] `nnpfc_mode_idc`의 값은 비트스트림에서 0 내지 1의 범위를 가지도록 제한될 수 있다. `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.
- [143] `nnpfc_reserved_zero_bit_a`는 비트스트림 제한에 의해 0과 동일한 값을 가지도록 제한될 수 있다. 디코더들은 `nnpfc_reserved_zero_bit_a`의 값이 0이 아닌 NNPF SEI 메시지들을 무시하도록 제한될 수 있다.
- [144] `nnpfc_tag_uri`는 기본 NNPF로 이용되는 신경망 또는 `nnpfc_uri`에 의해 특정되는 `nnpfc_id` 값을 사용한 기본 NNPF에 대한 업데이트를 식별하는 IETF RFC 4151에 특정된 신택스 및 시멘틱스를 가지는 태그 URI를 포함할 수 있다. `nnpfc_tag_uri`를 사용하면 중앙 등록 기관 없이도 `nnpfc_uri`가 지정한 신경망 데이터의 형식을 고유하게 식별할 수 있다. "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.
- [145] `nnpfc_uri`는 기본 NNPF로 사용되는 신경망 또는 동일한 `nnpfc_id` 값을 사용하는 기본 NNPF에 관련된 업데이트를 식별하는 IETF Internet Standard 66에 지정된 신택스 및 시멘틱스를 가지는 URI를 포함할 수 있다.
- [146] `nnpfc_property_present_flag`의 값 1은 필터 목적, 입력 포매팅, 출력 포매팅 및 복잡성과 관련된 신택스 엘리먼트들이 존재함을 나타낼 수 있다. `nnpfc_property_present_flag`의 값 0은 필터 목적, 입력 포매팅, 출력 포매팅 및 복잡성과 관련된 신택스 엘리먼트들이 존재하지 않음을 나타낼 수 있다. `nnpfc_base_flag`의 값이 1이면, `nnpfc_property_present_flag`는 1의 값을 가지도록 제한될 수 있다. `nnpfc_property_present_flag`의 값이 0이면, `nnpfc_property_present_flag`의 값이 1인 경우에만 존재할 수 있는 모든 신택스 엘리먼트들의 값들은 SEI 메시지가 업데이트를 제공하는 기본 NNPF를 포함하는 NNPF SEI 메시지 내에서 그들의 대응하는 신택스 엘리먼트들의 값과 동일하게 추론될 수 있다.
- [147] NNPF SEI 메시지 `nnpfcCurr`이 디코딩 순서 상 현재 CLVS 내에서 특정한 `nnpfc_id` 값을 가지는 첫 번째 NNPF SEI 메시지가 아니고, 특정한 `nnpfc_id` 값을

가지는 첫 번째 NNPF SEI 메시지의 반복이 아니며(즉, `nnpfc_base_flag`의 값이 0인 경우), `nnpfc_property_present_flag`의 값이 1인 경우, 아래와 같은 제한들이 적용될 수 있다.

- [148] - NNPF SEI 메시지 내 `nnpfc_purpose`의 값은 디코딩 순서 상 현재 CLVS 내에서 특정한 `nnpfc_id` 값을 가지는 첫 번째 NNPF SEI 메시지 내 `nnpfc_purpose`의 값과 동일해야 한다.
- [149] - NNPF SEI 메시지 내 선택 요소들인 `nnpfc_base_flag` 및 `preceding_nnpfc_complexity_info_present_flag`의 값은 디코딩 순서 상 현재 CLVS 내에서 특정한 `nnpfc_id` 값을 가지는 첫 번째 NNPF SEI 메시지 내 대응하는 선택 요소들의 값과 동일해야 한다.
- [150] - 디코딩 순서 상 현재 CLVS 내에서 특정한 `nnpfc_id` 값을 가지는 첫 번째 NNPF SEI 메시지 내 `nnpfc_complexity_info_present_flag`는 0과 같거나 모두 1과 같아야 하며, 아래와 같은 내용들이 적용될 수 있다
- [151] (1) `nnpfcCurr` 내 `nnpfc_parameter_parameter_type_idc`은 `nnpfcBase` 내 `nnpfc_parameter_parameter_type_idc`와 동일해야 한다.
- [152] (2) `nnpfcCurr` 내 `nnpfc_log2_parameter_bit_length_minus3`가 존재하는 경우, `nnpfcCurr` 내 `nnpfc_log2_parameter_bit_length_minus3`은 `nnpfcBase` 내 `nnpfc_log2_parameter_bit_length_minus3` 이하이어야 한다.
- [153] (3) `nnpfcBase` 내 `nnpfc_num_parameters_idc`가 0과 동일한 경우, `nnpfcCurr` 내 `nnpfc_num_parameters_idc`는 0과 동일해야 한다.
- [154] (4) 그렇지 않은 경우(`nnpfcBase` 내 `nnpfc_num_parameters_idc`가 0보다 큰 경우), `nnpfcCurr` 내 `nnpfc_num_parameters_idc`는 0보다 크거나 `nnpfcBase` 내 `nnpfc_num_parameters_idc` 이하이어야 한다.
- [155] (5) `nnpfcBase` 내 `nnpfc_num_kmac_operations_idc`가 0과 동일한 경우, `nnpfcCurr` 내 `nnpfc_num_kmac_operations_idc`은 0과 동일해야 한다.
- [156] (6) 그렇지 않은 경우(`nnpfcBase` 내 `nnpfc_num_kmac_operations_idc`가 0보다 큰 경우), `nnpfcCurr` 내 `nnpfc_num_kmac_operations_idc`는 0보다 크고 `nnpfcBase` 내 `nnpfc_num_kmac_operations_idc` 이하이어야 한다.
- [157] (7) `nnpfcBase` 내 `nnpfc_total_kilobyte_size`가 0과 동일한 경우, `nnpfcCurr` 내 `nnpfc_total_kilobyte_size`은 0과 동일해야 한다.
- [158] (8) 그렇지 않은 경우(`nnpfcBase` 내 `nnpfc_total_kilobyte_size`가 0보다 큰 경우), `nnpfcCurr` 내 `nnpfc_total_kilobyte_size`는 0보다 크거나 `nnpfcBase` 내 `nnpfc_total_kilobyte_size` 이하이어야 한다.
- [159] `nnpfc_num_input_pics_minus1 + 1`은 NNPF의 입력으로 사용되는 디코딩된 출력 픽처의 수를 나타낼 수 있다. `nnpfc_num_input_pics_minus1`의 값은 0 내지 63 범위 내에 존재하도록 제한될 수 있다. `pictureRateUpsamplingFlag`의 값이 1인 경우, `nnpfc_num_input_pics_minus1`의 값은 0보다 크도록 제한될 수 있다.

[160] NNPF에 대한 입력으로 이용되는 픽처들의 수를 나타내는 변수 numInputPics은 수학적식 1 같이 유도될 수 있다.

[161] [수식1]

$$\text{numInputPics} = \text{nnpfc_num_input_pics_minus1} + 1$$

[162] nnpfc_input_pic_output_flag[i]의 값 1은 i-번째 입력 픽처에 대해 NNPF가 대응하는 출력 픽처를 생성함을 나타낼 수 있다. nnpfc_input_pic_output_flag[i]의 값 0은 i-번째 입력 픽처에 대해 NNPF가 대응하는 출력 픽처를 생성하지 않음을 나타낼 수 있다. nnpfc_num_input_pics_minus1의 값이 0이면 nnpfc_input_pic_output_flag[0]의 값은 1로 추론될 수 있다. pictureRateUpsamplingFlag의 값이 0과 같고 nnpfc_num_input_pics_minus1의 값이 0보다 큰 경우, nnpfc_input_pic_output_flag[i]는 0 내지 nnpfc_num_input_pics_minus1 사이의 범위 내의 i의 적어도 하나의 값에 대해 1과 같도록 제한될 수 있다. nnpfc_input_pic_output_flag[i]는 nnpfc_input_pic_filtering_flag[i]로 지칭될 수 있다.

[163] nnpfc_absent_input_pic_zero_flag의 값 1은 비트스트림에 존재하지 않는 입력 픽처가 0의 샘플 값을 가지는 샘플 어레이들로 표현되는 것을 NNPF가 기대함을 나타낼 수 있다. nnpfc_absent_input_pic_zero_flag의 값 0은 비트스트림에 존재하지 않는 입력 픽처가 비트스트림 내에서 출력 순서 상 최인접한 입력 픽처에 의해 표현되는 것을 NNPF가 기대함을 나타낼 수 있다.

[164] nnpfc_out_sub_c_flag는 chromaUpsamplingFlag의 값이 0인 경우에 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 같아야 한다.

[165] nnpfc_out_colour_format_idc는 colourizationFlag의 값이 1인 경우에 NNPF 출력의 컬러 포맷 및 그에 따른 변수들 outSubWidthC 및 outSubHeightC의 값들을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 1은 NNPF 출력의 컬러 포맷이 4:2:0 포맷이고 outSubWidthC 및 outSubHeightC가 모두 2와 동일함을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 2는 NNPF 출력의 컬러 포맷이 4:2:2 포맷이고 outSubWidthC가 2이며 outSubHeightC가 1임을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값 3은 NNPF 출력의 컬러 포맷이 4:4:4 포맷이고 outSubWidthC 및 outSubHeightC가 모두 1임을 나타낼 수 있다. nnpfc_out_colour_format_idc의 값은 0과 동일하지 않도록 제한될 수 있다. chromaUpsamplingFlag 및 colourizationFlag가 모두 0과 동일한 경우, outSubWidthC 및 outSubHeightC 각각은 SubWidthC 및 SubHeightC와 동일한 것으로 추론될 수 있다.

- [166] `nnpfc_pic_width_num_minus1+1` 및 `nnpfc_pic_width_denom_minus1+1`은 각각 `CroppedWidth`에 대한 NNPF 출력 픽처 너비의 리샘플링 비율에 대한 분자와 분모를 나타낼 수 있다. $(\text{nnpfc_pic_width_num_minus1} + 1)$ 을 $(\text{nnpfc_pic_width_denom_minus1} + 1)$ 으로 나눈 값은 1/16 내지 16의 범위 내에 존재해야 한다. `nnpfc_pic_width_num_minus1` 및 `nnpfc_pic_width_denom_minus1`이 존재하지 않는 경우, `nnpfc_pic_width_num_minus1` 및 `nnpfc_pic_width_denom_minus1` 모두는 0으로 추론될 수 있다.
- [167] 변수 `nnpfcOutputPicWidth`는 `nnpfc_id`에 의해 식별되는 NNPF를 입력 픽처(들)에 적용한 결과에 해당하는 픽처(들)의 루마 샘플 어레이들의 너비를 나타내며, 수학식 2와 같이 유도될 수 있다.
- [168] [수식2]
- $$\text{nnpfcOutputPicWidth} = \text{Ceil}(\text{CroppedWidth} * (\text{nnpfc_pic_width_num_minus1} + 1) \div (\text{nnpfc_pic_width_denom_minus1} + 1))$$
- [169] `nnpfcOutputPicWidth`를 `outSubWidthC`로 나눈 나머지는 0과 같아야 한다.
- [170] `nnpfc_pic_height_num_minus1+1` 및 `nnpfc_pic_height_denom_minus1+1`은 각각 `CroppedHeight`에 대한 NNPF 출력 픽처 높이의 리샘플링 비율에 대한 분자와 분모를 나타낼 수 있다. $(\text{nnpfc_pic_height_num_minus1} + 1)$ 을 $(\text{nnpfc_pic_height_denom_minus1} + 1)$ 으로 나눈 값은 1/16 내지 16의 범위 내에 존재해야 한다. `nnpfc_pic_height_num_minus1` 및 `nnpfc_pic_height_denom_minus1`이 존재하지 않는 경우, `nnpfc_pic_height_num_minus1` 및 `nnpfc_pic_height_denom_minus1` 모두는 0으로 추론될 수 있다.
- [171] 변수 `nnpfcOutputPicHeight`는 `nnpfc_id`에 의해 식별되는 NNPF를 입력 픽처(들)에 적용한 결과에 해당하는 픽처(들)의 루마 샘플 어레이들의 높이를 나타내며, 수학식 3과 같이 유도될 수 있다.
- [172] [수식3]
- $$\text{nnpfcOutputPicHeight} = \text{Ceil}(\text{CroppedHeight} * (\text{nnpfc_pic_height_num_minus1} + 1) \div (\text{nnpfc_pic_height_denom_minus1} + 1))$$
- [173] `nnpfcOutputPicHeight`를 `outSubHeightC`로 나눈 나머지는 0과 같아야 한다.
- [174] `nnpfc_pic_width_num_minus1`, `nnpfc_pic_width_denom_minus1`, `nnpfc_pic_height_num_minus1`, 및 `nnpfc_pic_height_denom_minus1`이 존재하는 경우, 아래의 제한들 중 적어도 하나가 참(true)일 수 있다.
- [175] - `nnpfcOutputPicWidth`의 값은 `CroppedWidth`와 동일하지 않음
- [176] - `nnpfcOutputPicHeight`의 값은 `CroppedHeight`와 동일하지 않음
- [177] `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. `nnpfc_interpolated_pics[i]`의 값은 0 내지 63 범위 내로 제한될 수 있다.

nnpfc_interpolated_pics[i]의 값은 0 내지 nnpfc_num_input_pics_minus1 - 1 범위 내의 적어도 하나의 i에 대해 0보다 크도록 제한될 수 있다.

- [178] 대응하는 입력 픽처를 가지며 NNPF의 출력 텐서(tensor) 내에 존재하는 픽처들의 수를 나타내는 변수 NumInpPicsInOutputTensor, 대응하는 입력 픽처를 가지며 NNPF의 출력 텐서 내에 존재하는 idx-번째 픽처의 입력 픽처 인덱스를 나타내는 변수 InpIdx[idx], 및 NNPF의 출력 텐서 내에 존재하는 픽처들의 전체 개수를 나타내는 변수 numOutputPics는 표 6에 나타낸 바와 같이 유도될 수 있다.

- [179] [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 ]

```

- [180] 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 세 번째 차원과 NNPF의 결과인 출력 텐서 outputTensor가 현재 채널에 사용됨을 나타낼 수 있다. 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. 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.

- [181] nnpfc_inp_format_idc는 입력 픽처의 샘플 값을 NNPF의 입력 값으로 변환하는 방법을 나타낼 수 있다. nnpfc_inp_format_idc의 값이 1인 경우, NNPF의 입력 값들은 실수일 수 있으며, 함수들 InpY() 및 InpC()는 수학적 식 4와 같이 나타낼 수 있다.

[182] [수식4]

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

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

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

[184] [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 )

```

[185] 변수 `inpTensorBitDepthY`는 아래에서 설명되는 선택스 요소

`nnpfc_inp_tensor_luma_bitdepth_minus8`로부터 유도될 수 있다. `inpTensorBitDepthC`는 아래에서 설명되는 선택스 요소 `nnpfc_inp_tensor_chroma_bitdepth_minus8`로부터 유도될 수 있다. 1보다 큰 `nnpfc_inp_format_idc`의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않아야 한다. 디코더들은 `nnpfc_inp_format_idc`의 예약된 값들을 포함하는 NNPF SEI 메시지를 무시해야 한다.

[186] 0보다 큰 `nnpfc_auxiliary_inp_idc`는 NNPF의 입력 텐서에 보조 입력 데이터가 존재함을 나타낼 수 있다. `nnpfc_auxiliary_inp_idc`의 값 0은 보조 입력 데이터가 입력 텐서에 존재하지 않음을 나타낼 수 있다. `nnpfc_auxiliary_inp_idc`의 값 1은 보조 입력 데이터가 표 10 내지 표 12에 나타낸 방법을 통해 유도됨을 나타낼 수 있다. `nnpfc_auxiliary_inp_idc`의 값은 비트스트림에서 0 내지 1의 범위에 존재해야 한다. `nnpfc_auxiliary_inp_idc`에 대한 2 내지 255의 값들은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 2 내지 255의 범위의 `nnpfc_auxiliary_inp_idc`를 가지는 NNPF SEI 메시지를 무시해야 한다. 255보다 큰 `nnpfc_auxiliary_inp_idc`의 값은 비트스트림에 존재하지 않으며 향후 사용을 위해 예약되지 않는다.

[187] `nnpfc_inp_order_idc`는 NNPF에 대한 입력 텐서를 형성하기 위해 입력 픽처의 샘플 어레이들을 정렬하는 방법을 나타낼 수 있다. `nnpfc_inp_order_idc`의 값은 비트스트림에서 0 내지 3의 범위에 존재해야 한다. `nnpfc_inp_order_idc`에 대한 4 내지 255의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에 존재

하지 않는다. 디코더들은 4 내지 255의 범위에 존재하는 `nnpfc_inp_order_idc`를 가지는 NNPFCE SEI 메시지를 무시해야 한다. 255보다 큰 `nnpfc_inp_order_idc`의 값은 비트스트림에 존재하지 않으며 향후 사용을 위해 예약되지 않는다. `ChromaFormatIdc`의 값이 1이 아닌 경우, `nnpfc_inp_order_idc`의 값은 3이 아니어야 한다. `ChromaFormatIdc`의 값이 0인 경우, `nnpfc_inp_order_idc`의 값은 3이어야 한다. `chromaUpsamplingFlag`의 값이 1인 경우, `nnpfc_inp_order_idc`의 값은 0이 아니어야 한다.

[188] 표 8에는 `nnpfc_inp_order_idc` 값에 대한 설명이 나타나 있다.

[189] [표8]

<code>nnpfc_inp_order_idc</code>	Description
0	<code>nnpfc_auxiliary_inp_idc</code> 의 값이 0이면, 각 입력 픽처에 대한 입력 텐서에 루마 행렬이 1개씩 존재할 수 있으며, 채널 수는 1일 수 있다. 그렇지 않고, <code>nnpfc_auxiliary_inp_idc</code> 의 값이 1이면 루마 행렬과 보조 입력 행렬이 하나씩 존재할 수 있으며, 채널 수는 2일 수 있다.
1	<code>nnpfc_auxiliary_inp_idc</code> 의 값이 0이면, 입력 텐서에 크로마 행렬이 2개 존재할 수 있으며 채널 개수는 2일 수 있다. 그렇지 않고, <code>nnpfc_auxiliary_inp_idc</code> 의 값이 1이면 크로마 행렬 2개와 보조 입력 행렬이 1개 존재할 수 있으며 채널 개수는 3일 수 있다.
2	<code>nnpfc_auxiliary_inp_idc</code> 의 값이 0이면, 입력 텐서에는 루마 행렬 1개와 크로마 행렬 2개가 존재할 수 있으며, 채널 개수는 3일 수 있다. 그렇지 않고, <code>nnpfc_auxiliary_inp_idc</code> 의 값이 1이면 루마 행렬 1개, 크로마 행렬 2개, 보조 입력 행렬 1개가 존재할 수 있으며, 채널 개수는 4일 수 있다.
3	<code>nnpfc_auxiliary_inp_idc</code> 의 값이 0이면, 입력 텐서에는 루마 행렬 4개와 크로마 행렬 2개가 존재할 수 있으며, 채널 수는 6일 수 있다. 그렇지 않고, <code>nnpfc_auxiliary_inp_idc</code> 의 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. <code>nnpfc_inp_order_idc</code> 는 입력 크로마 포맷이 4:2:0인 경우에만 사용할 수 있다.
4 ... 255	예약됨

[190] `nnpfc_inp_tensor_luma_bitdepth_minus8+8`은 입력 정수 텐서 내 루마 샘플 값들의 비트 뎀스를 나타낼 수 있다. `inpTensorBitDepthY`는 수식 5와 같이 유도될 수 있다.

[191] [수식5]

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

[192] `nnpfc_inp_tensor_luma_bitlength_minus8`의 값은 0 내지 24의 범위에 존재해야 하는 것으로 제한될 수 있다.

[193] `nnpfc_inp_tensor_chroma_bitdepth_minus8+8`은 입력 정수 텐서 내 크로마 샘플 값들의 비트 맵스를 나타낼 수 있다. `inpTensorBitDepthC`의 값은 수학식 6과 같이 유도될 수 있다.

[194] [수식6]

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

[195] `nnpfc_inp_tensor_chroma_bitdepth_minus8`의 값은 0 내지 24의 범위에 존재해야 하는 것으로 제한될 수 있다.

[196] `nnpfc_auxiliary_inp_idc`의 값이 1과 동일한 경우, 변수 `strengthControlScaledVal`은 표 9와 같이 유도될 수 있다.

[197] [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 ]
```

[198] 패치는 픽처의 성분(e.g., 루마 또는 크로마 성분)으로부터의 샘플들의 직사각형 어레이일 수 있다.

[199] 입력 텐서에 포함된 샘플 패치의 좌상단 샘플 위치를 나타내는 주어진 수직 샘플 좌표 `cTop` 및 수평 샘플 좌표 `cLeft`에 대한 입력 텐서 `inputTensor`를 유도하는 프로세스 `DeriveInputTensors()`는 표 10 내지 표 12의 결합과 같이 나타낼 수 있다.

[200] [표10]

```

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 ]
    }
}

```

[201] [圖11]

```

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 ]
}
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 ]
    }
}

```

[203] nnpfc_out_format_idc의 값 0은 후속하는 후처리 또는 디스플레이를 위해 요구되는 비트맵스 bitDepth에 대해, NNPF에 의해 출력된 샘플 값이 0 내지 1의 값 범위가 0 내지 $(1 \ll \text{bitDepth}) - 1$ 의 부호 없는 정수 값 범위에 선형적으로 매핑되는 실수임을 나타낼 수 있다. nnpfc_out_format_idc의 값 1은 NNPF에 의해 출력된 루마 샘플 값이 0 내지 $(1 \ll \text{outTensorBitDepth}_Y) - 1$ 범위의 부호 없는 정수임을 나타낼 수 있으며, NNPF에 의해 출력된 크로마 샘플 값이 0 내지 $(1 \ll (1 \ll \text{outTensorBitDepth}_C) - 1)$ 범위의 부호 없는 정수임을 나타낼 수 있다. 1보다 큰 nnpfc_out_format_idc의 값은 향후 사용을 위해 예약될 수 있으며 비트스트림에는 존재하지 않는다. 디코더들은 nnpfc_out_format_idc의 예약된 값을 포함하는 NNPF SEI 메시지를 무시해야 한다.

[204] nnpfc_out_order_idc는 NNPF로부터 출력되는 샘플들의 출력 순서를 나타낼 수 있다. nnpfc_out_order_idc의 값은 비트스트림에서 0 내지 3의 범위에 존재해야 한다. nnpfc_out_order_idc에 대한 4 내지 255의 값들은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 4 내지 255의 범위에 존재하는 nnpfc_out_order_idc를 가지는 NNPF SEI 메시지를 무시해야 한다. 255보다 큰 nnpfc_out_order_idc의 값은 비트스트림에 존재하지 않으며 향후 사용을 위해 예약되지 않는다. chromaUpsamplingFlag의 값이 1인 경우, nnpfc_out_order_idc의 값은 0 또는 3과 같지 않아야 한다. colourizationFlag의 값이 1인 경우, nnpfc_out_order_idc의 값은 0과 같지 않아야 한다.

[205] 표 13에는 nnpfc_out_order_idc의 값에 대한 설명이 나타나 있다.

[206] [표13]

nnpfc_out_order_idc	Description
0	루마 행렬만이 출력 텐서에 존재하므로, 채널의 수는 1과 같을 수 있다.
1	오직 크로마 행렬들만이 출력 텐서에 존재하므로, 채널의 수는 2와 같을 수 있다.
2	루마 및 크로마 행렬들이 출력 텐서에 존재하므로, 채널의 수는 3과 같을 수 있다.
3	4개의 루마 행렬들과 2개의 크로마 행렬들이 출력 텐서에 존재하므로, 채널의 수는 6과 같을 수 있다. nnpfc_out_order_idc 출력 크로마 포맷이 4:2:0인 경우에만 이용될 수 있다.
4 ... 255	예약됨

[207] nnpfc_out_tensor_luma_bitdepth_minus8+8은 출력 정수 텐서 내 루마 샘플 값들의 비트 맵스를 나타낼 수 있다. nnpfc_out_tensor_luma_bitdepth_minus8의 값은 0 내지 24의 범위에 존재해야 한다. outTensorBitDepth_Y의 값은 수학적 식 7과 같이 유도될 수 있다.

[208] [수식7]

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

[209] nnpfc_out_tensor_chroma_bitdepth_minus8+8은 출력 정수 텐서 내 크로마 샘플 값들의 비트 맵스를 나타낼 수 있다. nnpfc_out_tensor_chroma_bitdepth_minus8의 값은 0 내지 24의 범위에 존재해야 한다. outTensorBitDepth_C의 값은 수학적 식 8과 같이 유도될 수 있다.

[210] [수식8]

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

[211] bitDepthUpsamplingFlag의 값이 1인 경우, nnpfc_out_format_idc의 값은 1과 동일해야 하며, 아래의 제한들 중에서 적어도 하나가 참일 수 있다.

[212] - nnpfc_out_tensor_luma_bitdepth_minus8+8이 존재하며, outTensorBitDepth_Y이 BitDepth_Y보다 큼

[213] - nnpfc_out_tensor_chroma_bitdepth_minus8+8이 존재하며, outTensorBitDepth_C이 BitDepth_C보다 큼

[214] nnpfc_in_tensor_luma_bitdepth_minus8, nnpfc_in_tensor_chroma_bitdepth_minus8, nnpfc_out_tensor_luma_bitdepth_minus8, 및 nnpfc_out_tensor_chroma_bitdepth_minus8이 존재하고 outTensorBitDepth_Y이 inpTensorBitDepth_Y보다 크면, outTensorBitDepth_C는 inpTensorBitDepth_C보다 작지 않아야 한다. nnpfc_in_tensor_luma_bitdepth_minus8, nnpfc_in_tensor_chroma_bitdepth_minus8, nnpfc_out_tensor_luma_bitdepth_minus8, 및 nnpfc_out_tensor_chroma_bitdepth_minus8이 존재하고,

outTensorBitDepth_C이 inpTensorBitDepth_C보다 크면, outTensorBitDepth_Y은 inpTensorBitDepth_Y보다 작지 않아야 한다.

[215] 입력 텐서에 포함된 샘플 패치의 좌상단 샘플 위치를 나타내는 주어진 수직 샘플 좌표 cTop 및 수평 샘플 좌표 cLeft에 대한 출력 텐서로부터 필터링된 샘플 어레이들인 FilteredYPic, FilteredCbPic, 및 FilteredCrPic 내 샘플 값들을 유도하는 프로세스 StoreOutputTensors()는 표 14 및 표 15의 결합과 같이 나타낼 수 있다.

[216] [표14]

```

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
        if ( yY < nnpfcOutputPicHeight && xY < nnpfcOutputPicWidth )

```

[217] [표15]

```

        if( !nnpfc_component_last_flag ) {
            FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ 0 ][ yP ][ xP ]
            FilteredCbPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ 1 ][ yP ][ xP ]
            FilteredCrPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ 2 ][ yP ][ xP ]
        } else {
            FilteredYPic[ i ][ xY ][ yY ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 0 ]
            FilteredCbPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 1 ]
            FilteredCrPic[ i ][ xC ][ yC ] = outputTensor[ 0 ][ i ][ yP ][ xP ][ 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 ]
        }
    }
}
}

```

- [218] nnpfc_separate_colour_description_present_flag의 값 1은 NNPF로 인한 픽처에 대한 기본 색상(colour primaries), 변환 특성들, 행렬 계수들, 및 스케일링 및 오프셋 값들의 고유한 조합(distinct combination)이 SEI 메시지 선택스 구조에 지정됨을 나타낼 수 있다. nnpfc_separate_colour_description_present_flag의 값 0은 NNPF로 인한 픽처의 기본 색상, 변환 특성들, 행렬 계수들, 및 스케일링 및 오프셋 값들의 조합이 CLVS에 대한 VUI 파라미터들에 표시된 것과 동일함을 나타낼 수 있다.
- [219] nnpfc_colour_primaries는 다음을 제외하고 vui_colour_primaries 선택스 요소에 대해 정의된 것과 동일한 시멘틱스를 가질 수 있다.
- [220] - nnpfc_colour_primaries는 CLVS에 사용되는 기본 색상이 아닌, SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처의 기본 색상을 나타낼 수 있다.
- [221] - NNPF SEI 메시지에 nnpfc_colour_primaries가 존재하지 않으면, nnpfc_colour_primaries의 값은 vui_colour_primaries의 값과 동일한 것으로 추론될 수 있다.
- [222] nnpfc_transfer_characteristics는 다음을 제외하고 vui_transfer_characteristics 선택스 요소에 대해 정의된 것과 동일한 시멘틱스를 가질 수 있다.

- [223] - `nnpfc_transfer_characteristics`는 CLVS에 사용되는 변환 특성이 아닌, SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처의 변환 특성을 나타낼 수 있다.
- [224] - NNPF SEI 메시지에 `nnpfc_transfer_characteristics`가 존재하지 않으면, `nnpfc_transfer_characteristics`의 값은 `vui_transfer_characteristics`의 값과 동일한 것으로 추론될 수 있다.
- [225] `nnpfc_matrix_coeffs`는 그린, 블루 및 레드 또는 Y, Z 및 X 프라이머리들로부터 루마 및 크로마 신호들을 유도하기 위해 이용되는 수식들(equations)을 설명할 수 있다. `nnpfc_matrix_coeffs`의 시멘틱스는 SEI 메시지에 지정된 NNPF를 적용한 결과로 나타나는 픽처들에 적용될 수 있으며, `MatrixCoefficients`에 대해 나타낸 바와 같이 `BitDepthY` 및 `BitDepthC` 각각은 `outTensorBitDepthY` 및 `outTensorBitDepthC`와 동일할 수 있다. `nnpfc_matrix_coeffs`가 NNPF SEI 메시지에 존재하지 않는 경우, `nnpfc_matrix_coeffs`의 값은 `vui_matrix_coeffs`와 동일하게 추론될 수 있다.
- [226] 아래의 조건들이 만족하는 경우를 제외하면, `nnpfc_matrix_coeffs`은 0과 동일하지 않아야 한다.
- [227] - `nnpfc_out_tensor_chroma_bitdepth_minus8`이 `nnpfc_out_tensor_luma_bitdepth_minus8`와 동일함
- [228] - `nnpfc_out_order_idc`이 2와 동일하고, `outSubHeightC`이 1과 동일하며, `outSubWidthC`이 1과 동일함
- [229] 아래의 조건들 중 어느 하나가 만족하는 경우를 제외하면, `nnpfc_matrix_coeffs`은 8과 동일하지 않아야 한다.
- [230] - `nnpfc_out_tensor_chroma_bitdepth_minus8`이 `nnpfc_out_tensor_luma_bitdepth_minus8`와 동일함
- [231] - `nnpfc_out_tensor_chroma_bitdepth_minus8`는 `nnpfc_out_tensor_luma_bitdepth_minus8+1`과 동일하며, `nnpfc_out_order_idc`는 2와 동일하고, `outSubHeightC`는 2와 동일하며, `outSubWidthC`는 1과 동일함.
- [232] `nnpfc_full_range_flag`는 `nnpfc_matrix_coeffs`에 의해 특정된 것과 같이 행렬 계수들과 관련하여 적용된 스케일링 및 오프셋 값들을 나타낼 수 있다. `nnpfc_full_range_flag`의 시멘틱스는 `VideoFullRangeFlag`에 대해 특정된 것과 같을 수 있다. `nnpfc_full_range_flag`이 존재하지 않는 경우, `nnpfc_full_range_flag`의 값은 0과 동일하게 추론될 수 있다.
- [233] `nnpfc_chroma_loc_info_present_flag`의 값 1은 NNPF SEI 메시지 내에 `nnpfc_chroma_sample_loc_type_frame` 선택스 요소가 존재함을 나타낼 수 있다. `nnpfc_chroma_loc_info_present_flag`의 값 0은 NNPF SEI 메시지 내에 `nnpfc_chroma_sample_loc_type_frame` 선택스 요소가 존재하지 않음을 나타낼 수 있다. `colourizationFlag`의 값이 0이거나 `nnpfc_out_colour_format_idc`가 1이 아닌 경우, `nnpfc_chroma_loc_info_present_flag`의 값은 0과 동일하게 제한될 수 있다.

- [234] `nnpfc_chroma_sample_loc_type_frame`이 6과 동일하지 않고 `nnpfc_out_colour_format_idc`가 1과 동일한 경우, `nnpfc_chroma_sample_loc_type_frame`는 출력 픽처들의 크로마 샘플들의 위치를 나타낼 수 있다. `nnpfc_chroma_sample_loc_type_frame`이 6과 동일하고 `nnpfc_out_colour_format_idc`가 1과 동일한 경우는 크로마 샘플들의 위치를 알 수 없거나 이 크로마 샘플들의 위치가 지정되지 않았거나 다른 방법으로 지정되었음을 나타낼 수 있다. `nnpfc_chroma_sample_loc_type_frame`의 값은 0 내지 6의 범위 내에 존재해야 한다.
- [235] `nnpfc_overlap`은 NNPF의 인접한 입력 텐서들의 오버랩핑(overlapping) 수평 및 수직 샘플 수를 나타낼 수 있다. `nnpfc_overlap`의 값은 0 내지 16 383 범위 내에 존재해야 한다.
- [236] `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 받아들인다는 것을 나타낼 수 있다. `nnpfc_constant_patch_size_flag`의 값 0은 NNPF가 너비 `inpPatchWidth` 및 높이 `inpPatchHeight`를 가지는 임의의 패치 크기를 입력으로 받아들인다는 것을 나타낼 수 있다. 여기서, `inpPatchWidth + 2*nnpfc_overlap`와 동일한 확장된 패치의 너비(즉, 패치와 오버랩핑 영역을 더한 것)는 `nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2*nnpfc_overlap`의 양의 정수배이며, `inpPatchHeight + 2*nnpfc_overlap`와 동일한 확장된 패치의 높이는 `nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2*nnpfc_overlap`의 양의 정수배이다.
- [237] `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 $\text{Min}(32\ 766, \text{CroppedWidth} - 1)$ 의 범위에 존재해야 한다.
- [238] `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 $\text{Min}(32\ 766, \text{CroppedHeight} - 1)$ 의 범위에 존재해야 한다.
- [239] `nnpfc_extended_patch_width_cd_delta_minus1 + 1 + 2*nnpfc_overlap`은 `nnpfc_constant_patch_size_flag`의 값이 0일 때, NNPF에 입력하는 데 요구되는 확장된 패치의 너비의 허용된 값의 공통 제수(common divisor)를 나타낼 수 있다. `nnpfc_extended_patch_width_cd_delta_minus1`의 값은 0 내지 $\text{Min}(32\ 766, \text{CroppedWidth} - 1)$ 의 범위에 존재해야 한다.
- [240] `nnpfc_extended_patch_height_cd_delta_minus1 + 1 + 2*nnpfc_overlap`은 `nnpfc_constant_patch_size_flag`의 값이 0일 때 NNPF에 입력하는 데 요구되는 확장된 패치의 높이의 허용된 값의 공통 제수(common divisor)를 나타낼 수 있

다. `nnpfc_extended_patch_height_cd_delta_minus1`의 값은 0 내지 $\text{Min}(32766, \text{CroppedHeight} - 1)$ 의 범위에 존재해야 한다.

- [241] 변수들 `inpPatchWidth` 및 `inpPatchHeight` 각각은 패치 크기 너비와 패치 크기 높이로 설정될 수 있다.
- [242] `nnpfc_constant_patch_size_flag`의 값이 0이면 다음이 적용될 수 있다.
- [243] - `inpPatchWidth` 및 `inpPatchHeight`의 값은 외부 수단에 의해 제공되거나 포스트-프로세서에 의해 설정될 수 있다.
- [244] - `inpPatchWidth+2*nnpfc_overlap`의 값은 `nnpfc_extended_patch_width_cd_delta_minus1+1+2*nnpfc_overlap`의 양의 정수배이어야 하며, `inpPatchWidth`는 `CroppedWidth`보다 작거나 같아야 한다. `inpPatchHeight+2*nnpfc_overlap`의 값은 `nnpfc_extended_patch_height_cd_delta_minus1+1+2*nnpfc_overlap`의 양의 정수배이어야 하며, `inpPatchHeight`는 `CroppedHeight`보다 작거나 같아야 한다.
- [245] 그렇지 않으면(`nnpfc_constant_patch_size_flag`의 값이 1이면), `inpPatchWidth`의 값은 `nnpfc_patch_width_minus1+1`과 동일하게 설정되고 `inpPatchHeight`의 값은 `nnpfc_patch_height_minus1+1`과 동일하게 설정될 수 있다.
- [246] 변수들 `outPatchWidth`, `outPatchHeight`, `horCScaling`, `verCScaling`, `outPatchCWidth` 및 `outPatchCHeight`는 표 16와 같이 유도될 수 있다.
- [247] [표 16]

```

outPatchWidth = ( nnpfcOutputPicWidth * inpPatchWidth ) / CroppedWidth
outPatchHeight = ( nnpfcOutputPicHeight * inpPatchHeight ) / CroppedHeight
horCScaling = SubWidthC / outSubWidthC
verCScaling = SubHeightC / outSubHeightC
outPatchCWidth = outPatchWidth * horCScaling
outPatchCHeight = outPatchHeight * verCScaling

```

- [248] `outPatchWidth * CroppedWidth`는 `nnpfcOutputPicWidth * inpPatchWidth`와 같아야 하며, `outPatchHeight * CroppedHeight`는 `nnpfcOutputPicHeight * inpPatchHeight`와 같아야 한다.
- [249] `nnpfc_padding_type`은 표 17에 설명된 대로 입력 픽처의 경계 외부에 있는 샘플 위치를 참조할 때의 패딩 프로세스를 나타낼 수 있다. `nnpfc_padding_type`의 값은 0 내지 4의 범위에 존재해야 한다. `nnpfc_padding_type`에 대한 5 내지 15의 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더는 5 내지 15 범위의 `nnpfc_padding_type`을 가지는 NNPF SEI 메시지를 무시해야 한다. 15를 초과하는 `nnpfc_padding_type`의 값은 비트스트림에 존재하지 않아야 하며 향후 사용을 위해 예약될 수 없다.

[250] [표17]

nnpfc_padding_type	Description
0	zero padding
1	replication padding
2	reflection padding
3	wrap-around padding
4	fixed padding
5 ... 15	예약됨

[251] nnpfc_luma_padding_val은 nnpfc_padding_type의 값이 4일 때 패딩에 사용할 루마 값을 나타낼 수 있다. nnpfc_luma_padding_val의 값은 0 내지 $(1 \ll \text{BitDepthY}) - 1$ 의 범위 내에 존재해야 한다.

[252] nnpfc_cb_padding_val은 nnpfc_padding_type의 값이 4일 때 패딩에 사용할 Cb 값을 나타낼 수 있다. nnpfc_cb_padding_val의 값은 0 내지 $(1 \ll \text{BitDepthC}) - 1$ 의 범위 내에 존재해야 한다.

[253] nnpfc_cr_padding_val은 nnpfc_padding_type의 값이 4일 때 패딩에 사용할 Cr 값을 나타낼 수 있다. nnpfc_cr_padding_val의 값은 0 내지 $(1 \ll \text{BitDepthC}) - 1$ 의 범위 내에 존재해야 한다.

[254] 수직 샘플 위치 y, 수평 샘플 위치 x, 픽처 높이 picHeight, 픽처 너비 picWidth, 샘플 어레이 CroppedPic 및 공통 인덱스 cIdx(루마에 대해 0, Cb에 대해 1 및 Cr에 대해 2)의 입력을 가지는 함수 InpSampleVal(y, x, picHeight, picWidth, croppedPic, cIdx)는 표 18과 같이 유도된 SampleVal의 값을 반환할 수 있다.

[255] [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 ]

```

[256] NNPF PostProcessingFilter()는 NNPF SEI 메시지의 시멘틱스들에서 유도된 대상 NNPF일 수 있다.

[257] 표 19의 프로세스는 NNPF PostProcessingFilter()를 사용하여 패치 방식으로 필터링하여 필터링된 및/또는 보간된 픽처를 생성하는 데 사용될 수 있으며, 필터링된 및/또는 보간된 픽처는 nnpfc_out_order_idc에 의해 표시된 바와 같이 Y 샘플 어레이 FilteredYPic, Cb 샘플 어레이 FilteredCbPic 및 Cr 샘플 어레이 FilteredCrPic을 포함할 수 있다.

[258] [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()
        }
}

```

[259] 인덱스 i를 가지는 NNPF 생성(NNPF-generated) 픽처는 존재하는 경우에 샘플 어레이들인 FilteredYPic[i], FilteredCbPic[i] 및 FilteredCrPic[i]를 포함할 수 있다. NNPF 생성 픽처는 오버랩 영역들을 포함하지 않을 수 있다.

[260] NNPF 프로세스는 표 19에 정의된 프로세스에 따라 NNPF 생성 픽처를 인덱스가 증가하는 순서대로 출력하는 것으로 구성되며, 여기서 NNPF에 의해 보간된 모든 NNPF 생성 픽처들이 출력되고 NNPPA SEI 메시지의 시멘틱스에 지정된 대로 NNPF에 입력된 픽처에 해당하는 모든 NNPF 생성 픽처들이 출력될 수 있다.

[261] nnpfc_complexity_info_present_flag의 값 1은 nnpfc_id와 연관된 NNPF의 복잡도를 나타내는 하나 이상의 신택스 요소가 존재함을 나타낼 수 있다. nnpfc_complexity_info_present_flag의 값 0은 nnpfc_id와 연관된 NNPF의 복잡도를 나타내는 신택스 요소가 존재하지 않음을 나타낼 수 있다.

[262] nnpfc_parameter_type_idc의 값 0은 신경망이 정수 파라미터들만을 사용함을 나타낼 수 있다. nnpfc_parameter_type_flag의 값 1은 신경망이 부동 소수점 또는 정수 파라미터들을 사용할 수 있음을 나타낼 수 있다. nnpfc_parameter_type_idc의 값 2는 신경망이 이진 파라미터들만을 사용함을 나타낼 수 있다. nnpfc_parameter_type_idc의 값 3은 향후 사용을 위해 예약될 수 있으며, 비트스

트림에는 존재하지 않는다. 디코더들은 `nnpfc_parameter_type_idc`의 값이 3인 NNPF SEI 메시지를 무시해야 한다.

- [263] `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.
- [264] `nnpfc_num_parameters_idc`는 NNPF에 대한 신경망 파라미터들의 최대 수를 2 048의 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 비트스트림에는 존재하지 않는다. 디코더들은 52보다 큰 `nnpfc_num_parameters_idc`를 가진 NNPF SEI 메시지를 무시해야 한다.
- [265] `nnpfc_num_parameters_idc`의 값이 0보다 큰 경우, 변수 `maxNumParameters`는 수학적 9와 같이 유도될 수 있다.
- [266] [수식9]
- $$\text{maxNumParameters} = (2\,048 \ll \text{nnpfc_num_parameters_idc}) - 1$$
- [267] NNPF의 신경망 파라미터들의 수는 `maxNumParameters`보다 작거나 같도록 제한될 수 있다.
- [268] 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 * 1 000`보다 작거나 같음을 나타낼 수 있다. `nnpfc_num_kmac_operations_idc`의 값 0은 상기 네트워크의 곱누산 연산의 최대 수를 알 수 없음을 나타낼 수 있다. `nnpfc_num_kmac_operations_idc`의 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [269] 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 8 000으로 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.
- [270] `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`은 `nnpfc_reserved_metadata_extension`의 길이(비트)를 나타낼 수 있다. `nnpfc_metadata_extension_num_bits`은 0과 동일해야 한다.

nnpfc_metadata_extension_num_bits에 대한 1 내지 2 048 범위 내 값들은 향후 사용을 위해 예약되며, 비트스트림에 존재하지 않는다. 디코더들은 0 내지 2 048 범위의 nnpfc_metadata_extension_num_bits의 임의의 값을 허용할 수 있다. 2 048보다 큰 nnpfc_metadata_extension_num_bits의 값은 비트스트림에 존재하지 않으며, 향후 사용을 위해 예약될 수 없다.

- [271] nnpfc_reserved_metadata_extension는 비트스트림에 존재하지 않는다. 그러나, 디코더들은 nnpfc_reserved_metadata_extension의 존재와 그 값을 무시해야 한다. nnpfc_reserved_metadata_extension이 존재하는 경우, nnpfc_reserved_metadata_extension의 길이는 nnpfc_metadata_extension_num_bits이 동일할 수 있다.
- [272] nnpfc_reserved_zero_bit_b는 비트스트림에서 0과 동일해야 한다. 디코더들은 nnpfc_reserved_zero_bit_b가 0이 아닌 NNPF SEI 메시지를 무시해야 한다.
- [273] nnpfc_payload_byte[i]는 비트스트림의 i번째 바이트를 포함할 수 있다. i의 모든 존재하는 값에 대한 바이트 시퀀스 nnpfc_payload_byte[i]는 ISO/IEC 15938-17을 준수하는 완전한 비트스트림이어야 한다.
- [274] **신경망 포스트-필터 활성화(Neural-network post-filter activation, NNFA)**
- [275] NNFA에 대한 선택스 구조가 표 20에 나타나 있다.
- [276] [표20]

nn_post_filter_activation(payloadSize) {	Descriptor
nnpfa_target_id	ue(v)
nnpfa_cancel_flag	u(1)
if(!nnpfa_cancel_flag) {	
nnpfa_target_base_flag	u(1)
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)
}	
}	

- [277] 표 20의 NNFA 선택스 구조는 SEI 메시지 형태로 시그널링될 수 있다. 표 20의 NNFA 선택스 구조를 시그널링하는 SEI 메시지는 NNFA SEI 메시지라고 지칭될 수 있다.
- [278] NNFA SEI 메시지는 픽처 세트의 후처리 필터링을 위해 nnpfa_target_id에 의해 식별되는 대상 신경망 후처리 필터(NNPF)의 가능한 사용을 활성화하거나 비활성화할 수 있다. NNPF가 활성화된 특정 픽처에 대해, 대상 NNPF는 다음과 같이 유도될 수 있다.
- [279] - nnpfa_target_base_flag가 1이면, 대상 NNPF는 nnpfa_target_id와 동일한 nnpfc_id를 가지는 기본 NNPF일 수 있다.
- [280] - 그렇지 않으면(nnpfa_target_base_flag가 0이면), 대상 NNPF는 nnpfa_target_id와 동일한 nnpfc_id를 가지는 마지막 NNPF SEI 메시지에 의해 특정되는 NNPF

일 수 있다. 여기서, 마지막 NNPFC SEI 메시지는 디코딩 순서 상 현재 픽처의 첫 번째 VCL NAL 유닛에 선행하면서 기본 NNPF를 포함하는 NNPFC SEI 메시지의 반복에 해당하지 않을 수 있다.

- [281] NNPF가 다른 목적을 위해 사용되거나 다른 색상 성분을 필터링하는 경우에, 동일한 픽처에 대해 여러 개의 NNPF SEI 메시지가 존재할 수 있다.
- [282] `nnpfa_target_id`는 현재 픽처와 관련되고 `nnfpa_target_id`와 동일한 `nnpfc_id`를 갖는 하나 이상의 NNPFC SEI 메시지에 의해 지정되는 대상 NNPF를 나타낼 수 있다. `nnpfa_target_id`의 값은 0 내지 $2^{32} - 2$ 의 범위에 존재해야 한다.
- [283] 특정 값의 `nnpfa_target_id`를 가진 NNPF SEI 메시지는 다음 조건 중 하나 또는 둘 모두가 참이 아닌 한 현재 PU에 존재하지 않아야 한다.
- [284] - 현재 CLVS 내에서, 디코딩 순서에서 현재 PU에 우선하는 PU 내에 존재하는 특정 값의 `nnpfa_target_id`와, 동일한 `nnpfc_id`를 갖는 NNPFC SEI 메시지가 존재함
- [285] - 현재 PU의 특정 값의 `nnpfa_target_id`와 동일한 `nnpfc_id`를 갖는 NNPFC SEI 메시지가 존재함
- [286] PU가 특정 값의 `nnpfc_id`를 갖는 NNPFC SEI 메시지와, 특정 값의 `nnpfc_id`와 동일한 `nnpfa_target_id`를 갖는 NNPF SEI 메시지를 모두 포함하는 경우, NNPFC SEI 메시지는 디코딩 순서에서 NNPF SEI 메시지보다 선행해야 한다.
- [287] `nnpfa_cancel_flag`의 값 1은 현재 SEI 메시지와 동일한 `nnpfa_target_id`를 갖는 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 0과 동일한 `nnpfa_cancel_flag`를 갖는 다른 NNPF SEI 메시지에 의해 활성화되지 않는 한, 더 이상 사용되지 않는다. `nnpfa_cancel_flag`의 값 0은 `nnpfa_target_base_flag`, `nnpfa_persistence_flag` 및 `nnpfa_num_output_entries`가 뒤따른다는 것을 나타낼 수 있다.
- [288] `nnpfa_target_base_flag`의 값 1은 대상 NNPF가 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 기본 NNPF임을 나타낼 수 있다. `nnpfa_target_base_flag`의 값 0은 대상 NNPF가 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 마지막 NNPFC SEI 메시지에 의해 특정되는 NNPF임을 나타낼 수 있다. 여기서, 마지막 NNPFC SEI 메시지는 디코딩 순서 상 현재 픽처의 첫 번째 VCL NAL 유닛에 선행하면서 기본 NNPF를 포함하는 NNPFC SEI 메시지의 반복에 해당하지 않을 수 있다.
- [289] `nnpfa_persistence_flag`는 현재 레이어에 대한 대상 NNPF의 지속성을 나타낼 수 있다. `nnpfa_persistence_flag`의 값 0은 대상 NNPF가 현재 픽처에 대한 후처리 필터링에만 사용될 수 있음을 나타낼 수 있다. `nnpfa_persistence_flag`의 값 1은 대상 NNPF가 다음 조건 중 하나 이상이 참일 때까지 출력 순서 상 현재 픽처와 현재 레이어 내 모든 후속 픽처들에 대한 후처리 필터링에 사용될 수 있음을 나타낼 수 있다.
- [290] - 현재 레이어의 새로운 CLVS가 시작됨

- [291] - 비트스트림이 종료됨
- [292] - 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 1과 동일한 `nnpfa_cancel_flag`를 갖는 NNPPFA SEI 메시지와 연관된 현재 레이어 내 픽처가 출력 순서에서 현재 픽처 다음에 출력됨
- [293] 대상 NNPF는 현재 SEI 메시지와 동일한 `nnpfa_target_id` 및 1과 동일한 `nnpfa_cancel_flag`를 갖는 NNPPFA SEI 메시지와 연관된 현재 레이어 내 후속 픽처에는 적용되지 않는다.
- [294] `nnpfcTargetPictures`는 디코딩 순서 상 현재 NNPPFA SEI 메시지에 선행하면서 `nnpfa_target_id`와 동일한 `nnpfc_id`를 가지는 마지막 NNPPFC SEI 메시지에 관련된 픽처들의 세트일 수 있다. `nnpfaTargetPictures`는 현재 NNPPFA SEI 메시지에 의해 대상 NNPF가 활성화되는 픽처들의 세트일 수 있다. `nnpfaTargetPictures`에 포함된 모든 임의의 픽처는 `nnpfcTargetPictures`에도 포함되어야 한다.
- [295] `nnpfa_num_output_entries`는 NNPPFA SEI 메시지 내 `nnpfa_output_flag[ i ]` 신택스 요소들의 개수를 나타낼 수 있다. `nnpfa_output_flag[ i ]`의 값은 0 내지 `NumInpPicsInOutTensor` 범위 내에 존재해야 한다.
- [296] `nnpfa_output_flag[ i ]`의 값 1은 인덱스 `InpIdx[ i ]`를 가지는 입력 픽처에 대응하는 NNPF 생성 픽처가 NNPPFA SEI 메시지에 의해 활성화된 NNPF 프로세스에 의해 출력됨을 나타낼 수 있다. 여기서, NNPF 프로세스는 NNPPFC SEI 메시지의 시멘틱스들에 의해 특정될 수 있다. `nnpfa_output_flag[ i ]`의 값 0은 인덱스 `InpIdx[ i ]`를 가지는 입력 픽처에 대응하는 NNPF 생성 픽처가 NNPPFA SEI 메시지에 의해 활성화된 NNPF 프로세스에 의해 출력되지 않음을 나타낼 수 있다. `nnpfa_num_output_entries`가 `NumInpPicsInOutTensor`보다 작은 경우, `nnpfa_output_flag[ i ]`은 `nnpfa_num_output_entries` 내지 `NumInpPicsInOutTensor - 1` 범위 내 i 의 각 값에 대해 1로 추론될 수 있다.
- [297] **포스트-필터 힌트(Post-filter hint)**
- [298] 포스트-필터 힌트에 대한 신택스 구조가 표 21에 나타나 있다.
- [299] [표21]

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

- [300] 표 21의 포스트-필터 힌트 선택스 구조는 SEI 메시지 형태로 시그널링될 수 있다. 표 21의 포스트-필터 힌트 선택스 구조를 시그널링하는 SEI 메시지는 포스트-필터 힌트 SEI 메시지라고 지칭될 수 있다.
- [301] 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.
- [302] 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.
- [303] 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.
- [304] - 현재 레이어의 새로운 CLVS가 시작됨
- [305] - 비트스트림이 종료됨
- [306] - 포스트-필터 힌트 SEI 메시지와 연관된 AU의 현재 레이어 내 픽처는 출력 순서에서 현재 픽처 다음에 출력됨
- [307] filter_hint_size_y는 필터 계수 또는 상관 어레이의 수직 크기를 나타낼 수 있다. filter_hint_size_y의 값은 1 내지 15의 범위에 존재해야 한다.
- [308] filter_hint_size_x는 필터 계수 또는 상관 어레이의 수평 크기를 나타낼 수 있다. filter_hint_size_x의 값은 1 내지 15의 범위에 존재해야 한다.
- [309] filter_hint_type은 표 22에 나타낸 바와 같이 전송된 필터 힌트의 유형을 나타낼 수 있다. filter_hint_type의 값은 0 내지 2의 범위에 존재해야 한다. 3과 동일한 filter_hint_type 값은 향후 사용을 위해 예약될 수 있으며, 비트스트림에는 존재하지 않는다. 디코더들은 filter_hint_type이 3인 포스트-필터 힌트 SEI 메시지를 무시해야 한다.
- [310] [표22]

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

- [311] filter_hint_chroma_coeff_present_flag의 값 1은 크로마에 대한 필터 계수들이 존재함을 나타낼 수 있다. filter_hint_chroma_coeff_present_flag의 값 0은 크로마에 대한 필터 계수들이 존재하지 않음을 나타낼 수 있다.

- [312] $filter_hint_value[cIdx][cy][cx]$ 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.
- [313] - $filter_hint_type$ 의 값이 0이면, $filter_hint_size_y * filter_hint_size_x$ 크기의 2차원 FIR(Finite Impulse Response) 필터의 계수들이 전송될 수 있다.
- [314] - 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 계수들을 나타낼 수 있다. 필터링 과정에서는 수평 필터가 먼저 적용되고 그 결과는 수직 필터에 의해 필터링될 수 있다.
- [315] - 그렇지 않으면($filter_hint_type$ 의 값이 2이면), 전송된 힌트는 원래 신호 s 와 디코딩된 신호 s' 사이의 상호 상관 행렬을 나타낼 수 있다.
- [316] $filter_hint_size_y * filter_hint_size_x$ 크기의 $cIdx$ 로 식별된 관련 색상 성분에 대한 정규화된 상호 상관 행렬은 수학적 식 10과 같이 정의될 수 있다.
- [317] [수식10]

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

- [318] 수학적 식 10에서, s 는 원본 픽처의 색상 성분 $cIdx$ 의 샘플 어레이를 나타내며, s' 는 이에 대응하는 디코딩된 픽처의 어레이를 나타내고, h 는 관련 색상 성분의 수직 높이를 나타내며, w 는 관련 색상 성분의 수평 너비를 나타내고, $bitDepth$ 는 색상 성분의 비트 뎀스를 나타낸다. 또한, $OffsetY$ 는 $(filter_hint_size_y \gg 1)$ 와 동일하고, $OffsetX$ 는 $(filter_hint_size_x \gg 1)$ 와 동일하고, cy 의 범위는 $0 \leq cy < filter_hint_size_y$ 이고, cx 의 범위는 $0 \leq cx < filter_hint_size_x$ 이다.
- [319] 디코더는 원본 신호와 디코딩된 신호의 cross-correlation matrix 및 디코딩된 신호의 auto-cross-correlation matrix로부터 Wiener 포스트-필터를 유도할 수 있다.
- [320] 종래 기술의 문제점
- [321] NNPF 관련 메시지(예를 들어, NNPF(Neural-network post-filter characteristics) SEI(Supplemental Enhancement Information) 및/혹은 NNPFA(Neural-network post-filter activation) SEI 등)에 대한 현재 설계에서는 활성화된(activated) NNPF에 복수의 입력 픽처가 이용될 수 있는데, 입력 픽처 중 하나 이상이 사용 가능하지 않을(unavailable) 수 있다(예를 들어, 입력 픽처가 비트스트림에 존재하지 않음 등).

예를 들어, 입력 픽처 중 하나 이상이 사용 가능하지 않은 경우란, 비트스트림의 첫 시작부의 픽처에서 NNPF가 활성화 되는 경우 등이 포함될 수 있다. 한편, 하나 이상의 입력 픽처가 사용 가능하지 않은 경우, 사용 가능한(available) 마지막 입력 픽처의 복제본(duplicate) 혹은 0 값 픽셀(zero value pel)을 갖는 픽처로, 사용 가능하지 않은(unavailable)(예를 들어, 손실(missing)된) 픽처를 대체할 것인지를 지시하는 플래그가 제공될 수 있다. 이 때, 해당 플래그는 NNPF SEI 메시지에서 시그널링되는 플래그로서, `nnpfc_absent_input_pic_zero_flag`일 수 있다. 또한, NNPF에 대한 각각의 입력 픽처에 대해 NNPF를 사용하는 필터링 프로세스가 입력 픽처와 연관된 출력 픽처를 출력하는지 혹은 아닌지에 대한 정보가 시그널링될 수 있다. 입력 픽처와 연관된 출력 픽처의 시그널링은 NNPF SEI 메시지에 포함될 수 있는데, 이러한 시그널링은 NNPF SEI 메시지와 연관된 NNPF SEI 메시지에서의 시그널링으로 업데이트될 수 있으며, 일부 예에서는 픽처의 출력이 취소(CANCEL)될 수 있다.

- [322] 한편, 이 경우, 사용 가능하지 않은 입력 픽처와 연관된 출력 픽처와 관련하여 문제가 발생할 수 있다. 사용 가능하지 않은 입력 픽처와 연관된 출력 픽처가 존재하지 않는 것이 바람직할 수 있으나, NNPF 및 NNPF SEI 메시지에 대한 현재 설계에서는 이를 방지할 수 있는 방법이 명확하지 않다.
- [323] 실시예 개요
- [324] 본 개시에서는 상기를 포함한 종래 설계의 문제점을 해결할 수 있는 다양한 실시예를 제안한다. 하기의 실시예는 독립적으로 이용되거나, 서로 혹은 다른 실시예와 조합되어 이용될 수 있으며, 이 또한 본 개시에 포함된다고 할 것이다.
- [325] 1. 일 실시예로서, NNPF SEI 메시지는 복수의 입력 픽처를 입력으로 할 수 있는 타겟 NNPF(즉, 대상 NNPF)에 대한 NNPF SEI 메시지를 활성화할 수 있는데, 하나 이상의 입력 픽처가 사용 가능하지 않으며(예를 들어, 비트스트림에 존재하지 않음 등), 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, 입력 픽처의 `nnpfc_input_pic_output_flag[i]`의 값이 1), 관련 출력 픽처를 갖지만 사용 가능하지 않은 입력 픽처 각각에 대해, `nnpfa_output_flag[i]`가 존재해야 하며, 해당 신택스의 값은 특정 값(예를 들어, 0)이어야 할 수 있다.
- [326] 2. 혹은, NNPF SEI 메시지는 복수의 입력 픽처를 입력으로 할 수 있는 타겟 NNPF(즉, 대상 NNPF)에 대한 NNPF SEI 메시지를 활성화할 수 있는데, 하나 이상의 입력 픽처가 사용 가능하지 않으며(예를 들어, 비트스트림에 존재하지 않음 등), 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, 입력 픽처의 `nnpfc_input_pic_output_flag[i]`의 값이 1 등), 다음이 적용될 수 있다:
- [327] o 특정 신택스(예를 들어, `nnpfa_num_output_entries` 등)의 값은 특정 값(예를 들어, 0)보다 커야 할 수 있다.

- [328] - 혹은, 특정 신택스(예를 들어, `nnpfa_num_output_entries` 등)의 값은 특정 범위 내의 값으로 제한될 수 있다. 예를 들어, 특정 신택스의 값은 특정 값(예를 들어, $X+1$)보다 작지 않아야 하며, 여기서 X 는 관련 출력 픽처를 갖는 마지막 입력 픽처의 인덱스(예를 들어, 현재 픽처에서 가장 먼 픽처) 값일 수 있다.
- [329] o 사용 가능하지 않지만 NNPF에서 관련 출력 영상을 갖도록 시그널링된 입력 영상에 대한 특정 신택스(예를 들어, `nnpfa_output_flag[i]` 등)의 값은 특정 값으로 제한될 수 있다(예를 들어, 0).
- [330] 3. 혹은, NNPF SEI 메시지가 복수의 입력 픽처를 입력으로 할 수 있는 타겟 NNPF(즉, 대상 NNPF)에 대한 NNPF SEI 메시지를 활성화하고, 하나 이상의 입력 픽처가 사용 가능하지 않으며(예를 들어, 비트스트림에 존재하지 않음 등), 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, 입력 픽처의 `nnpfa_input_pic_output_flag[i]`의 값이 1 등), NNPF 기반의 필터링 프로세스는 이러한 입력 픽처와 관련된 출력 픽처를 출력하지 않도록 제한될 수 있다.
- [331] 하기에서는 상기의 실시예를 포함한 다양한 실시예에 대하여 보다 상세하게 설명하며, 코딩된 비디오 비트스트림의 NNPF(neural-network post-filter) 관련 SEI 메시지에서의 입력 픽처 및 출력 픽처에 대한 개선을 제시한다. 하기에서 설명되는 실시예들은 표준 비디오 코덱(예를 들어, VVC(Versatile Video Coding) 등) 및 VSEI(Versatile Supplemental Enhancement Information Messages for Coded Video Bitstreams)를 기반으로 작성되었지만 다른 비디오 코딩 기술에도 적용될 수 있음은 자명하여, 이 또한 본 개시의 내용에 포함된다 할 것이다.
- [332] 한편, 하기에서 실시예를 설명함에 있어서 NNPF(Neural-network post-filter) SEI 메시지 혹은 NNPF 관련 SEI 메시지란 NNPF(Neural-network post-filter Characteristic) SEI 메시지 및/혹은 NNPF(Neural-network post-filter activation) SEI 메시지 등을 포함할 수 있다.
- [333] 한편, 하기에서 실시예를 설명함에 있어서 사용되는 신택스의 명칭은 설명의 명료함을 위해 임의로 지정한 것이므로, 신택스의 명칭이 변경될 수 있음은 자명하며, 신택스의 명칭이 변경되더라도 본 개시에 포함된다 할 것이다.
- [334] 이하, 도면을 참조하여 본원의 실시예에 대하여 상세하게 설명한다.
- [335] **실시예 1**
- [336] 실시예 1에서는 상기 실시예 개요에서 설명한 목차 1의 실시예에 대하여 상세히 설명한다. 하기에서는 VSEI 메시지 신택스 및 시맨틱스에 대하여 설명한다.
- [337] 일 예로서, NNPF(Neural-network post-filter) 관련 SEI 메시지(예를 들어, NNPF SEI 메시지 등)의 신택스 및 시맨틱스는 아래와 같이 수정될 수 있다. NNPF SEI 메시지 신택스는 다음 표와 같이 시그널링될 수 있다.
- [338] [표23]

nn_post_filter_activation(payloadSize) {	Descriptor
nnpfa_target_id	ue(v)
nnpfa_cancel_flag	u(1)
if(!nnpfa_cancel_flag) {	
nnpfa_target_base_flag	u(1)
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)
}	
}	

- [339] 일 예로서, 표 23의 실시예에 개시된 타겟 id와 관련된 정보인 **nnpfa_target_id**, NNPF 지속성 취소에 관련된 정보인 **nnpfa_cancel_flag**, 기본 NNPF에 관련된 정보인 **nnpfa_target_base_flag**, NNPF 지속성에 관련된 정보인 **nnpfa_persistence_flag**, NNPF 생성 픽처, 즉 NNPF 기반의 입력 픽처에 대응되는 출력 픽처의 출력과 관련된 정보인 **nnpfa_output_flag[i]**와, **nnpfa_output_flag[i]**의 개수와 관련된 정보인 **nnpfa_num_output_entries**에 대하여는 상기에서 설명한 바와 같으므로, 중복 설명은 생략한다. 한편, 입력 픽처에 대응되는 출력 픽처(예를 들어, NNPF 생성 픽처 등)의 출력에 관한 정보(예를 들어, 출력 픽처 출력 정보 등)에 포함될 수 있는 **nnpfa_output_flag[i]**는, 입력 픽처 별로 존재할 수 있으나, 타겟 NNPF(대상 NNPF)가 복수의 입력 픽처를 필요로 하고 하나 이상의 입력 픽처가 비트스트림에 존재하지 않으나(예를 들어, 사용 불가능함 등) 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, NNPF SEI 메시지의 관련 **nnpfc_input_pic_output_flag[i]**의 값이 1임 등), 해당 선택스의 값은 특정 값으로 제한될 수 있다. 예를 들어, **nnpfa_output_flag[i]**는 관련 출력 픽처를 갖는 사용 가능하지 않은 입력 픽처 각각에 대한 **nnpfa_output_flag[i]**가 존재해야 하나, 해당 선택스의 값은 0으로 제한될 수 있다.

[340] **실시예 2**

- [341] 실시예 2에서는 상기 실시예 개요에서 설명한 목차 2의 실시예에 대하여 상세히 설명한다. 하기에서는 VSEI 메시지 선택스 및 시맨틱스에 대하여 설명한다.

- [342] 일 예로서, 표 23에 따라 VSEI 메시지에 **nnpfa_target_id** 등의 선택스가 포함되어 시그널링될 수 있으며, **nnpfa_output_flag[i]**가 SEI 메시지에 포함될 수 있다. 한편, 입력 픽처에 대응되는 출력 픽처(예를 들어, NNPF 생성 픽처 등)의 출력에 관한 정보(예를 들어, 출력 픽처 출력 정보 등)에 포함될 수 있는 **nnpfa_output_flag[i]**는, 입력 픽처 별로 존재할 수 있으나, 타겟 NNPF(대상 NNPF)가 복수의 입력 픽처를 필요로 하고 하나 이상의 입력 픽처가 비트스트림에 존재하지 않으나(예를 들어, 사용 불가능함 등) 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, NNPF SEI 메시지의 관련 **nnpfc_input_pic_output_flag[i]**의 값이 1임 등), **nnpfa_output_flag[i]** 및/혹은 **nnpfa_num_output_entries**에 대한 제약이 존재할 수 있다. 예를 들어,

`nnpfa_output_flag[i]`의 개수와 관련된 정보인 `nnpfa_num_output_entries`의 값이 제한될 수 있다. 일 예로서, `nnpfa_num_output_entries` 선택스의 값은 특정 값(예를 들어, $X+1$)보다 작은 값이어야 한다는 제약이 적용될 수 있으며, X 는 관련 출력 픽처가 존재하는 마지막 입력 픽처의 인덱스(예를 들어, 현재 픽처에서 가장 먼 픽처의 인덱스)일 수 있다. 또한, 사용 가능하지 않지만 NNPF에 관련 출력 픽처가 포함되어 시그널링되는 입력 픽처의 `nnpfa_output_flag`의 값은 특정 값(예를 들어, 0)으로 정해질 수 있다.

[343] **실시예 3**

[344] 실시예 3에서는 상기 실시예 개요에서 설명한 목차 3의 실시예에 대하여 상세히 설명한다. 하기에서는 VSEI 메시지 선택스 및 시맨틱스에 대하여 설명한다.

[345] 일 예로서, NNPF에 입력되는 픽처의 수에 관련된 정보인 `nnpfc_num_input_pics_minus1`, 출력 픽처 생성 여부와 관련된 정보(예를 들어, 출력 픽처 존재 정보 등)인 `nnpfc_input_pic_output_flag[i]`, 비트스트림에 존재하지 않는 입력 픽처의 샘플 값의 표현 방법에 관련된 정보인 `nnpfc_absent_input_pic_zero_flag` 등의 정보가 NNPF 관련 메시지에 포함되어 시그널링될 수 있다.

[346] 한편, NNPF SEI 메시지가 복수의 입력 픽처를 입력으로 하는 타겟 NNPF에 대한 NNPF SEI 메시지에 의해 활성화되고, 하나 이상의 입력 픽처가 사용 가능하지 않으며(예를 들어, 비트스트림에 존재하지 않음 등), 사용 가능하지 않은 입력 픽처에 대한 관련 출력 픽처가 존재하는 경우(예를 들어, 입력 픽처의 `nnpfc_input_pic_output_flag[i]`의 값이 1임 등), 필터링 프로세스는 이러한 입력 픽처와 관련된 출력 픽처를 출력하지 않아야 한다. 일 예로서, 이 경우, 필터링 프로세스는 NNPF 기반의 필터링 프로세스일 수 있으며, 출력 픽처가 존재하지 않게, 즉 출력 픽처가 생성되지 않을 수 있다.

[347] 본 개시에서 설명한 상기 실시예에 의하면, 상기에서 설명한 종래 기술의 문제점을 해결할 수 있을 뿐만 아니라 정보의 선택스 및/혹은 시맨틱스를 다각화하거나 정보에 대한 제약 조건을 명확히 하거나 정보의 시맨틱스를 명확히 함으로써 디코더의 오차를 줄이고 코딩 품질과 효율을 향상시킬 수 있다.

[348] **영상 복호화 방법의 실시예**

[349] 이하에서는, 본원의 다양한 실시예들에 따른 영상 부호화 방법과 영상 복호화 방법을 설명하도록 한다. 도 5의 영상 복호화 방법은 영상 복호화 장치(200)에 의해 수행될 수 있으며, 도 6의 영상 부호화 방법은 영상 부호화 장치(100)에 의해 수행될 수 있다. 또한, 도 5 및 도 6의 영상 복호화 및 부호화 방법은 각각 상기에서 설명한 실시예(실시예 1-3을 포함)를 기반으로 할 수 있다.

[350] 도 5는 본 개시의 일 실시예에 따른 영상 복호화 장치에 의해 수행될 수 있는 영상 복호화 방법을 설명하기 위한 도면이다.

[351] 먼저, 일 예로서, 입력 픽처에 대한 (대응) 출력 픽처 정보가 NNPF(neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지

(예를 들어, NNPF(neural-network post-filter characteristics) 메시지 및/혹은 NNPFA(neural-network post-filter activation) 메시지 등)가 획득(S510)될 수 있다. 포스트 필터 기반 (대응) 출력 픽처 정보란 NNPF 관련 정보를 포함할 수 있다. NNPF 관련 정보는 상기에서 표 등을 참조하여 설명한 정보를 포함할 수 있다.

- [352] 이후, 획득된 (대응) 출력 픽처 정보에 기반하여, 입력 픽처에 대한 (대응) 출력 픽처가 획득(S520)될 수 있다. 이 경우, 입력 픽처에 대한 출력 픽처가 존재하지 않거나(생성되지 않거나) 입력 픽처에 대한 출력 픽처가 존재하나 출력 픽처가 출력되지 않는 경우 출력 픽처는 획득되지 않을 수 있다. 한편, 입력 픽처에 대한 출력 픽처가 존재하고 입력 픽처에 대한 출력 픽처가 출력되는 것으로 결정되는 경우 출력 픽처가 획득될 수 있다. 여기서, 출력 픽처 정보는 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함할 수 있다. 일 예로서, 출력 픽처 출력 정보는 다른 정보에 기반하여 해당 값이 결정될 수 있으며, 해당 정보의 값을 결정하기 위한 제약 조건이 존재할 수 있다. 일 예로서, 출력 픽처 출력 정보는, 예를 들어, `nnpfa_output_flag[ i ]`를 포함할 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처의 사용 가능 여부(예를 들어, 입력 픽처의 비트스트림 내 존재 여부)에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처의 사용 가능 여부는 입력 픽처가 사용 가능한지 여부를 나타내는 정보인 입력 픽처 존재 정보에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처 존재 정보는 `inputPresentFlag`를 포함할 수 있으며, 입력 픽처 존재 정보는 다른 정보나 다른 인덱스에 기반하여 유도될 수 있다. 예를 들어, 입력 픽처가 사용 가능 하지 않은 경우란 입력 픽처가 비트스트림에 존재하지 않는 경우를 포함할 수 있다. 또한, 일 예로서, 입력 픽처가 사용 가능하지 않은 경우(예를 들어, 입력 픽처 존재 정보가 입력 픽처가 사용 가능하지 않음을 나타내는 경우) 출력 픽처 출력 정보는 해당 입력 픽처에 대한 출력 픽처가 출력되지 않음을 나타낼 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 항상 특정 값(예를 들어, 0)보다 큰 값으로 결정될 수 있다. 이와 같은 경우를 포함하여, 출력 픽처 출력 정보의 개수에 대하여 일정 제약 조건이 존재할 수 있다. 예를 들어, 출력 픽처 출력 정보의 개수를 나타낼 수 있는 정보인 `nnpfa_num_output_entries`의 값은 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 특정 범위는 특정 입력 픽처의 인덱스 값에 기반하여 결정될 수 있는데, 일 예로서, 특정 입력 픽처는 특정 픽처에서 가장 먼 픽처, 즉 마지막 입력 픽처일 수 있다. 한편, 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처에 대한 출력 픽처의 존재 여부(즉, 출력 픽처의 생성 여부)를 나타내는 정보인 출력 픽처 존재 정보(즉, 출력 픽처 생성 정보)에 기반하여 결정될 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 출력 픽처 존재 여부 및/혹은 입력 픽처 존재 여부에 기반하여 결정될 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 출력 픽처 존재 정보 및/혹은 입력 픽처 존재 정보에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처가 사용 가능 하지 않고(예를 들어, 입력 픽처가 비트스트림에

존재하지 않음 등) 그럼에도 (입력 픽처에 대한) 출력 픽처가 존재하는 경우 출력 픽처는 출력되지 않을 수 있다. 보다 상세하게는, 입력 픽처 존재 정보가 입력 픽처가 사용 가능하지 않음(예를 들어, 입력 픽처가 비트스트림에 존재하지 않음 등)을 나타내고 (입력 픽처에 대한) 출력 픽처 존재 정보가 출력 픽처가 존재함을 나타냄(예를 들어, `nnpfc_input_pic_output_flag[i]`의 값이 1임 등)에 기반하여 출력 픽처 출력 정보는 출력 픽처가 출력되지 않음을 나타낼 수 있다. 일 예로서, 출력 픽처가 출력되지 않음은 출력 픽처 출력 정보의 값이 특정 값(예를 들어, 0)인 것으로 표현될 수 있다. 이 경우, 일 예로서, NNPF에 기반한 필터링 프로세스에서 해당 출력 픽처의 출력이 제외될 수 있다. 한편, 일 예로서, 출력 픽처 존재 정보는 NNPF SEI 메시지로 시그널링될 수 있으며, 출력 픽처 출력 정보는 NNPFA SEI 메시지로 시그널링될 수 있다.

[353] 이후, 도면에는 표현되지 않았으나 (대응) 출력 픽처 정보를 기반으로 픽처가 복원될 수도 있다.

[354] 한편, 도 5의 영상 복호화 방법은 본 개시의 일 실시예에 해당하므로, 일정 단계가 변경되거나 단계의 순서가 변경되거나 일부 단계가 추가 혹은 삭제될 수도 있으며, 이러한 수정이 있는 경우도 본 개시에 포함됨은 자명하다 할 것이다.

[355] 영상 부호화 방법의 실시예

[356] 도 6은 본 개시의 일 실시예에 따른 영상 부호화 장치에 의해 수행될 수 있는 영상 부호화 방법을 설명하기 위한 도면이다.

[357] 먼저 입력 픽처에 대한 포스트 필터 기반 출력 픽처 정보가 결정(S610)될 수 있다. 포스트 필터 기반 출력 픽처 정보란 NNPF 관련 정보를 포함할 수 있으며, NNPF 관련 정보는 상기에서 표 등을 참조하여 설명한 바와 동일할 수 있다.

[358] 이후, 출력 픽처 정보는 NNPF(neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지로 시그널링(S620)될 수 있다. 상기에서 설명한 바와 같이, NNPF 관련 SEI 메시지는 NNPF SEI 메시지 및/혹은 NNPFA SEI 메시지 등을 포함할 수 있다. 한편, 입력 픽처에 대한 출력 픽처가 존재하지 않거나 (생성되지 않았거나) 출력 픽처가 존재하나(생성되었으나) 출력 픽처가 출력되지 않는 것도 가능할 수 있다. 한편, 입력 픽처에 대한 출력 픽처가 존재하고, 출력 픽처가 출력되는 것으로 결정된 경우에는 출력 픽처가 그대로 출력될 수도 있다. 이와 관련하여, 출력 픽처 존재 정보, 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보, 및/혹은 입력 픽처 존재 정보 등이 출력 픽처 정보에 포함되어 시그널링될 수도 있으나, 다른 신택스에 의해 특정 값으로 유도될 수도 있다. 일 예로서, 출력 픽처 출력 정보의 값을 결정함에 있어서, 특정 제약 조건이 있을 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처가 사용 가능한지 여부(즉, 비트스트림에 입력 픽처의 존재 여부 혹은 부호화될지 여부)에 기반하여 결정될 수 있다. 보다 상세하게는, 출력 픽처 출력 정보의 값은, 예를 들어, 입력 픽처가 비트스트림에 존재하지 않도록 결정됨(즉, 디코더 측에서 사용 가능하지 않게 됨)에 기반하여 특정 값(예를 들어, 0)으로 결정

될 수 있다. 또한, 일 예로서, 입력 픽처의 비트스트림 내 존재 여부는 다른 정보에 기반하여 결정될 수도 있다. 한편, 입력 픽처가 비트스트림에 존재하지 않도록 결정됨에 기반하여 생성된 출력 픽처(입력 픽처에 대응되는 결과 픽처)가 출력되지 않을 수 있는데, 이를 출력 픽처 출력 정보가 나타낼 수 있다. 일 예로서, 출력 픽처 출력 정보는, 예를 들어, `nnpfa_output_flag[ i ]`를 포함할 수 있다. 예를 들어, 입력 픽처가 사용 불가능한 경우란 입력 픽처가 비트스트림에 존재하지 않게 되는 경우를 포함할 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 항상 특정 값(예를 들어, 0)보다 큰 값으로 결정될 수 있다. 이와 같은 경우를 포함하여, 출력 픽처 출력 정보의 개수에 대하여 일정 제약 조건이 존재할 수 있다. 예를 들어, 출력 픽처 출력 정보의 개수를 나타낼 수 있는 정보인 `nnpfa_num_output_entries`의 값은 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 특정 범위는 특정 입력 픽처의 인덱스 값에 기반하여 결정될 수 있는데, 일 예로서, 특정 입력 픽처는 특정 픽처에서 가장 먼 픽처, 즉 마지막 입력 픽처일 수 있다. 한편, 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처에 대한 출력 픽처의 존재 여부(즉, 출력 픽처의 생성 여부)에 기반하여 결정될 수 있다. 보다 상세하게는, 출력 픽처 출력 정보의 값은 입력 픽처에 대한 출력 픽처가 존재함에 기반하여 특정 값으로 결정될 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 출력 픽처가 존재하는지 여부 및/혹은 입력 픽처가 존재하는지 여부에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처가 비트스트림에 존재할지는 입력 픽처 존재 여부에 대한 정보로 표현될 수 있으며, `inputPresentFlag`로 표현될 수 있다. 일 예로서, 이 경우, 입력 픽처에 대한 출력 픽처가 존재하지 않거나(생성되지 않거나) 입력 픽처에 대한 출력 픽처가 존재하나 출력 픽처가 출력되지 않는 경우 출력 픽처는 획득되지 않을 수 있다. 한편, 입력 픽처에 대한 출력 픽처가 존재하고 입력 픽처에 대한 출력 픽처가 출력되는 것으로 결정되는 경우 출력 픽처가 획득될 수 있다. 여기서, 출력 픽처 정보는 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함할 수 있다. 일 예로서, 출력 픽처 출력 정보는 다른 정보에 기반하여 해당 값이 결정될 수 있으며, 해당 정보의 값을 결정하기 위한 제약 조건이 존재할 수 있다. 일 예로서, 출력 픽처 출력 정보는, 예를 들어, `nnpfa_output_flag[ i ]`를 포함할 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처의 사용 가능 여부(예를 들어, 입력 픽처의 비트스트림 내 존재 여부)에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처의 사용 가능 여부는 입력 픽처가 사용 가능한지 여부를 나타내는 정보인 입력 픽처 존재 정보에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처 존재 정보는 `inputPresentFlag`를 포함할 수 있으며, 입력 픽처 존재 정보는 다른 정보나 다른 신호에 기반하여 유도될 수 있다. 예를 들어, 입력 픽처가 사용 가능하지 않은 경우란 입력 픽처가 비트스트림에 존재하지 않는 경우를 포함할 수 있다. 또한, 일 예로서, 입력 픽처가 사용 가능하지 않은 경우(예를 들어, 입력 픽처 존재 정

보가 입력 픽처가 사용 가능하지 않음, 즉 비트스트림에 존재하지 않음 등을 나타내게 되는 경우) 출력 픽처 출력 정보는 해당 입력 픽처에 대한 출력 픽처가 출력되지 않음을 나타내도록 결정될 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 출력 픽처 출력 정보의 개수는 항상 특정 값(예를 들어, 0)보다 큰 값으로 결정될 수 있다. 이와 같은 경우를 포함하여, 출력 픽처 출력 정보의 개수에 대하여 일정 제약 조건이 존재할 수 있다. 예를 들어, 출력 픽처 출력 정보의 개수를 나타낼 수 있는 정보인 `nnpfa_num_output_entries`의 값은 특정 범위 내의 값으로 결정될 수 있다. 또한, 일 예로서, 특정 범위는 특정 입력 픽처의 인덱스 값에 기반하여 결정될 수 있는데, 일 예로서, 특정 입력 픽처는 특정 픽처에서 가장 먼 픽처, 즉 마지막 입력 픽처일 수 있다. 한편, 일 예로서, 출력 픽처 출력 정보의 값은 입력 픽처에 대한 출력 픽처의 존재 여부(즉, 출력 픽처의 생성 여부)에 기반하여 결정될 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 출력 픽처 존재 여부 및/혹은 입력 픽처 존재 여부에 기반하여 결정될 수 있다. 일 예로서, 출력 픽처 출력 정보의 값은 출력 픽처 존재 정보(즉, 출력 픽처 생성 정보)가 어떤 값을 나타내도록 결정되었는지 및/혹은 입력 픽처 존재 정보가 어떤 값을 나타내도록 결정되었는지에 기반하여 결정될 수 있다. 일 예로서, 입력 픽처가 사용 가능하지 않고(예를 들어, 입력 픽처가 비트스트림에 존재하지 않음 등) 그럼에도 (입력 픽처에 대한) 출력 픽처가 존재하는 경우 출력 픽처는 출력되지 않을 수 있다. 보다 상세하게는, 입력 픽처 존재 정보가 입력 픽처가 사용 가능하지 않음을 나타내도록 결정되고(예를 들어, 입력 픽처가 비트스트림에 존재하지 않도록 결정됨 혹은 입력 픽처가 비트스트림에 부호화되지 않음 등), (입력 픽처에 대한) 출력 픽처 존재 정보가 출력 픽처가 존재함을 나타내도록 결정됨(예를 들어, `nnpfc_input_pic_output_flag[i]`의 값이 1이도록 결정됨 등)에 기반하여 출력 픽처 출력 정보는 출력 픽처가 출력되지 않음을 나타낼 수 있다. 일 예로서, 출력 픽처가 출력되지 않음은 출력 픽처 출력 정보의 값이 특정 값(예를 들어, 0)인 것으로 표현될 수 있다. 이 경우, 일 예로서, NNPF에 기반한 필터링 프로세스에서 해당 출력 픽처의 출력이 제외될 수 있다.

- [359] 한편, 일 예로서, 출력 픽처 존재 정보는 NNPF SEI 메시지로 시그널링될 수 있으며, 출력 픽처 출력 정보는 NNPFA SEI 메시지로 시그널링될 수 있다.
- [360] 이후, 도면에는 표현되지 않았으나 출력 픽처 정보를 기반으로 픽처가 복원될 수도 있다.
- [361] 또한, 일 예로서, 영상 부호화 방법에 의해 생성된 비트스트림을 기록한 컴퓨터 판독 가능한 매체가 제공될 수 있으며, 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법이 제공될 수도 있다.
- [362] 한편, 도 6의 영상 부호화 방법은 본 개시의 일 실시예에 해당하므로, 일정 단계가 변경되거나 단계의 순서가 변경되거나 일부 단계가 추가 혹은 삭제될 수도 있으며, 이러한 수정이 있는 경우도 본 개시에 포함됨은 자명하다 할 것이다.

- [363] 본원에 의하면, VSEI에 포함될 수 있는 정보의 의미를 명확히 할 수 있어 디코더의 오차를 줄이고 보다 정확한 시나리오를 표현할 수 있으며, 이에 따라 코딩 품질을 향상시킬 수 있다. 뿐만 아니라, 본원에 의하면, 대응 출력 픽처가 생성되었으나 입력 픽처가 존재하지 않는 경우에 대한 명확한 처리가 가능하므로, 코딩 효율을 향상시킬 수 있다.
- [364] 도 7은 본 개시에 따른 실시예가 적용될 수 있는 콘텐츠 스트리밍 시스템을 예시적으로 나타낸 도면이다.
- [365] 도 7에 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.
- [366] 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.
- [367] 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.
- [368] 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.
- [369] 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.
- [370] 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, 위치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)), 디지털 TV, 데스크탑 컴퓨터, 디지털 사이니지 등이 있을 수 있다.

- [371] 상기 콘텐츠 스트리밍 시스템 내 각 서버들은 분산 서버로 운영될 수 있으며, 이 경우 각 서버에서 수신하는 데이터는 분산 처리될 수 있다.
- [372] 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-판독 가능 매체(non-transitory computer-readable medium)를 포함한다.

산업상 이용가능성

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

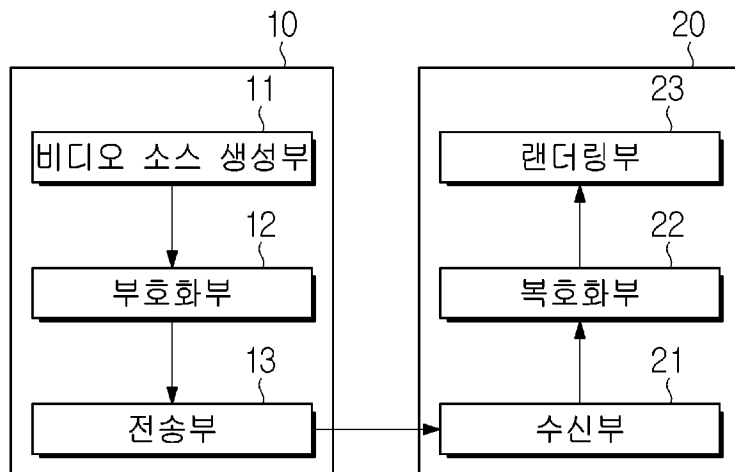
청구범위

- [청구항 1] 영상 복호화 장치에 의해 수행되는 영상 복호화 방법으로서,
입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 NNPF (neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지에서로부터 획득하는 단계; 및
상기 출력 픽처 정보에 기반하여, 상기 입력 픽처에 대한 출력 픽처를 획득하는 단계;를 포함하되,
상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함하는, 영상 복호화 방법.
- [청구항 2] 제1 항에 있어서,
상기 출력 픽처 출력 정보의 값은 상기 입력 픽처의 존재 여부에 기반하여 결정되는, 영상 복호화 방법.
- [청구항 3] 제2 항에 있어서,
상기 출력 픽처 출력 정보의 값은 상기 입력 픽처에 대한 상기 출력 픽처의 존재 여부를 나타내는 정보인 출력 픽처 존재 정보에 더 기반하여 결정되는, 영상 복호화 방법.
- [청구항 4] 제3 항에 있어서,
상기 입력 픽처가 존재하지 않고 상기 출력 픽처 존재 정보가 상기 출력 픽처가 존재함을 나타냄에 기반하여 상기 출력 픽처 출력 정보는 상기 출력 픽처가 출력되지 않음을 나타내는, 영상 복호화 방법.
- [청구항 5] 제4 항에 있어서,
상기 출력 픽처 출력 정보의 개수는 특정 범위 내의 값으로 결정되는, 영상 복호화 방법.
- [청구항 6] 제5 항에 있어서,
상기 특정 범위는 특정 입력 픽처의 인덱스 값에 기반하여 결정되는, 영상 복호화 방법.
- [청구항 7] 제4 항에 있어서,
상기 출력 픽처 출력 정보의 개수는 특정 값보다 항상 큰 값으로 결정되는, 영상 복호화 방법.
- [청구항 8] 제4 항에 있어서,
상기 NNPF에 기반한 필터링 프로세스에서 상기 존재하지 않는 입력 픽처에 대한 상기 출력 픽처의 출력이 제외되는, 영상 복호화 방법.
- [청구항 9] 제3 항에 있어서,
상기 출력 픽처 존재 정보는 NNPF SEI 메시지에서 획득되는, 영상 복호화 방법.
- [청구항 10] 제1 항에 있어서,

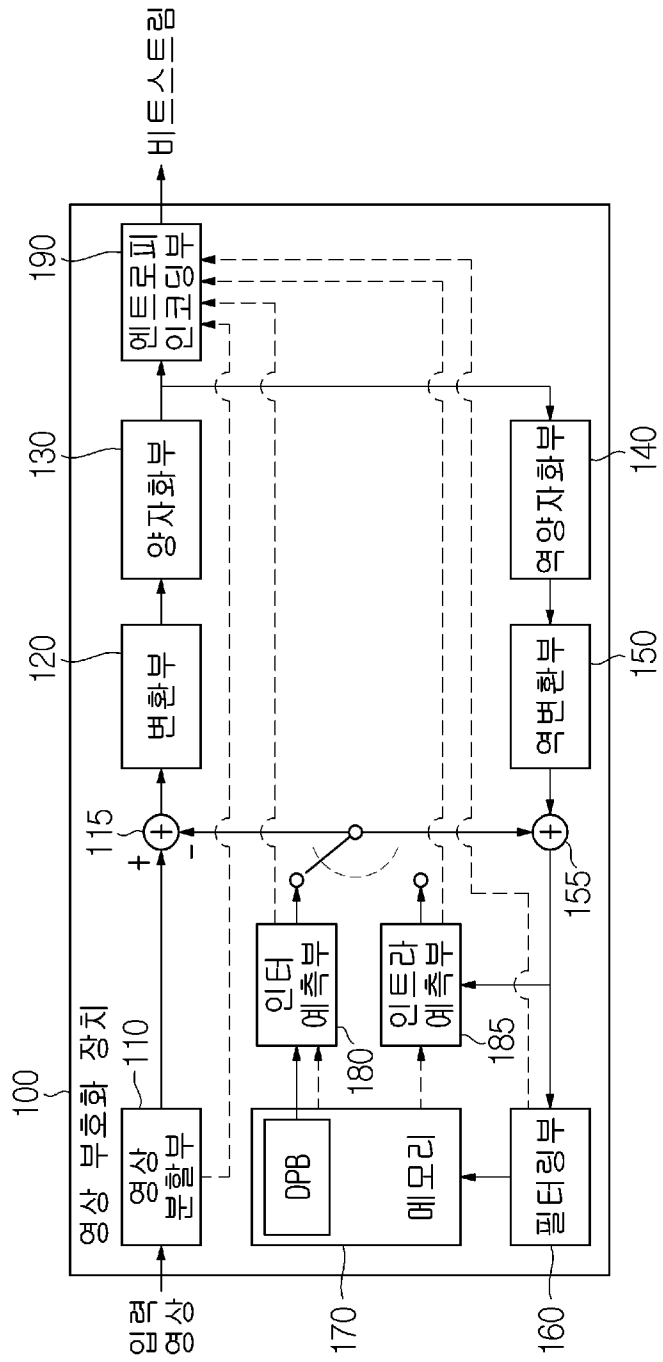
상기 출력 픽처 출력 정보는 NNPF SEI 메시지에서 획득되는, 영상 복호화 방법.

- [청구항 11] 영상 부호화 장치에 의해 수행되는 영상 부호화 방법으로서, 입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 결정하는 단계; 및
상기 출력 픽처 정보를 NNPF(neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지로 시그널링하는 단계;를 포함하되,
상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함하는, 영상 부호화 방법.
- [청구항 12] 제11 항의 영상 부호화 방법에 의해 생성된 비트스트림을 기록한 컴퓨터 판독 가능한 매체.
- [청구항 13] 영상 부호화 방법에 의해 생성된 비트스트림을 전송하는 방법에 있어서, 상기 영상 부호화 방법은,
입력 픽처에 대한 포스트 필터 기반의 출력 픽처 정보를 결정하는 단계; 및
상기 출력 픽처 정보를 NNPF(neural-network post-filter) 관련 SEI(supplemental enhancement information) 메시지로 시그널링하는 단계;를 포함하되,
상기 출력 픽처 정보는 상기 입력 픽처에 대한 출력 픽처가 출력되는지 여부를 나타내는 정보인 출력 픽처 출력 정보를 포함하는, 방법.

[도1]



[도2]



비트스트림

영상 부호화 장치

200

210

엔트로피 디코딩부

220

역양자화부

230

역변환부

235

+

240

필터링부

250

메모리

DPB

260

인터 예측부

265

인트라 예측부

영상 스트림

Diagram illustrating the transformation of a 5x5 grid into a 4x4 grid of 2x2 blocks.

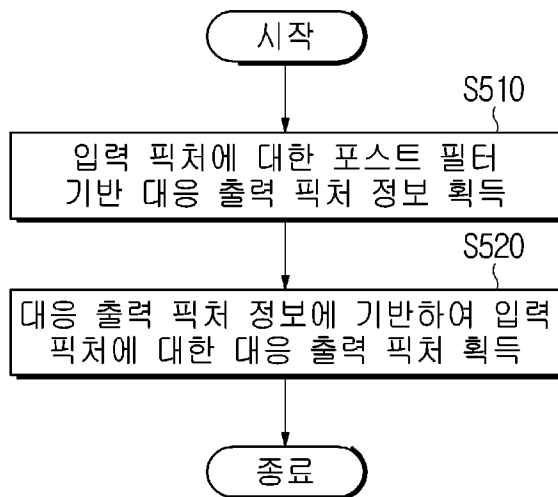
Original 5x5 Grid:

0	1	4	5	...
2	3	6	7	...
8	9	12	13	...
10	11	14	15	...
...	...	...	...	...

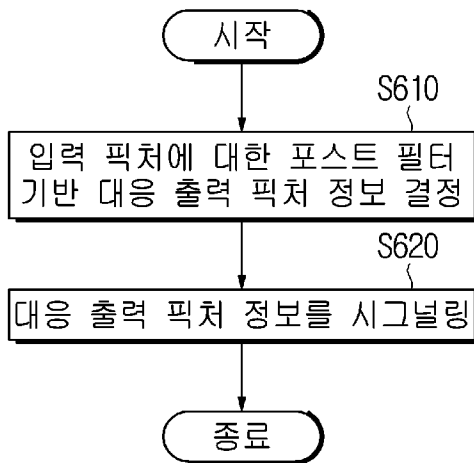
Transformed 4x4 Grid (each cell is a 2x2 block):

<table border="1"> <tr><td>0</td><td>4</td><td>...</td></tr> <tr><td>8</td><td>12</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table>	0	4	...	8	12	...	...	...	...	<table border="1"> <tr><td>1</td><td>5</td><td>...</td></tr> <tr><td>9</td><td>13</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table>	1	5	...	9	13	...	...	...	...
0	4	...																	
8	12	...																	
...	...	...																	
1	5	...																	
9	13	...																	
...	...	...																	
<table border="1"> <tr><td>2</td><td>6</td><td>...</td></tr> <tr><td>10</td><td>14</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table>	2	6	...	10	14	...	...	...	...	<table border="1"> <tr><td>3</td><td>7</td><td>...</td></tr> <tr><td>11</td><td>16</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table>	3	7	...	11	16	...	...	...	...
2	6	...																	
10	14	...																	
...	...	...																	
3	7	...																	
11	16	...																	
...	...	...																	

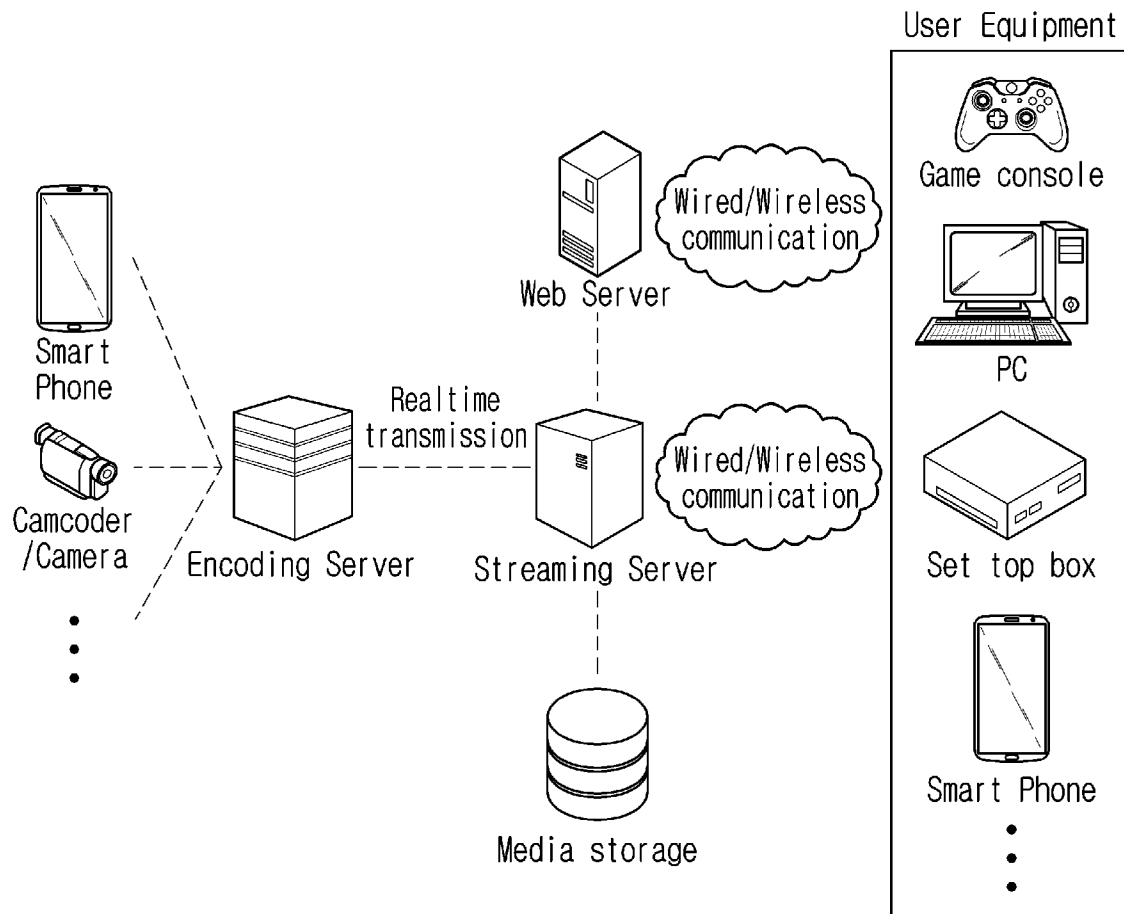
[도5]



[도6]



[도7]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2024/008958

A. CLASSIFICATION OF SUBJECT MATTER H04N 19/85(2014.01)i; H04N 19/117(2014.01)i; H04N 19/70(2014.01)i; H04N 19/172(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 19/85(2014.01); G06T 5/00(2006.01); G06T 9/00(2006.01); H04N 13/00(2006.01); H04N 19/126(2014.01); H04N 19/176(2014.01); H04N 19/46(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: 포스트 필터(post-filter), 출력 픽처 정보(output picture information), NNPF SEI messages(neural-network post-filter supplemental enhancement information messages)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-1776448 B1 (QUALCOMM INCORPORATED) 07 September 2017 (2017-09-07) See claims 1-3.	1-13
A	KR 10-2023-0010259 A (TENCENT AMERICA LLC) 18 January 2023 (2023-01-18) See claims 1-2.	1-13
A	JP 2017-152905 A (OKI ELECTRIC IND CO., LTD.) 31 August 2017 (2017-08-31) See paragraphs [0043]-[0046].	1-13
A	KR 10-2022-0137552 A (HYUNDAI MOTOR COMPANY et al.) 12 October 2022 (2022-10-12) See paragraphs [0097] and [0108].	1-13
A	KR 10-2020-0140096 A (SAMSUNG ELECTRONICS CO., LTD.) 15 December 2020 (2020-12-15) See claim 1.	1-13
☐ Further documents are listed in the continuation of Box C. ☒ 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 26 September 2024		Date of mailing of the international search report 26 September 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/KR2024/008958

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)		Publication date (day/month/year)
KR	10-1776448	B1	07 September 2017	BR	112015015802 A2	11 July 2017
				BR	112015015802 B1	27 September 2022
				BR	15015802 A2	11 July 2017
				CN	104919802 A	16 September 2015
				CN	104919802 B	12 October 2018
				EP	2941887 A1	11 November 2015
				JP	2016-506695 A	03 March 2016
				JP	6235042 B2	22 November 2017
				KR	10-2015-0104158 A	14 September 2015
				US	2014-0192149 A1	10 July 2014
				US	9521393 B2	13 December 2016
				WO	2014-107396 A1	10 July 2014
KR	10-2023-0010259	A	18 January 2023	CN	115956363 A	11 April 2023
				EP	4128764 A1	08 February 2023
				EP	4128764 A4	06 September 2023
				JP	2023-532397 A	28 July 2023
				US	11979565 B2	07 May 2024
				US	2022-0385896 A1	01 December 2022
				WO	2022-251828 A1	01 December 2022
JP	2017-152905	A	31 August 2017	JP	6597385 B2	30 October 2019
KR	10-2022-0137552	A	12 October 2022	CN	117044211 A	10 November 2023
				US	2024-0031612 A1	25 January 2024
				WO	2022-211490 A1	06 October 2022
KR	10-2020-0140096	A	15 December 2020	CN	113994691 A	28 January 2022
				EP	3954127 A1	16 February 2022
				EP	3954127 A4	08 June 2022
				US	10817990 B1	27 October 2020
				US	10825140 B1	03 November 2020
				WO	2020-246756 A1	10 December 2020

A. 발명이 속하는 기술분류(국제특허분류(IPC)) H04N 19/85(2014.01)i; H04N 19/117(2014.01)i; H04N 19/70(2014.01)i; H04N 19/172(2014.01)i; G06N 3/02(2006.01)i		
B. 조사된 분야 조사된 최소문헌(국제특허분류를 기재) H04N 19/85(2014.01); G06T 5/00(2006.01); G06T 9/00(2006.01); H04N 13/00(2006.01); H04N 19/126(2014.01); H04N 19/176(2014.01); H04N 19/46(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: 포스트 필터(post-filter), 출력 픽처 정보(output picture information), NNPF SEI messages(neural-network post-filter supplemental enhancement information messages)		
C. 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	KR 10-1776448 B1 (켈컴 인코포레이티드) 2017.09.07 청구항 1-3	1-13
A	KR 10-2023-0010259 A (텐센트 아메리카 엘엘씨) 2023.01.18 청구항 1-2	1-13
A	JP 2017-152905 A (OKI ELECTRIC IND CO., LTD.) 2017.08.31 단락 [0043]-[0046]	1-13
A	KR 10-2022-0137552 A (현대자동차주식회사 등) 2022.10.12 단락 [0097], [0108]	1-13
A	KR 10-2020-0140096 A (삼성전자주식회사) 2020.12.15 청구항 1	1-13
☐ 추가 문헌이 C(계속)에 기재되어 있습니다. ☒ 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 2024년09월26일(26.09.2024)	국제조사보고서 발송일 2024년09월26일(26.09.2024)	
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 10-1776448 B1	2017/09/07	BR 112015015802 A2	2017/07/11
		BR 112015015802 B1	2022/09/27
		BR 15015802 A2	2017/07/11
		CN 104919802 A	2015/09/16
		CN 104919802 B	2018/10/12
		EP 2941887 A1	2015/11/11
		JP 2016-506695 A	2016/03/03
		JP 6235042 B2	2017/11/22
		KR 10-2015-0104158 A	2015/09/14
		US 2014-0192149 A1	2014/07/10
		US 9521393 B2	2016/12/13
		WO 2014-107396 A1	2014/07/10
KR 10-2023-0010259 A	2023/01/18	CN 115956363 A	2023/04/11
		EP 4128764 A1	2023/02/08
		EP 4128764 A4	2023/09/06
		JP 2023-532397 A	2023/07/28
		US 11979565 B2	2024/05/07
		US 2022-0385896 A1	2022/12/01
		WO 2022-251828 A1	2022/12/01
JP 2017-152905 A	2017/08/31	JP 6597385 B2	2019/10/30
KR 10-2022-0137552 A	2022/10/12	CN 117044211 A	2023/11/10
		US 2024-0031612 A1	2024/01/25
		WO 2022-211490 A1	2022/10/06
KR 10-2020-0140096 A	2020/12/15	CN 113994691 A	2022/01/28
		EP 3954127 A1	2022/02/16
		EP 3954127 A4	2022/06/08
		US 10817990 B1	2020/10/27
		US 10825140 B1	2020/11/03
		WO 2020-246756 A1	2020/12/10