(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년09월19일
(11) 등록번호 10-2707540
(24) 등록일자 2024년09월11일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/159* (2014.01)
H04N 19/174 (2014.01) *H04N 19/593* (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/159 (2015.01)
(21) 출원번호 10-2021-7027127
(22) 출원일자(국제) 2020년12월18일
심사청구일자 2021년08월25일
(85) 번역문제출일자 2021년08월25일
(65) 공개번호 10-2021-0118160
(43) 공개일자 2021년09월29일
(86) 국제출원번호 PCT/US2020/065899
(87) 국제공개번호 WO 2021/127365
국제공개일자 2021년06월24일
(30) 우선권주장
62/950,453 2019년12월19일 미국(US)
17/026,748 2020년09월21일 미국(US)
(56) 선행기술조사문헌
US20120189053 A1
(뒷면에 계속)
전체 청구항 수 : 총 15 항

(73) 특허권자
텐센트 아메리카 엘엘씨
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
(72) 발명자
리, 링
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
쉬, 샤오중
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
(뒷면에 계속)
(74) 대리인
양영준, 김연송, 백만기

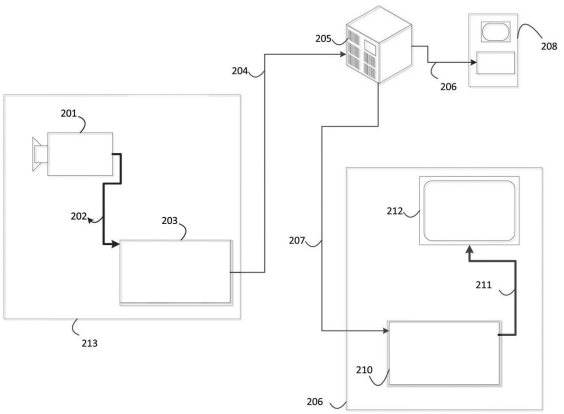
심사관 : 이남숙

(54) 발명의 명칭 **화상 헤더 파라미터들의 시그널링**

(57) 요 약

비디오 데이터를 인코딩 또는 디코딩하고, 코딩된 화상의 모든 슬라이스들에 대한 슬라이스들의 타입들을 신택스 요소로 표시하기 위한 방법, 컴퓨터 프로그램, 및 컴퓨터 시스템이 제공되고, 신택스 요소는 부호 없는 정수를 사용하여 코딩된다.

대 표 도



(52) CPC특허분류
H04N 19/174 (2015.01)
H04N 19/593 (2015.01)

(72) 발명자
최, 병두
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
리, 샹
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
웽거, 스테판
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
류, 산
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내

(56) 선행기술조사문헌
US20140036999 A1
US20140198181 A1
Benjamin Bross, et al., Versatile Video Coding (Draft 7), JVET-P2001-v14, 2019.11.14.*
H. 265: High efficiency video coding. Recommendation ITU-T H.265, 2013.04.
Ling Li, et al. AHG 9: On picture header, JVET-Q0419-v2, 2020.01.03.
Shih-Ta Hsiang, et al. AHG9: Overhead reduction for picture header, Document: JVET-Q0176-v3, 2020.01.09.
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

비디오 데이터를 인코딩하는 방법으로서,

상기 방법은, 적어도 하나의 프로세서에 의해 수행되고,

코딩된 화상의 모든 슬라이스들에 대한 슬라이스들의 타입들을 신택스 요소로 표시하는 단계 - 상기 신택스 요소는 부호 없는 정수를 사용하여 코딩됨 -, 및

상기 신택스 요소를 통해 표시되는 상기 슬라이스들의 타입들에 기초하여 상기 비디오 데이터를 인코딩하는 단계

를 포함하고,

상기 코딩된 화상에 대해, 관련 신택스 요소들만이 코딩되고,

상기 코딩된 화상의 상기 모든 슬라이스들이 인트라 예측을 포함하는 것으로 표시될 때, 인터 예측 신택스 요소들은 코딩되지 않고

화상 헤더 관련 신택스 요소들은 슬라이스 계층 원시 바이트 시퀀스 페이로드 네트워크 추상화 계층 유닛(slice layer raw byte sequence payload network abstraction layer unit)에 포함되고, 플래그는 상기 슬라이스 계층 원시 바이트 시퀀스 페이로드 네트워크 추상화 계층 유닛 내의 상기 화상 헤더 관련 신택스 요소들의 존재를 표시하기 위해 사용되는, 방법.

청구항 2

제1항에 있어서,

상기 슬라이스들의 타입들은 디코딩된 액세스 유닛 디리미터(access unit delimiter) 값으로부터 추론되는, 방법.

청구항 3

제1항에 있어서,

상기 슬라이스들의 타입들은 고 레벨 신택스에서 시그널링될 때 추론되는, 방법.

청구항 4

제1항에 있어서,

상기 슬라이스들의 타입들은 상기 코딩된 화상 내의 직사각형 슬라이스들의 수에 기초하여 추론되는, 방법.

청구항 5

제1항에 있어서,

상기 신택스 요소는 0차 지수-골룸-코딩된(0-th order Exp-Golomb-coded) 신택스 요소인, 방법.

청구항 6

제1항에 있어서,

상기 신택스 요소는 3개의 상태로 구성가능한 2 비트 신택스 요소 또는 4개의 상태로 구성가능한 2 비트 신택스 요소인, 방법.

청구항 7

비디오 데이터를 인코딩하기 위한 장치로서,

컴퓨터 프로그램 코드를 저장하도록 구성된 적어도 하나의 메모리; 및

상기 적어도 하나의 메모리에 액세스하고, 제1항 내지 제6항 중 어느 한 항의 방법을 수행하도록 상기 컴퓨터 프로그램 코드에 따라 동작하도록 구성된 적어도 하나의 프로세서

를 포함하는, 장치.

청구항 8

비디오 데이터를 디코딩하는 방법으로서,

상기 방법은, 적어도 하나의 프로세서에 의해 수행되고,

신택스 요소로부터, 코딩된 화상의 모든 슬라이스들에 대한 슬라이스들의 타입들을 결정하는 단계 - 상기 신택스 요소는 부호 없는 정수를 사용하여 코딩됨 -, 및

상기 신택스 요소를 통해 표시되는 상기 슬라이스들의 타입들에 기초하여 상기 비디오 데이터를 디코딩하는 단계

를 포함하고,

상기 코딩된 화상에 대해, 관련 신택스 요소들만이 코딩되고,

상기 코딩된 화상의 모든 슬라이스들이 인트라 예측을 포함하는 것으로 표시될 때, 인터 예측 신택스 요소들은 코딩되지 않고

화상 헤더 관련 신택스 요소들은 슬라이스 계층 원시 바이트 시퀀스 페이로드 네트워크 추상화 계층 유닛(slice layer raw byte sequence payload network abstraction layer unit)에 포함되고, 플래그는 상기 슬라이스 계층 원시 바이트 시퀀스 페이로드 네트워크 추상화 계층 유닛 내의 상기 화상 헤더 관련 신택스 요소들의 존재를 표시하기 위해 사용되는, 방법.

청구항 9

제8항에 있어서,

상기 슬라이스들의 타입들은 디코딩된 액세스 유닛 디리미터(access unit delimiter) 값으로부터 추론되는, 방법.

청구항 10

제8항에 있어서,

상기 슬라이스들의 타입들은 고 레벨 신택스에서 시그널링될 때 추론되는, 방법.

청구항 11

제8항에 있어서,

상기 슬라이스들의 타입들은 상기 코딩된 화상 내의 직사각형 슬라이스들의 수에 기초하여 추론되는, 방법.

청구항 12

제8항에 있어서,

상기 신택스 요소는 0차 지수-골룸-코딩된(0-th order Exp-Golomb-coded) 신택스 요소인, 방법.

청구항 13

제8항에 있어서,

상기 신택스 요소는 3개의 상태로 구성가능한 2 비트 신택스 요소 또는 4개의 상태로 구성가능한 2 비트 신택스 요소인, 방법.

청구항 14

비디오 데이터를 디코딩하기 위한 장치로서,

컴퓨터 프로그램 코드를 저장하도록 구성된 적어도 하나의 메모리; 및

상기 적어도 하나의 메모리에 액세스하고, 제8항 내지 제13항 중 어느 한 항의 방법을 수행하도록 상기 컴퓨터 프로그램 코드에 따라 동작하도록 구성된 적어도 하나의 프로세서

를 포함하는, 장치.

청구항 15

명령어들을 저장하는 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령어들은 적어도 하나의 프로세서로 하여금 제1항 내지 제6항 중 어느 한 항의 방법 또는 제8항 내지 제13항 중 어느 한 항의 방법을 수행하게 하는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

기 술 분 야

[0001] 관련 출원에 대한 상호-참조

[0002] 본 출원은 2019년 12월 19일자로 출원된 미국 가특허 출원 제62/950,453호, 및 2020년 9월 21일자로 출원된 미국 특허 출원 제17/026,748호로부터 우선권을 주장하며, 이들 전체는 본원에 포함된다.

[0003] 기술분야

[0004] 본 개시내용은 일반적으로 비디오 인코딩/디코딩에 관한 것이고, 일반적으로, 고효율 비디오 코딩(High Efficiency Video Coding)(HEVC)을 넘어서는 차세대 비디오 코딩 기술들, 예컨대 다용도 비디오 코딩(Versatile Video Coding)(VVC)을 설명한다. 더 구체적으로, 본 개시내용은 일반적으로 화상 헤더 핸들링(picture header handling)과 관련된 방법들 및 장치들에 관한 것이다.

배 경 기 술

[0005] 제안되는 VVC 드래프트 7은, 예컨대, 화상의 모든 슬라이스들에 대해 동일한 값들을 갖도록 제약된 슬라이스 헤더(들) 내의 신택스 요소들의 시그널링을 피하기 위해, 코딩된 화상의 모든 슬라이스들에 적용되는 신택스 요소들을 포함하는 화상 헤더로 지칭되는 고 레벨 신택스(High Level Syntax)(HLS)를 포함한다.

[0006] 화상 파라미터 세트

[0007] HLS는 하위 레벨 코딩 툴들에 적용될 수 있는 신택스 요소들을 지정한다. 예컨대, 코딩 트리 유닛(Coding Tree

Unit)(CTU) 사이즈는 시퀀스 레벨 또는 시퀀스 파라미터 세트(Sequence Parameter Set)(SPS)에서 지정될 수 있고, 일반적으로 화상마다 변경되지 않는다. 전형적인 HLS는 SPS, 화상 파라미터 세트(Picture Parameter Set)(PPS), 화상 헤더(Picture Header)(PH), 슬라이스 헤더(Slice Header)(SH), 및 적응형 파라미터 세트(Adaptive Parameter Set)(APS)를 포함한다.

[0008] 상이한 HLS는 애플리케이션들의 레벨들을 포함하고, 그에 따라, 일반적으로 사용되는 신택스 요소들은 반복적으로 코딩될 필요가 없다. 예컨대, SPS는 시퀀스 레벨들에 적용가능한 일반적인 신택스 요소들을 지정한다. PH는 하나 이상의 슬라이스로 구성될 수 있는 코딩된 화상에 적용가능한 일반적인 신택스 요소들을 지정한다.

[0009] VVC 드래프트 7에서 PPS에 포함된 신택스 요소들은 다음과 같이 설명된다:

표 1: VVC 드래프트 7에서 PPS에 포함된 신택스 요소들

신택스 요소	디스크립터
pic_parameter_set_rbsp( ) {	
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
scaling_window_flag	u(1)
if(scaling_window_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
mixed_nalu_types_in_pic_flag	u(1)
pps_subpic_id_signalling_present_flag	u(1)
if(pps_subpics_id_signalling_present_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	

[0010]

pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i++]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
entropy_coding_sync_enabled_flag	u(1)
if(!no_pic_partition_flag || entropy_coding_sync_enabled_flag)	
entry_point_offsets_present_flag	u(1)
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)

[0011]

rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
log2_transform_skip_max_size_minus2	ue(v)
cu_qp_delta_enabled_flag	u(1)
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbcr_qp_offset_present_flag	u(1)
if(pps_joint_cbcr_qp_offset_present_flag)	
pps_joint_cbcr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
joint_cbcr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
if(constant_slice_header_params_enabled_flag) {	
pps_dep_quant_enabled_idc	u(2)
for(i = 0; i < 2; i++)	
pps_ref_pic_list_sps_idc[i]	u(2)
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(v)
}	

[0012]

picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data( ))	
pps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

[0013]

[0014] 위의 표 1에 예시된 바와 같이, num_slices_in_pic_minus1 플러스 1은 PPS를 참조하는 각각의 화상 내의 직사각형 슬라이스들의 수를 지정한다. num_slices_in_pic_minus1의 값은 0 내지 MaxSlicesPerPicture - 1의 범위(0 및 MaxSlicesPerPicture - 1을 포함함)에 있다. no_pic_partition_flag가 1과 동일할 때, num_slices_in_pic_minus1의 값은 0과 동일한 것으로 추론될 수 있다.

[0015] 위의 표 1에 예시된 바와 같이, 0과 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 신택스 요소 mvd_l1_zero_flag가 존재한다는 것을 지정한다. 또한, 1 또는 2와 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 mvd_l1_zero_flag가 존재하지 않는다는 것을 지정한다. 게다가, 3과 동일한 pps_mvd_l1_zero_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다.

[0016] 위의 표 1에 예시된 바와 같이, 0과 동일한 pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더에 신택스 요소 collocated_from_l0_flag가 존재한다는 것을 지정한다. 또한, 1 또는 2와 동일한 pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더에 신택스 요소 collocated_from_l0_flag가 존재하지 않는다는 것을 지정한다. 게다가, 3과 동일한 pps_collocated_from_l0_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다.

[0017] 위의 표 1에 예시된 바와 같이, 0과 동일한 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재한다는 것을 지정한다. 또한, 0 초과의 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재하지 않는다는 것을 지정한다. pps_six_minus_max_num_merge_cand_plus1의 값은 0 내지 6의 범위(0 및 6을 포함함)에 있다.

[0018] 위의 표 1에 예시된 바와 같이, 0과 동일한 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 슬라이스들의 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재한다는 것을 지정한다. 또한, 0 초과의 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재하지 않는다는 것을 지정한다. pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1의 값은 0 내지 MaxNumMergeCand - 1의 범위에 있다.

[0019] 슬라이스 계층 RBSP

[0020] 슬라이스 계층 RBSP는 슬라이스 헤더 및 슬라이스 데이터로 구성될 수 있다.

표 2: 슬라이스 계층 RBSP.

[0021]

신택스 요소	디스크립터
slice_layer_rbsp( ) {	
slice_header( )	
slice_data( )	
rbsp_slice_trailing_bits( )	
}	

[0022] 화상 헤더 및 슬라이스 헤더

[0023] 현재 화상이 참조하는 PPS에서 코딩된 신택스 요소들은, PPS를 참조하는 PH 내의 pic_deblocking_filter_override_flag 또는 PPS를 참조하는 SH 내의 slice_deblocking_filter_override_flag가 설정되도록, PH 및 SH에서 오버라이드될(overridden) 수 있다. PH에 존재하지 않는 신택스 요소들은 그 대신 SH에 존재할 수 있다. 예컨대, SAO 관련 신택스 요소들의 존재를 지정하는 PH 내의 pic_sao_enabled_present_flag의 값이 0일 때, 루마 및 크로마 상의 SAO 사용을 표시하기 위해 slice_sao_luma_flag 및 slice_sao_chroma_flag가 SH에서 코딩될 수 있다.

[0024] PH의 사용으로 인해, 특히 화상에 소수의 슬라이스가 있을 때, 시그널링 오버헤드를 피하기 위해, 화상의 모든 슬라이스들에서 동일하도록 이미 제약된 신택스 요소들이 화상마다 한 번 PH에서 송신될 수 있다. 여전히, 유연성을 제공하기 위해, 슬라이스마다 대체로 변하는 신택스 요소들은 SH에서 송신될 수 있다.

[0025] VVC 드래프트 7에서 PH 및 SH에 포함된 신택스 요소들은 아래의 표 3 및 표 5에서 설명된다.

표 3: 일반적인 슬라이스 헤더 신택스

신택스 요소	디스크립터
picture_header_rbsp( ) {	
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	

[0026]

if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	

[0027]

}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constraints_override_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
pic_cu_qp_delta_subdiv_intra_slice	ue(v)
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
pic_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	

[0028]

if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	
pic_joint_cbcr_sign_flag	u(1)
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)

[0029]

if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
if (! pps_dep_quant_enabled_idc)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	
pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits( )	
}	

[0030]

slice_header( ) {	
slice_pic_order_cnt_lsb	u(v)
if(subpics_present_flag)	
slice_subpic_id	u(v)
if(rect_slice_flag || NumTilesInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)
slice_type	ue(v)
if(!pic_rpl_present_flag &&((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_sps_flag[i]	u(1)
if(slice_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
slice_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
slice_poc_lsb_lt[i][j]	u(v)
slice_delta_poc_msb_present_flag[i][j]	u(1)
if(slice_delta_poc_msb_present_flag[i][j])	
slice_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(pic_rpl_present_flag || ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) || sps_idr_rpl_present_flag)) {	
if((slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) || (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {	

[0031]

[0032]

num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplsIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(pic_temporal_mvp_enabled_flag) {	
if(slice_type == B && !pps_collocated_from_l0_idc)	
collocated_from_l0_flag	u(1)
if((collocated_from_l0_flag && NumRefIdxActive[0] > 1) || (!collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
collocated_ref_idx	ue(v)
}	
if((pps_weighted_pred_flag && slice_type == P) || (pps_weighted_bipred_flag && slice_type == B))	
pred_weight_table( )	
}	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_joint_cbcr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !pic_sao_enabled_present_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	

slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
}	
if(deblocking_filter_override_enabled_flag && !pic_deblocking_filter_override_present_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(entry_point_offsets_present_flag && NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment( )	
}	

[0033]

[0034] 위 및 아래에서 표시된 바와 같이, slice_type은 아래의 표 4에 따라 슬라이스의 코딩 타입을 지정할 수 있다:

표 4: slice_type

slice_type	slice_type의 이름
0	B (B 슬라이스)
1	P (P 슬라이스)
2	I (I 슬라이스)

[0035]

[0036] 액세스 유닛 디리미터(Access Unit Delimiter)

[0037] 액세스 유닛(Access Unit)(AU) 디리미터는 AU의 시작, 및 AU 디리미터 네트워크 추상화 계층(Network Abstraction Layer)(NAL) 유닛을 포함하는 AU 내의 코딩된 화상들에 존재하는 슬라이스들의 타입을 표시하기 위해 사용된다. 현재, AU 디리미터와 연관된 규범적인 디코딩 프로세스는 없다.

[0038] 또한, pic_type은 AU 디리미터 NAL 유닛을 포함하는 AU 내의 코딩된 화상들의 모든 슬라이스들에 대한 slice_type 값들이 pic_type의 주어진 값에 대해 표 4에 열거된 세트의 멤버들인 것을 표시한다. pic_type의 값은 비트스트림들에서 0, 1, 또는 2와 동일할 수 있다. pic_type의 다른 값들은 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다. 이의 이러한 버전을 따르는 디코더들은 pic_type의 예비된 값들을 무시할 수 있다.

표 5: pic_type의 해석

pic_type	AU에 존재할 수 있는 slice_type 값들
0	I
1	P, I
2	B, P, I

[0039]

[0040] 비-특허 문헌 [1]("NPL 1")은 커버되는 저 레벨 코딩 계층들에 대해 파라미터들의 세트가 필요하다는 것을 표시

하기 위한 고 레벨 제어 플래그를 제안한다.

[0041] NPL 1은, 적어도 하나의 인터 코딩된 슬라이스가 있을 때에만, 또는 화상 내부에 서브-파티션이 존재할 때에만, 모든 인터 예측 관련 신택스 요소들 또는 파라미터들이 시그널링될 필요가 있는 방법을 설명한다. 그렇지 않은 경우, 이러한 신택스 요소들 또는 파라미터는 시그널링되지 않는다.

[0042] NPL 1에서 설명되는 일 실시예에서, pic_intra_only_flag로 지칭되는 화상 헤더 내의 제어 플래그는 화상 내부의 모든 슬라이스들(또는 이 화상의 임의의 종류의 서브-파티션들)이 인트라 예측(또는 비-인터 관련 예측)만을 갖게 될 것인지를 표시하기 위해 시그널링된다. 이 플래그가 참일 때, 인트라 코딩 관련 신택스 요소들 또는 파라미터들만이 화상 헤더에서 추후에 시그널링된다. 그렇지 않은 경우, 이 플래그가 거짓일 때, 인터 예측 관련 신택스 요소들 또는 파라미터들이 시그널링된다. 이 실시예를 반영하는 신택스 표가 아래에 제공된다:

표 6: NPL 1의 제1 실시예

신택스 요소	디스크립터
pic_intra_only_flag	u(1)
if(!pic_intra_only_flag){	
if(sps_temporal_mvp_enabled_flag && !pps_temporal_mvp_enabled_idc)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag && !pps_five_minus_max_num_subblock_merge_cand_plus1)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_dmvr_slice_present_flag)	
pic_disable_bdof_dmvr_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_minus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	

[0043]

[0044] NPL 1의 다른 방법에서, 인트라 슬라이스 또는 인트라 서브-파티션에 대해서만 사용되는 모든 관련 신택스 요소들 또는 파라미터들은, 인터 코딩된 슬라이스가 없을 때 또는 서브-파티션이 화상 내부에 존재할 때, 시그널링될 필요가 있다. 그렇지 않은 경우, 이러한 신택스 요소들 또는 파라미터는 시그널링되지 않는다.

[0045] NPL 1의 다른 실시예에서, pic_inter_only_flag로 지칭되는 화상 헤더 내의 제어 플래그는 화상 내부의 모든 슬라이스들(또는 이 화상의 임의의 종류의 서브-파티션)이 인터 예측(또는 비-인트라 관련 예측)을 갖게 될 것인지를 표시하기 위해 시그널링된다. 이 플래그가 참일 때, 인트라 슬라이스 관련 신택스 요소들 또는 파라미터들은 화상 헤더에서 추후에 시그널링되지 않는다. 그렇지 않은 경우, 이 플래그가 거짓일 때, 화상 내의 슬라이스(들) 또는 서브-파티션(들) 중 적어도 하나에서 인트라 슬라이스가 사용될 수 있다. 인트라 슬라이스 또는 서브-파티션들에 대한 관련 신택스 요소들 또는 파라미터들이 시그널링될 것이다. 이 실시예를 반영하는 신택스 표가 아래에 제공된다:

표 7: NPL 1의 제2 실시예

신택스 요소	디스크립터
pic_inter_only_flag	u(1)
if(!pic_inter_only_flag){	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_chroma	ue(v)
pic_max_mtt_hierarchy_depth_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	

[0046]

[0047] NPL 1에서 설명되는 위의 방법들에서, 화상이 인트라 화상 또는 인터 화상인 것과 같이 자체적인 타입을 갖는 경우, 위의 제어 플래그들 pic_intra_only_flag 및 pic_inter_only_flag가 시그널링될 필요가 없고, 이들의 값들은 화상 타입으로부터 도출될 수 있다.

[0048] 또한, 현재 화상이 인트라-전용(intra-only) 화상(화상 내의 모든 슬라이스들이 I 슬라이스들임)으로서 화상 타입을 갖는 경우, pic_intra_only_flag는 참으로서 추론될 수 있다. 다른 예에서, 현재 화상이 인터-전용(inter-only) 화상(화상 내의 모든 슬라이스들이 P 또는 B 슬라이스들임)으로서 화상 타입을 갖는 경우, pic_inter_only_flag는 참으로서 추론될 수 있다. NPL 1의 또 다른 예에서, 현재 화상이 인트라 슬라이스들과 인터 슬라이스들 둘 모두가 화상에서 가능한 것을 표시하는 화상 타입을 갖는 경우, pic_intra_only_flag와 pic_inter_only_flag 둘 모두는 거짓으로서 추론될 수 있다.

[0049] 문제점

[0050] PH가 화상 내의 슬라이스들에 공통인 신택스 요소들의 시그널링을 피하기 위해 화상마다 한 번 시그널링될 수 있지만, 이러한 시그널링은, 인트라 슬라이스들(I 슬라이스들) 또는 인터 슬라이스들(B, P 슬라이스들)에 대해서만 사용되는 신택스 요소들을 고려하지 않는 경우, 대신 오버헤드를 도입할 수 있다.

발명의 내용

[0051] 실시예들은 비디오 인코딩/디코딩을 위한 방법, 시스템, 및 컴퓨터 판독가능 매체에 관한 것이고, 더 구체적으로는 화상 헤더 핸들링에 관한 것이다.

도면의 간단한 설명

[0052] 이들 및 다른 목적들, 피처들, 및 이점들은 첨부 도면들과 관련하여 읽게 될 예시적인 실시예들의 다음의 상세한 설명으로부터 명백하게 될 것이다. 도면들의 다양한 피처들은 실척대로 도시된 것이 아닌데, 이는 예시들이 상세한 설명과 함께 관련 기술분야의 통상의 기술자의 이해를 용이하게 하는 데 있어서의 명료성을 위한 것이기 때문이다.

도 1은 실시예에 따른 통신 시스템의 단순화된 블록도의 개략도이다.

도 2는 실시예에 따른 통신 시스템의 단순화된 블록도의 개략도이다.

도 3은 실시예에 따른 컴퓨터 시스템의 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0053] 도 1은 본 개시내용의 실시예에 따른 통신 시스템(100)의 단순화된 블록도를 예시한다. 시스템(100)은 네트워크(150)를 통해 상호연결되는 적어도 2개의 단말(110-120)을 포함할 수 있다. 데이터의 단방향 송신을 위해, 제1 단말(110)은 네트워크(150)를 통해 다른 단말(120)로 송신하기 위해 로컬 위치에서 비디오 데이터를 코딩할 수 있다. 제2 단말(120)은 네트워크(150)로부터 다른 단말의 코딩된 비디오 데이터를 수신할 수 있고, 코딩된

데이터를 디코딩할 수 있고, 복구된 비디오 데이터를 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 애플리케이션(media serving application)들 등에서 일반적일 수 있다.

[0054] 도 1은, 예컨대, 화상 회의 동안 발생할 수 있는 코딩된 비디오의 양방향 송신을 지원하기 위해 제공되는 제2쌍의 단말들(130, 140)을 예시한다. 데이터의 양방향 송신을 위해, 각각의 단말(130, 140)은 네트워크(150)를 통해 다른 단말로 송신하기 위해 로컬 위치에서 캡처된 비디오 데이터를 코딩할 수 있다. 각각의 단말(130, 140)은 또한, 다른 단말에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 데이터를 디코딩할 수 있고, 복구된 비디오 데이터를 로컬 디스플레이 디바이스에서 디스플레이할 수 있다.

[0055] 도 1에서, 단말들(110-140)은 서버들, 개인용 컴퓨터들, 및 스마트폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 그렇게 제한되지 않을 수 있다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들, 및/또는 전용 화상 회의 장비에 적용된다. 네트워크(150)는, 예컨대 유선 및/또는 무선 통신 네트워크들을 포함하여, 단말들(110-140) 사이에서 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크를 표현한다. 통신 네트워크(150)는 회선 교환 및/또는 패킷 교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 전기통신(telecommunications) 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들, 및/또는 인터넷을 포함한다. 본 논의의 목적들을 위해, 네트워크(150)의 아키텍처 및 토폴로지는, 본원에서 아래에 설명되지 않는 한, 본 개시내용의 동작에 중요하지 않을 수 있다.

[0056] 도 2는, 개시되는 주제에 대한 애플리케이션에 대한 예로서, 스트리밍 환경에서 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시되는 주제는, 예컨대, 화상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 매체들 상의 압축된 비디오의 저장 등을 포함하는 다른 비디오 인에이블 애플리케이션들에 동등하게 적용가능할 수 있다.

[0057] 스트리밍 시스템은, 예컨대 압축되지 않은 비디오 샘플 스트림(202)을 생성하는 비디오 소스(201), 예컨대 디지털 카메라를 포함할 수 있는 캡처 서브시스템(213)을 포함할 수 있다. 인코딩된 비디오 비트스트림들과 비교할 때 높은 데이터 볼륨을 강조하기 위해 굵은 라인으로서 도시된 그 샘플 스트림(202)은 카메라(201)에 커플링된 인코더(203)에 의해 프로세싱될 수 있다. 인코더(203)는, 아래에서 더 상세히 설명되는 바와 같은 개시되는 주제의 양태들을 가능하게 하거나 또는 구현하기 위해, 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 샘플 스트림과 비교할 때 더 낮은 데이터 볼륨을 강조하기 위해 얇은 라인으로서 도시된 인코딩된 비디오 비트스트림(204)은 향후의 사용을 위해 스트리밍 서버(205) 상에 저장될 수 있다. 하나 이상의 스트리밍 클라이언트(206, 208)는, 인코딩된 비디오 비트스트림(204)의 카피들(207, 209)을 검색하기 위해, 스트리밍 서버(205)에 액세스할 수 있다. 클라이언트(206)는 비디오 디코더(210)를 포함할 수 있고, 비디오 디코더(210)는 인코딩된 비디오 비트스트림(207)의 착신 카피를 디코딩하고, 발신 비디오 샘플 스트림(211)을 생성하고, 발신 비디오 샘플 스트림(211)은 디스플레이(212) 또는 다른 렌더링 디바이스(도시되지 않음) 상에 렌더링될 수 있다. 일부 스트리밍 시스템들에서, 비디오 비트스트림들(204, 207, 209)은 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 ITU-T 권고 H.265를 포함한다. 다용도 비디오 코딩 또는 VVC로서 비공식적으로 알려져 있는 비디오 코딩 표준이 개발 중에 있다. 개시되는 주제는 VVC의 컨텍스트에서 사용될 수 있다.

[0058] 실시예들에서, 신택스 요소 pic_type_idc는 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 표시하기 위해 사용될 수 있다.

[0059] 일 실시예에서, pic_type_idc는 먼저 좌측 비트를 갖는 부호 없는 정수 0차 지수-골룸-코딩된(unsigned integer 0-th order Exp-Golomb-coded) 신택스 요소를 사용하여 코딩될 수 있다. 여기서, pic_type_idc는 3개의 값 0, 1 및 2, 및 3개의 상태, 이를테면, I 슬라이스 전용, B, P, I 슬라이스들 및 B, P 슬라이스들을 가질 수 있다. 값은 임의의 순서로 상태들로 매핑될 수 있다. 아래의 표 8은 가능한 pic_type_idc 시맨틱스의 예들을 도시한다.

표 8: 가능한 pic_type_idc 시맨틱스의 예

[0060]

pic_type_idc	코딩된 화상에 존재하는 slice_type
0	B, P, I
1	I
2	B, P

[0061] 실시예에서, pic_type_idc는 2개의 비트를 사용하는 부호 없는 정수를 사용하여 코딩될 수 있다. 여기서,

pic_type_idc는 3개의 값 0, 1 및 2, 및 3개의 상태, 이를테면: I 슬라이스 전용, B, P, I 슬라이스들 및 B, P 슬라이스들을 가질 수 있지만, 반드시 이에 제한되는 것은 아니다. pic_type_idc의 다른 값들이 추가 정의를 위해 예비될 수 있다.

표 9 - 가능한 pic_type_idc 시맨틱스의 예들

pic_type_idc	코딩된 화상에 존재하는 slice_type
0	B, P, I
1	I
2	B, P
3	예비됨

[0062]

[0063] 실시예에서, 예비된 pic_type_idc 값 3은 코딩된 화상에 P, I 슬라이스들만이 존재하는 것을 표시할 수 있다.

[0064] 예에서, pic_type_idc는 2개의 비트를 사용하는 부호 없는 정수를 사용하여 코딩될 수 있다. 또한, pic_type_idc는 4개의 값 0, 1, 2 및 3, 및 4개의 상태, 이를테면, I 슬라이스 전용, B, P, I 슬라이스들, B 슬라이스들 및 P 슬라이스들을 가질 수 있다.

표 10 - 가능한 pic_type_idc 시맨틱스의 예들

pic_type_idc	코딩된 화상에 존재하는 slice_type
0	B, P, I
1	I
2	B
3	P

[0065]

[0066] 시그널링 오버헤드를 감소시키기 위해, 관련 신택스 요소들만이 코딩되거나 또는 존재하도록, HLS에서 pic_type_idc를 시그널링하는 것이 제안된다. 예컨대, pic_type_idc가 화상이 인트라 전용인 것을 표시할 때, 인터 관련 신택스 요소들은 시그널링되지 않는다.

[0067] 일 예에서, pic_type_idc가 PPS를 참조하는 각각의 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정하도록, pic_type_idc가 PPS에서 시그널링될 수 있다. 상세한 신택스 및 시맨틱스는 다음과 같이 주어진다. 아래의 표 뿐만 아니라 본 개시내용의 다른 표들에서, VVC 드래프트 7과 비교할 때의 변화들은 이탤릭체로 되어 있다.

표 11: 상세한 신택스 및 시맨틱스

신택스 요소	디스크립터
pic_parameter_set_rbsp( ) {	
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_type_idc	*ue(v)*
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
scaling_window_flag	u(1)
if(scaling_window_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
mixed_nalu_types_in_pic_flag	u(1)
pps_subpic_id_signalling_present_flag	u(1)
if(pps_subpics_id_signalling_present_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
rect_slice_flag	u(1)
if(rect_slice_flag)	

[0068]

single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i++]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
entropy_coding_sync_enabled_flag	u(1)
if(!no_pic_partition_flag || entropy_coding_sync_enabled_flag)	
entry_point_offsets_present_flag	u(1)
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
log2_transform_skip_max_size_minus2	ue(v)
cu_qp_delta_enabled_flag	u(1)
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbcr_qp_offset_present_flag	u(1)
if(pps_joint_cbcr_qp_offset_present_flag)	
pps_joint_cbcr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)

[0069]

pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
joint_cbcr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
if(constant_slice_header_params_enabled_flag) {	
pps_dep_quant_enabled_idc	u(2)
for(i = 0; i < 2; i++)	
pps_ref_pic_list_sps_idc[i]	u(2)
if (pic_type_idc !=1) {	
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(v)
}	
}	
picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data( ))	
pps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

[0070]

[0071] 일 예에서, pic_type_idc는 PPS를 참조하는 각각의 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정한다.

[0072] 일 실시예에서, 1과 동일한 pic_type_idc 세트는 PPS를 참조하는 각각의 코딩된 화상이 하나 이상의 I 슬라이스만을 갖는다는 것을 표시한다. 그러한 경우들에서, 인터 슬라이스들(B, P 슬라이스) 관련 신택스 요소들 pps_mvd_l1_zero_idc, pps_collocated_from_l0_idc, pps_six_minus_max_num_merge_cand_plus1, 및 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 0과 동일한 것으로 추론된다.

[0073] 여기서, 0과 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 신택스 요소 mvd_l1_zero_flag가 존재한다는 것을 지정한다. 또한, 1 또는 2와 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 mvd_l1_zero_flag가 존재하지 않는다는 것을 지정한다. 게다가, 3과 동일한 pps_mvd_l1_zero_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다. 존재하지 않을 때, pps_mvd_l1_zero_idc는 0인 것으로 추론될 수 있다.

[0074] 추가적으로, 0와 동일한 pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더에 신택스 요소 collocated_from_l0_flag가 존재한다는 것을 지정한다. 또한, 1 또는 2와 동일한

pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더에 신택스 요소 collocated_from_l0_flag가 존재하지 않는다는 것을 지정한다. 게다가, 3과 동일한 pps_collocated_from_l0_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다. 존재하지 않을 때, pps_collocated_from_l0_idc는 0과 동일한 것으로 추론될 수 있다.

[0075] 또한, 0과 동일한 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재한다는 것을 지정한다. 추가적으로, 0 초과의 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재하지 않는다는 것을 지정한다. pps_six_minus_max_num_merge_cand_plus1의 값은 0 내지 6의 범위(0 및 6을 포함함)에 있다. 존재하지 않을 때, pps_six_minus_max_num_merge_cand_plus1은 0과 동일한 것으로 추론될 수 있다.

[0076] 예시된 바와 같이, 0과 동일한 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 슬라이스들의 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재한다는 것을 지정한다. 또한, 0 초과의 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재하지 않는다는 것을 지정한다. pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1의 값은 0 내지 MaxNumMergeCand - 1의 범위에 있다. 존재하지 않을 때, pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 0과 동일한 것으로 추론될 수 있다.

표 12: 제안된 화상 헤더 RBSP 신택스

[0077]

신택스 요소	디스크립터
picture_header_rbsp( ) {	
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)

if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	

[0078]

ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constraints_override_flag) {	
if (pic_type_idc !=2) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	*ue(v)*
pic_log2_diff_max_tt_min_qt_inter_slice	*ue(v)*
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	

[0079]

[0080]

pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if (pic_type_idc !=1) {	
pic_log2_diff_min_qt_min_cb_inter_slice	
pic_max_mtt_hierarchy_depth_inter_slice	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
if (pic_type_idc !=2)	
pic_cu_qp_delta_subdiv_intra_slice	*ue(v)*
if (pic_type_idc !=1)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
if (pic_type_idc !=2)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	*ue(v)*
if (pic_type_idc !=1)	
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	
if (pic_type_idc !=1) {	
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)

if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	
pic_joint_cbcr_sign_flag	u(1)
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	

[0081]

}	
if (! pps_dep_quant_enabled_idc)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	
pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits( )	
}	

[0082]

[0083] PPS를 참조하는 각각의 코딩된 화상에 대해, pic_type_idc는 인트라 슬라이스들(I 슬라이스들) 및 인터 슬라이스들(B, P 슬라이스들)과 관련된 신택스 요소들을 파싱할지 여부를 결정하기 위해 사용된다. 예컨대, 인트라 슬라이스 관련 신택스 요소들 pic_log2_diff_min_qt_min_cb_intra_slice_luma, pic_max_mtt_hierarchy_depth_intra_slice_luma, pic_log2_diff_max_bt_min_qt_intra_slice_luma, pic_log2_diff_max_tt_min_qt_intra_slice_luma, pic_log2_diff_min_qt_min_cb_intra_slice_chroma, pic_max_mtt_hierarchy_depth_intra_slice_chroma, pic_log2_diff_max_bt_min_qt_intra_slice_chroma, 및 pic_log2_diff_max_tt_min_qt_intra_slice_chroma는 I 슬라이스들만이 PH에 연관될 때마다 디코딩된다. 다른 한편으로, 인터 슬라이스 관련 신택스 요소들은 인터 슬라이스들이 존재할 때마다 디코딩된다.

[0084] 일 예에서, pic_type_idc가 PH에 연관된 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정하도록, pic_type_idc가 PH에서 시그널링된다. 상세한 신택스 및 시맨틱스는 다음과 같이 주어진다. VVC 드래프트 7과 비교할 때의 변화들은 이탤릭체로 되어 있다.

표 13: 제안된 화상 헤더 RBSP 신택스

신택스 요소	디스크립터
picture_header_rbsp( ) {	
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
pic_type_idc	*ue(v)*
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	

[0085]

ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	

[0086]

for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constraints_override_flag) {	
if (pic_type_idc !=2) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	*ue(v)*
pic_log2_diff_max_tt_min_qt_inter_slice	*ue(v)*
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)

[0087]

pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if (pic_type_idc !=1) {	
pic_log2_diff_min_qt_min_cb_inter_slice	
pic_max_mtt_hierarchy_depth_inter_slice	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
if (pic_type_idc !=2)	
pic_cu_qp_delta_subdiv_intra_slice	*ue(v)*
if (pic_type_idc !=1)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
if (pic_type_idc !=2)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	*ue(v)*
if (pic_type_idc !=1)	
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	
if (pic_type_idc !=1) {	
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)

[0088]

if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	
pic_joint_cbcr_sign_flag	u(1)
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
if (! pps_dep_quant_enabled_idc)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	

[0089]

sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	
pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits( )	
}	

[0090]

[0091] 각각의 코딩된 화상에 대해, pic_type_idc는 인트라 슬라이스들(I 슬라이스) 및 인터 슬라이스들(B, P 슬라이스)과 관련된 신택스 요소들을 파싱할지 여부를 결정하기 위해 사용된다. 예컨대, 인트라 슬라이스 관련 신택스 요소들 pic_log2_diff_min_qt_min_cb_intra_slice_luma, pic_max_mtt_hierarchy_depth_intra_slice_luma, pic_log2_diff_max_bt_min_qt_intra_slice_luma, pic_log2_diff_max_tt_min_qt_intra_slice_luma, pic_log2_diff_min_qt_min_cb_intra_slice_chroma, pic_max_mtt_hierarchy_depth_intra_slice_chroma, pic_log2_diff_max_bt_min_qt_intra_slice_chroma, 및 pic_log2_diff_max_tt_min_qt_intra_slice_chroma는 I 슬라이스들만이 PH에 연관될 때마다 디코딩된다. 다른 한편으로, 인터 슬라이스 관련 신택스 요소들은 인터 슬라이스들이 존재할 때마다 디코딩된다.

[0092] 일 실시예에서, pic_type_idc는 각각 pps_pic_type_idc 및 ph_pic_type_idc로서 PPS를 참조하는 PH와 PPS 둘 모두에 존재할 수 있다.

표 14: 제안된 화상 파라미터 세트 RBSP 신택스

[0093]

신택스 요소	**디스크립터**
pic_parameter_set_rbsp( ) {	
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_pic_type_idc	*ue(v)*
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
scaling_window_flag	u(1)
if(scaling_window_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)

mixed_nalu_types_in_pic_flag	u(1)
pps_subpic_id_signalling_present_flag	u(1)
if(pps_subpics_id_signalling_present_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i++]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	

[0094]

loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
entropy_coding_sync_enabled_flag	u(1)
if(!no_pic_partition_flag || entropy_coding_sync_enabled_flag)	
entry_point_offsets_present_flag	u(1)
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
log2_transform_skip_max_size_minus2	ue(v)
cu_qp_delta_enabled_flag	u(1)
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbcr_qp_offset_present_flag	u(1)
if(pps_joint_cbcr_qp_offset_present_flag)	
pps_joint_cbcr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
joint_cbcr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
}	
}	
constant_slice_header_params_enabled_flag	u(1)
if(constant_slice_header_params_enabled_flag) {	
pps_dep_quant_enabled_idc	u(2)

[0095]

for(i = 0; i < 2; i++)	
pps_ref_pic_list_sps_idc[i]	u(2)
if (pps_pic_type_idc !=1) {	
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(v)
}	
}	
picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data( ))	
pps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

[0096]

[0097] 여기서, pps_pic_type_idc는 PPS를 참조하는 각각의 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정한다.

[0098] 또한, 0과 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 신택스 요소 mvd_l1_zero_flag가 존재한다는 것을 지정한다. 추가로, 1 또는 2와 동일한 pps_mvd_l1_zero_idc는 PPS를 참조하는 PH들에 mvd_l1_zero_flag가 존재하지 않는다는 것을 지정한다. 추가적으로, 3과 동일한 pps_mvd_l1_zero_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다. 존재하지 않을 때, pps_mvd_l1_zero_idc는 0과 동일한 것으로 추론될 수 있다.

[0099] 또한, 0와 동일한 pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더들에 신택스 요소 collocated_from_l0_flag가 존재한다는 것을 지정한다. 추가로, 1 또는 2와 동일한 pps_collocated_from_l0_idc는 PPS를 참조하는 슬라이스들의 슬라이스 헤더들에 신택스 요소 collocated_from_l0_flag가 존재하지 않는다는 것을 지정한다. 게다가, 3과 동일한 pps_collocated_from_l0_idc는 ITU-T | ISO/IEC에 의한 향후의 사용을 위해 예비된다. 존재하지 않을 때, pps_collocated_from_l0_idc는 0과 동일한 것으로 추론될 수 있다.

[0100] 또한, 0과 동일한 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재한다는 것을 지정한다. 추가로, 0 초과의 pps_six_minus_max_num_merge_cand_plus1은 PPS를 참조하는 PH들에 pic_six_minus_max_num_merge_cand가 존재하지 않는다는 것을 지정한다. pps_six_minus_max_num_merge_cand_plus1의 값은 0 내지 6의 범위(0 및 6을 포함함)에 있을 것이다. 존재하지 않을 때, pps_six_minus_max_num_merge_cand_plus1은 0과 동일한 것으로 추론될 수 있다.

[0101] 또한, 0과 동일한 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 슬라이스들의 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재한다는 것을 지정한다. 추가로, 0 초과의 pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 PPS를 참조하는 PH들에 pic_max_num_merge_cand_minus_max_num_triangle_cand가 존재하지 않는다는 것을 지정한다. pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1의 값은 0 내지 MaxNumMergeCand - 1의 범위에 있을 것이다. 존재하지 않을 때, pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1은 0과 동일한 것으로 추론될 수 있다.

[0102] 일 예에서, pps_pic_type_idc의 값이 하나의 타입의 슬라이스(표 10의 값 1, 2 및 3에서와 같은 I 또는 B 또는 P 슬라이스)의 존재를 표시할 때, ph_pic_type_idc의 값은 pps_pic_type_idc의 값으로부터 추론될 수 있다.

표 15: 제안된 화상 헤더 RBSP 신택스

신택스 요소	디스크립터
picture_header_rbsp( ) {	
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
ph_pic_type_idc	*ue(v)*
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	
ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	

[0103]

if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)
pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constraints_override_flag) {	
if (ph_pic_type_idc !=2) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)

[0104]

pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	*ue(v)*
pic_log2_diff_max_tt_min_qt_inter_slice	*ue(v)*
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if (ph_pic_type_idc !=1) {	
pic_log2_diff_min_qt_min_cb_inter_slice	
pic_max_mtt_hierarchy_depth_inter_slice	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
if (ph_pic_type_idc !=2)	
pic_cu_qp_delta_subdiv_intra_slice	*ue(v)*
if (ph_pic_type_idc !=1)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
if (ph_pic_type_idc !=2)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	*ue(v)*
if (ph_pic_type_idc !=1)	

[0105]

pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	
if (ph_pic_type_idc !=1) {	
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)
if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	
pic_joint_cbcr_sign_flag	u(1)
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	

[0106]

[0107]

if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
if (! pps_dep_quant_enabled_idc)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	
pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	

pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits( )	
}	

[0108]

[0109] 여기서, ph_pic_type_idc는 PH에 연관된 각각의 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정한다.

[0110] 일 실시예에서, 1과 동일한 ph_pic_type_idc는 PH에 연관된 각각의 코딩된 화상이 하나 이상의 I 슬라이스만을 갖는다는 것을 표시한다.

표 16: 가능한 pic_type_idc 시맨틱스의 예들

ph_pic_type_idc	코딩된 화상에 존재하는 slice_type
0	B, P, I
1	I
2	B, P

[0111]

[0112] pps_pic_type_idc가 0(표 10에서와 같이 B, P, I 슬라이스들)과 동일한 경우, ph_pic_type_idc의 값은 0 내지 2의 범위(0 및 2를 포함함)를 갖는다. 그렇지 않은 경우, ph_pic_type_idc의 값은 pps_pic_type_idc로부터 추론될 수 있다(예컨대, 동일한 값). 이러한 경우, ph_pic_type_idc의 값들이 pps_pic_type_idc의 값들과 동일한 것이 비트스트림 순응(bitstream conformance) 요건이다.

[0113] 일 예에서, 신택스 ph_pic_type_idc의 시그널링은 pps_pic_type_idc의 값에 따라 좌우된다(예컨대, 이에 의해 제약됨). pps_pic_type_idc의 값이 코딩된 화상들 내의 인트라 슬라이스들(I 슬라이스)과 인터 슬라이스들(B, P 슬라이스) 둘 모두의 존재를 표시할 때, 화상 헤더와 연관된 화상에 존재하는 슬라이스 타입들을 표시하기 위해 ph_pic_type_idc가 시그널링/파싱될 필요가 있을 수 있다. 다른 경우들에서, pps_pic_type_idc가 단지 하나의 슬라이스 타입의 존재를 표시할 때, ph_pic_type_idc는 시그널링/파싱되지 않고, 이는 pps_pic_type_idc의 슬라이스 타입과 동일한 것으로(예컨대, 동일한 슬라이스 타입을 갖는 것으로) 추론된다. ph_pic_type_idc의 범위가 pps_pic_type_idc의 범위보다 더 크지 않다는 것이 비트스트림 순응 요건이다.

표 17: 제안된 화상 헤더 RBSP 신택스

신택스 요소	디스크립터
picture_header_rbsp( ) {	
non_reference_picture_flag	u(1)
gdr_pic_flag	u(1)
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
ph_pic_parameter_set_id	ue(v)
if (pps_pic_type_idc==0)	
ph_pic_type_idc	*ue(v)*
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_subpic_id_present_flag && !sps_subpic_id_signalling_flag) {	
ph_subpic_id_signalling_present_flag	u(1)
if(ph_subpics_id_signalling_present_flag) {	

[0114]

ph_subpic_id_len_minus1	ue(v)
for(i = 0; i <= sps_num_subpics_minus1; i++)	
ph_subpic_id[i]	u(v)
}	
}	
if(!sps_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_loop_filter_across_virtual_boundaries_disabled_present_flag	u(1)
if(ph_loop_filter_across_virtual_boundaries_disabled_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(output_flag_present_flag)	
pic_output_flag	u(1)
pic_rpl_present_flag	u(1)
if(pic_rpl_present_flag) {	
for(i = 0; i < 2; i++) {	
if(num_ref_pic_lists_in_sps[i] > 0 && !pps_ref_pic_list_sps_idc[i] && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_sps_flag[i]	u(1)
if(pic_rpl_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1 && (i == 0 || (i == 1 && rpl1_idx_present_flag)))	
pic_rpl_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
for(j = 0; j < NumLtrpEntries[i][RplsIdx[i]]; j++) {	
if(ltrp_in_slice_header_flag[i][RplsIdx[i]])	
pic_poc_lsb_lt[i][j]	u(v)

[0115]

pic_delta_poc_msb_present_flag[i][j]	u(1)
if(pic_delta_poc_msb_present_flag[i][j])	
pic_delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constraints_override_flag) {	
if (ph_pic_type_idc !=2) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_inter_slice	*ue(v)*
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	*ue(v)*
pic_log2_diff_max_tt_min_qt_inter_slice	*ue(v)*
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if (ph_pic_type_idc !=1) {	
pic_log2_diff_min_qt_min_cb_inter_slice	
pic_max_mtt_hierarchy_depth_inter_slice	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	

[0116]

pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
pic_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
}	
}	
if(cu_qp_delta_enabled_flag) {	
if (ph_pic_type_idc !=2)	
pic_cu_qp_delta_subdiv_intra_slice	*ue(v)*
if (ph_pic_type_idc !=1)	
pic_cu_qp_delta_subdiv_inter_slice	ue(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
if (ph_pic_type_idc !=2)	
pic_cu_chroma_qp_offset_subdiv_intra_slice	*ue(v)*
if (ph_pic_type_idc !=1)	
pic_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
}	
if (ph_pic_type_idc !=1) {	
if(sps_temporal_mvp_enabled_flag)	
pic_temporal_mvp_enabled_flag	u(1)
if(!pps_mvd_l1_zero_idc)	
mvd_l1_zero_flag	u(1)
if(!pps_six_minus_max_num_merge_cand_plus1)	
pic_six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enabled_flag)	
pic_five_minus_max_num_subblock_merge_cand	ue(v)
if(sps_fpel_mmvd_enabled_flag)	
pic_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
pic_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
pic_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
pic_disable_prof_flag	u(1)

[0117]

if(sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
if (sps_ibc_enabled_flag)	
pic_six_minus_max_num_ibc_merge_cand	ue(v)
if(sps_joint_cbcr_enabled_flag)	
pic_joint_cbcr_sign_flag	u(1)
if(sps_sao_enabled_flag) {	
pic_sao_enabled_present_flag	u(1)
if(pic_sao_enabled_present_flag) {	
pic_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
pic_sao_chroma_enabled_flag	u(1)
}	
}	
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
}	
}	
if (! pps_dep_quant_enabled_idc)	
pic_dep_quant_enabled_flag	u(1)
if(!pic_dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag) {	

[0118]

pic_deblocking_filter_override_present_flag	u(1)
if(pic_deblocking_filter_override_present_flag) {	
pic_deblocking_filter_override_flag	u(1)
if(pic_deblocking_filter_override_flag) {	
pic_deblocking_filter_disabled_flag	u(1)
if(!pic_deblocking_filter_disabled_flag) {	
pic_beta_offset_div2	se(v)
pic_tc_offset_div2	se(v)
}	
}	
}	
}	
if(sps_lmcs_enabled_flag) {	
pic_lmcs_enabled_flag	u(1)
if(pic_lmcs_enabled_flag) {	
pic_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
pic_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
pic_scaling_list_present_flag	u(1)
if(pic_scaling_list_present_flag)	
pic_scaling_list_aps_id	u(3)
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
rbsp_trailing_bits( )	
}	

[0119]

[0120] 여기서, ph_pic_type_idc는 PH에 연관된 각각의 코딩된 화상의 모든 슬라이스들에 대한 슬라이스 타입들을 지정한다. 또한, ph_pic_type_idc는 pps_pic_type_idc가 0과 동일할 때에만 비트스트림에 존재할 수 있다.

[0121] 추가로, 1과 동일한 ph_pic_type_idc는 PH에 연관된 각각의 코딩된 화상이 하나 이상의 I 슬라이스만을 갖는다는 것을 표시한다. pps_pic_type_idc가 0(표 8에서와 같이 B, P, I 슬라이스들)과 동일한 경우, ph_pic_type_idc의 값은 0 내지 2의 범위(0 및 2를 포함함)를 갖는다. 그렇지 않고, ph_pic_type_idc가 존재하지 않을 때, 이는 표 8에서와 같이 pps_ph_type_idc와 동일한 것으로 추론된다.

[0122] 일 실시예에서, PH 관련 신택스 요소들은 슬라이스 계층 RBSP NAL 유닛에 포함되고, ph_present_flag는 슬라이스 계층 RBSP NAL 유닛 내의 PH 관련 신택스의 존재를 표시하기 위해 사용된다. PH 관련 신택스 시그널링의 반복은 에러 내성(error resilience) 및 에러 복구의 이점을 가질 수 있다. PH NAL 유닛이 임의의 종류의 네트워크에서 송신 동안 손상될 때, 슬라이스 계층 RBSP NAL 유닛들은 슬라이스 계층 RBSP NAL 유닛들 내의 PH의 존재로 인해 에러로부터 복구될 수 있다. VVC 드래프트 7과 비교할 때의 변화들은 이탤릭체로 되어 있다.

표 18: 제안된 슬라이스 계층 RBSP 신택스

신택스 요소	디스크립터
slice_layer_rbsp( ) {	
ph_present_flag	*u(1)*
if (ph_present_flag)	
picture_header_rbsp( )	
slice_header( )	
slice_data( )	
rbsp_slice_trailing_bits( )	
}	

[0123]

[0124] 여기서, ph_present_flag는 슬라이스 계층 RBSP 내의 PH 관련 신택스의 존재를 지정하기 위해 사용될 수 있다. ph_present_flag가 1과 동일할 때, PH 관련 신택스가 존재한다. ph_present_flag가 0과 동일할 때, PH 관련 신택스가 슬라이스 계층 RBSP에 존재하지 않는다.

[0125] 일 실시예에서, 위에서 설명된 바와 같이 AU 디리미터에서 디코딩된 pic_type이 존재할 때, HLS에서 시그널링되는 pic_type_idc는 pic_type 값으로부터 추론될 수 있거나 또는 그에 의해 제약될 수 있다.

[0126] 일 예에서, pic_type이 표 5에서와 같이 0과 동일하여 I 슬라이스를 표시할 때, pic_type_idc의 값들이 각각의 화상에 인트라 슬라이스들만이 있다는 것을 지정하는 것이 비트스트림 순응 요건이다. 예컨대, pic_type_idc가 1인 것으로 순응될 때, 인트라 슬라이스들만이 있다.

[0127] 일 예에서, pic_type_idc가 pic_type 값에 의해 제약될 때, pic_type_idc 값의 범위는 pic_type의 값에 따라 좌우될 수 있다. 예컨대, pic_type_idc는 표 10에서 설명된 바와 같은 값들을 갖고, pic_type이 1과 동일한 경우, pic_type_idc의 값은 1 또는 3을 가질 수 있다. 다른 경우들에서, pic_type이 2와 동일할 때, pic_type_idc의 값은 0 내지 3의 범위에 있다.

[0128] 일 실시예에서, 위에서 언급된 방법(들)에 따라, HLS에서 pic_type_idc가 시그널링될 때, slice_type이 추론될 수 있다.

[0129] 일 예에서, pic_type_idc가 인트라 슬라이스들만이 있다는 것을 표시하는 값을 가질 때, slice_type은 2인 것으로 추론될 수 있다.

[0130] 일 예에서, pic_type_idc가 인터 슬라이스들만이 있다는 것을 표시하는 값을 가질 때, slice_type의 값은 0 내지 1(0 및 1을 포함함)의 범위를 갖는다. 예컨대, pic_type_idc가 2의 값(B, P 슬라이스)을 가질 때, slice_type에 대한 가능한 값들은 0 및 1이다.

[0131] 일 실시예에서, slice_type의 값은 값 pic_type_idc 및 num_slices_in_pic_minus1로부터 추론될 수 있다.

[0132] pic_type_idc의 값이 인트라 슬라이스들과 인터 슬라이스들 둘 모두가 있다는 것을 표시할 때, num_slices_in_pic_minus1의 값들이 1 이상인 것이 비트스트림 순응 요건이다.

[0133] pic_type_idc의 값이 인트라 슬라이스들과 인터 슬라이스들 둘 모두가 코딩된 화상에 존재한다는 것을 표시할 때, num_slices_in_pic_minus1의 값이 1 이상인 경우들이 있을 수 있다.

[0134] 모든 이전에 코딩된 슬라이스들이 인터 슬라이스들일 때, 최종 슬라이스는 slice_type이 2(I 슬라이스)와 동일한 인트라 슬라이스일 수 있다.

[0135] 모든 이전에 코딩된 슬라이스들이 인트라 슬라이스들일 때, 최종 슬라이스는 slice_type 값이 0 내지 1의 범위(0 및 1을 포함함)에 있는 인터 슬라이스일 수 있다.

[0136] 위에서 제안된 방법들은 프로세싱 회로부(예컨대, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 일 예에서, 하나 이상의 프로세서는 제안된 방법들 중 하나 이상을 수행하기 위해 비-일시적 컴퓨터 판독가능 매체에 저장된 프로그램을 실행한다.

[0137] 위에서 설명된 기법들은 컴퓨터 판독가능 명령어들을 사용하여 컴퓨터 소프트웨어로서 구현될 수 있고, 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예컨대, 도 3은 개시되는 주제의 특정 실시예들을 구현하는 데 적합한 컴퓨터 시스템(300)을 도시한다.

[0138] 컴퓨터 소프트웨어는, 컴퓨터 중앙 프로세싱 유닛(CPU)들, 그래픽 프로세싱 유닛(GPU)들 등에 의해, 직접적으로, 또는 해석, 마이크로-코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해, 어셈블리, 컴필레이션, 링킹, 또는 유사한 메커니즘들을 거칠 수 있는 임의의 적합한 기계 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.

[0139] 명령어들은, 예컨대, 개인용 컴퓨터들, 태블릿 컴퓨터들, 서버들, 스마트폰들, 게이밍 디바이스들, 사물 인터넷 디바이스들 등을 포함하는 다양한 타입의 컴퓨터들 또는 이들의 구성요소들 상에서 실행될 수 있다.

[0140] 컴퓨터 시스템(300)에 대한 도 3에 도시된 구성요소들은 본질적으로 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대해 어떠한 제한도 제시하는 것으로 의도되지 않는다. 또한, 구성요소들의 구성이 컴퓨터 시스템(300)의 예시적인 실시예에서 예시되는 구성요소들 중 임의의 하나 또는 이들의 조합과 관련된 임의의 종속성 또는 요건을 갖는 것으로 해석되지 않아야 한다.

[0141] 컴퓨터 시스템(300)은 특정 인간 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 입력 디바이스는, 예컨대, 촉각적 입력(이를테면: 키스트로크들, 스와이프들, 데이터 글러브 움직임(data glove movement)들), 오디오 입력(이를테면: 음성, 박수), 시각적 입력(이를테면: 제스처들), 후각적 입력(도시되지 않음)을 통한 하나 이상의 인간 사용자에 의한 입력에 응답할 수 있다. 인간 인터페이스 디바이스들은 또한, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련될 필요는 없는 특정 매체들, 이를테면, 오디오(이를테면: 스피치, 음악, 주변 사운드), 이미지들(이를테면: 스캐닝된 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(이를테면, 2차원 비디오, 스테레오스코픽 비디오를 포함하는 3차원 비디오)를 캡처하기 위해 사용될 수 있다.

[0142] 입력 인간 인터페이스 디바이스들은: 키보드(301), 마우스(302), 트랙패드(303), 터치 스크린(310) 및 연관된 그래픽 어댑터(350), 데이터 글러브, 조이스틱(305), 마이크로폰(306), 스캐너(307), 카메라(308) 중 하나 이상(각각 하나만 도시됨)을 포함할 수 있다.

[0143] 컴퓨터 시스템(300)은 또한, 특정 인간 인터페이스 출력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 출력 디바이스들은, 예컨대, 촉각적 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하고 있을 수 있다. 그러한 인간 인터페이스 출력 디바이스들은 촉각적 출력 디바이스들(예컨대, 터치 스크린(310), 데이터 글러브, 또는 조이스틱(305)에 의한 촉각적 피드백, 그러나 입력 디바이스들로서 기능하지 않는 촉각적 피드백 디바이스들이 또한 있을 수 있음), 오디오 출력 디바이스들(이를테면: 스피커들(309), 헤드폰들(도시되지 않음)), 시각적 출력 디바이스들(이를테면, 음극선관(CRT) 스크린들, 액정 디스플레이(LCD) 스크린들, 플라즈마 스크린들, 유기 발광 다이오드(OLED) 스크린들을 포함하는 스크린들(310), 이들 각각은 터치 스크린 입력 능력을 갖거나 또는 갖지 않고, 이들 각각은 촉각적 피드백 능력을 갖거나 또는 갖지 않음 - 이들 중 일부는 스테레오그래픽 출력; 가상 현실 안경(도시되지 않음), 홀로그래픽 디스플레이들 및 스모크 탱크(smoke tank)들(도시되지 않음)과 같은 수단을 통해 3차원 초과의 출력을 출력하는 것이 가능할 수 있거나 또는 2차원 시각적 출력을 출력하는 것이 가능할 수 있음), 및 프린터들(도시되지 않음)을 포함할 수 있다.

[0144] 컴퓨터 시스템(300)은 또한, 인간 액세스가능 저장 디바이스들 및 이들의 연관된 매체들, 이를테면, CD/DVD 또는 유사한 매체들(321)을 갖는 CD/DVD ROM/RW(920)를 포함하는 광학 매체들, 섬-드라이브(thumb-drive)(322), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(323), 레거시 자기 매체들, 이를테면 테이프 및 플로피 디스크(도시되지 않음), 특수 ROM/ASIC/PLD 기반 디바이스들, 이를테면 보안 동글들(도시되지 않음) 등을 포함할 수 있다.

[0145] 관련 기술분야의 통상의 기술자는 또한, 현재 개시되는 주제와 관련하여 사용되는 바와 같은 "컴퓨터 판독가능 매체들"이라는 용어가 송신 매체들, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 것을 이해해야 한다.

[0146] 컴퓨터 시스템(300)은 또한, 하나 이상의 통신 네트워크(355)에 대한 인터페이스(들)를 포함할 수 있다. 네트워크들은 예컨대 무선, 유선, 광학일 수 있다. 네트워크들은 추가로, 로컬, 광역, 도시권, 차량 및 산업, 실시간, 지연 감내형 등일 수 있다. 네트워크들의 예들은 로컬 영역 네트워크들, 이를테면 이더넷, 무선 LAN들, GSM(global systems for mobile communications), 3세대(3G), 4세대(4G), 5세대(5G), 롱-텀 에볼루션(LTE) 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 브로드캐스트 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 일반적으로, 특정 범용 데이터 포트들 또는 주변 버스들(349)(이를테면 예컨대, 컴퓨터 시스템(300)의 범용 직렬 버스(USB) 포

트들)에 부착된 외부 네트워크 인터페이스 어댑터들(354)을 요구하고; 다른 것들은 일반적으로, 아래에서 설명되는 바와 같이 시스템 버스에 대한 부착에 의해 컴퓨터 시스템(300)의 코어 내에 통합된다(예컨대, PC 컴퓨터 시스템 내로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템 내로의 셀룰러 네트워크 인터페이스). 예로서, 네트워크(355)는 네트워크 인터페이스(354)를 사용하여 주변 버스(349)에 연결될 수 있다. 이러한 네트워크들 중 임의의 네트워크를 사용하여, 컴퓨터 시스템(300)은 다른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향 수신 전용(예컨대, 브로드캐스트 TV), 단방향 전송 전용(예컨대, 특정 CANbus 디바이스들에 대한 CANbus), 또는 예컨대 로컬 또는 광역 디지털 네트워크들을 사용한 다른 컴퓨터 시스템들에 대한 양방향성일 수 있다. 위에서 설명된 바와 같은 그러한 네트워크들 및 네트워크 인터페이스들(354) 각각 상에서 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0147] 전술된 인간 인터페이스 디바이스들, 인간 액세스가능 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(300)의 코어(340)에 부착될 수 있다.

[0148] 코어(340)는 하나 이상의 중앙 프로세싱 유닛(CPU)들(341), 그래픽 프로세싱 유닛(GPU)들(342), 필드 프로그래머블 게이트 영역들(FPGA)(343)의 형태의 특수 프로그래머블 프로세싱 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(344) 등을 포함할 수 있다. 이러한 디바이스들은, 판독 전용 메모리(ROM)(345), 랜덤 액세스 메모리(RAM)(346), 내부 대용량 저장소, 이를테면 내부 비-사용자 액세스가능 하드 드라이드들, 솔리드 스테이트 드라이브(SSD)들 등(347)과 함께, 시스템 버스(348)를 통해 연결될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(348)는, 추가적인 CPU들, GPU 등에 의한 확장을 가능하게 하기 위해, 하나 이상의 물리적 플러그의 형태로 액세스가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(348)에 직접적으로 또는 주변 버스(349)를 통해 부착될 수 있다. 주변 버스에 대한 아키텍처들은 PCI(peripheral component interconnect), USB 등을 포함한다.

[0149] CPU들(341), GPU들(342), FPGA들(343), 및 가속기들(344)은, 조합되어 전술된 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(345) 또는 RAM(346)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(346)에 저장될 수 있는 반면에, 영구적인 데이터는 예컨대 내부 대용량 저장소(347)에 저장될 수 있다. 메모리 디바이스들 중 임의의 디바이스에 대한 고속 저장 및 검색은 하나 이상의 CPU(341), GPU(342), 대용량 저장소(347), ROM(345), RAM(346) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

[0150] 컴퓨터 판독가능 매체들은 다양한 컴퓨터에 의해 구현되는 동작들을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체들 및 컴퓨터 코드는 본 개시내용의 목적들을 위해 특별히 설계 및 구성된 것들일 수 있거나, 또는 이들은 컴퓨터 소프트웨어 기술분야의 통상의 기술자에게 널리 공지되고 그 통상의 기술자가 이용가능한 종류일 수 있다.

[0151] 제한이 아닌 예로서, 아키텍처를 갖는 컴퓨터 시스템(300), 구체적으로 코어(340)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의 컴퓨터 판독가능 매체들에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체들은 위에서 소개된 바와 같은 사용자 액세스가능 대용량 저장소 뿐만 아니라, 코어 내부 대용량 저장소(347) 또는 ROM(345)과 같은 비-일시적 성질을 갖는 코어(340)의 특정 저장소와 연관된 매체들일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어는 이러한 디바이스들에 저장될 수 있고 코어(340)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요들에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는, 코어(340), 구체적으로 코어 내부의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(346)에 저장된 데이터 구조들을 정의하는 것 및 소프트웨어에 의해 정의된 프로세스들에 따라 그러한 데이터 구조들을 수정하는 것을 포함하여, 본원에서 설명되는 특정 프로세스들 또는 그 특정 프로세스들의 특정 부분들을 실행하게 할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은, 본원에서 설명되는 특정 프로세스들 또는 그 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 소프트웨어와 함께 동작할 수 있는, 회로(예컨대: 가속기(344))에 하드와이어링되거나 또는 다른 방식으로 구현된 로직의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 언급은 적절한 경우 로직을 포함할 수 있고 그 반대도 마찬가지이다. 컴퓨터 판독가능 매체에 대한 언급은, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(이를테면, 집적 회로(IC)), 실행을 위한 로직을 구현하는 회로, 또는 이들 둘 모두를 포함할 수 있다. 본 개시내용은 하드웨어와 소프트웨어의 임의의 적합한 조합을 포함한다.

[0152] 본 개시내용이 여러 개의 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들,

및 다양한 대체 균등물들이 존재한다. 따라서, 본원에서 명시적으로 도시 또는 설명되지 않았지만 본 개시내용의 원리들을 구현하고 그에 따라 본 개시내용의 사상 및 범위 내에 있는 다수의 시스템 및 방법을 관련 기술분야의 통상의 기술자가 고안하는 것이 가능할 것이라는 점이 인식될 것이다.

[0153] 비-특허문헌:

[0154] [1] IDF_10092019_high level syntax control for video coding_v2

[0155] 약어들의 목록:

[0156] HLS: 고 레벨 신택스

[0157] HEVC: 고효율 비디오 코딩

[0158] VVC: 다용도 비디오 코딩

[0159] CTU: 코딩 트리 유닛

[0160] SPS: 시퀀스 파라미터 세트

[0161] PPS: 화상 파라미터 세트

[0162] APS: 적응형 파라미터 세트

[0163] PH: 화상 헤더

[0164] SH: 슬라이스 헤더

[0165] SAO: 샘플 적응형 오프셋2(Sample Adaptive Offset2)

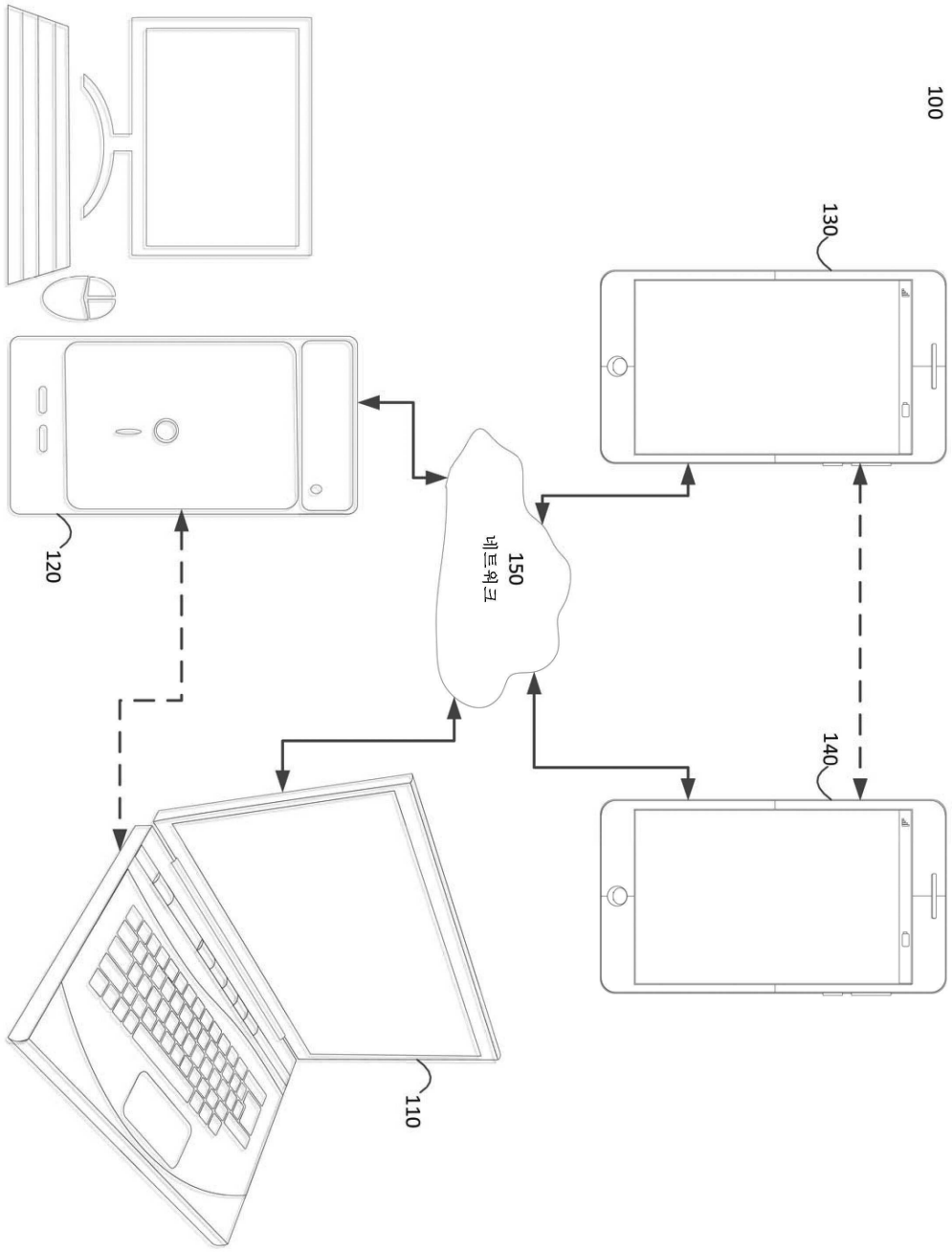
[0166] AU: 액세스 유닛

[0167] NAL: 네트워크 추상화 계층

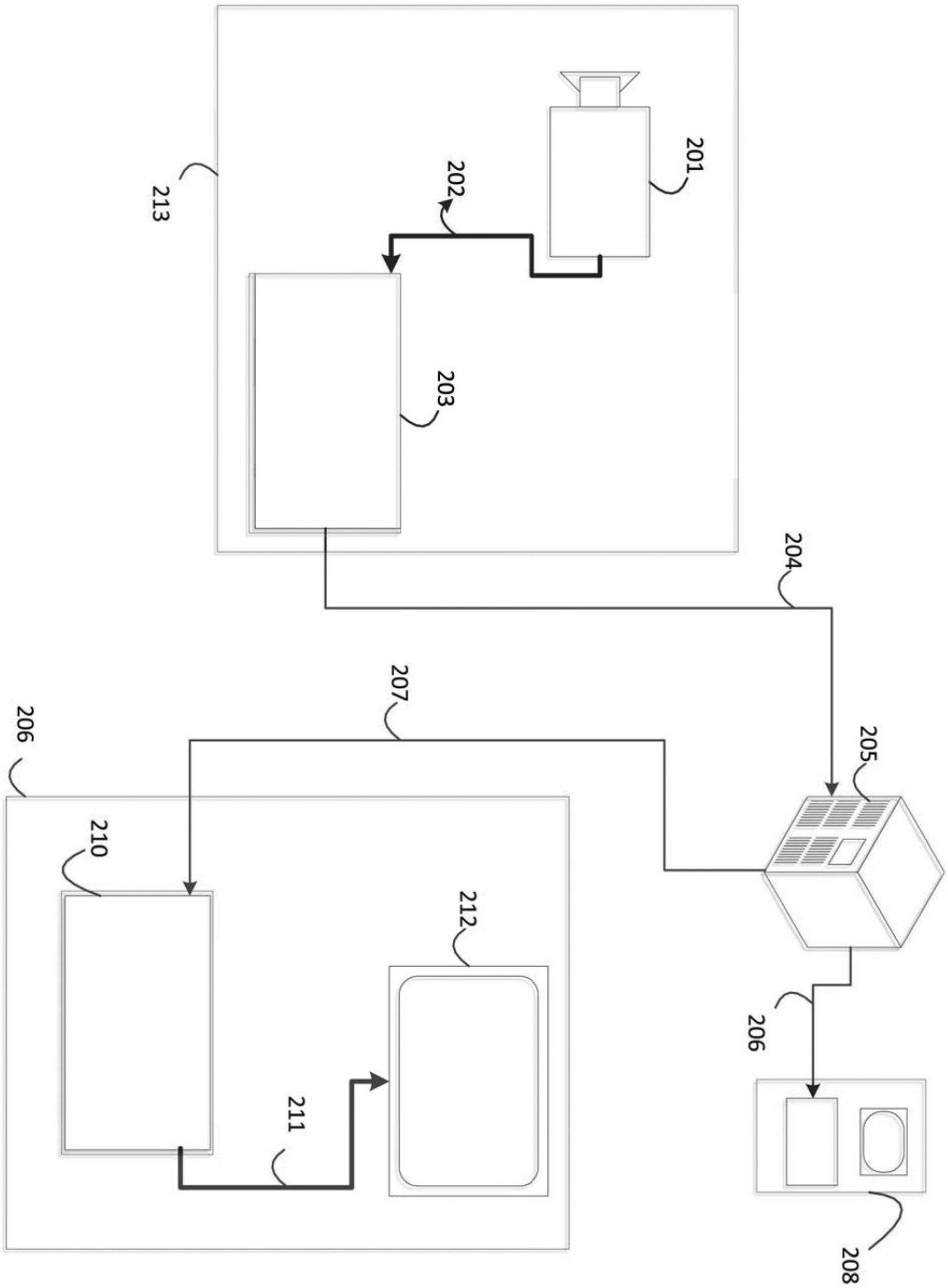
[0168] RBSP: 원시 바이트 시퀀스 페이로드(Raw Byte Sequence Payload)

도면

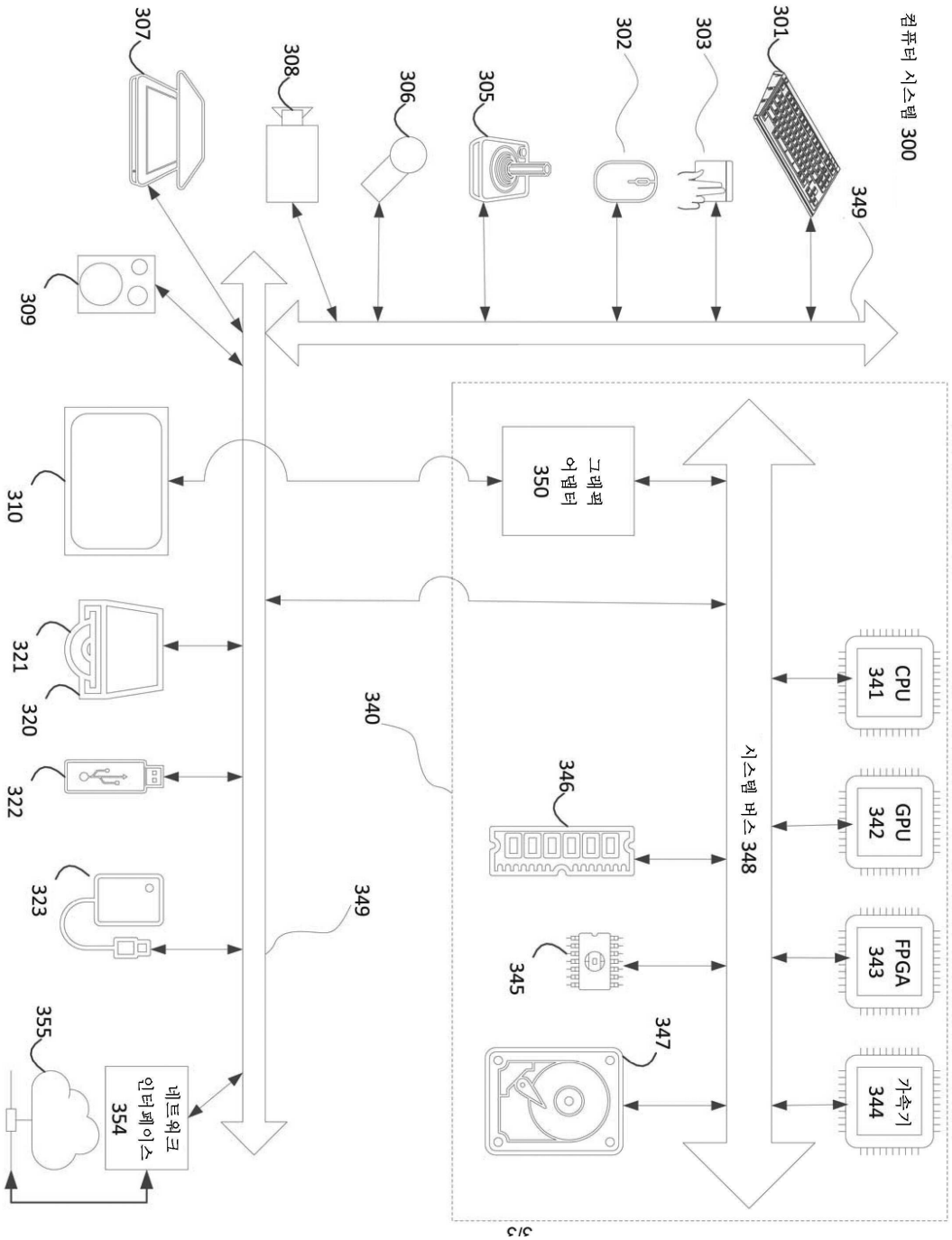
도면1



도면2



도면3


컴퓨터 시스템 300
301
303
302
305
306
308
307
309
310
321
320
322
323
355
네트워크 인터페이스 354
그래픽 어댑터 350
340
346
345
347
349
349
시스템 버스 348
CPU 341
GPU 342
FPGA 343
가속기 344

3/3