US012395650B2

(12) **United States Patent**
Zhu et al.

(10) **Patent No.: US 12,395,650 B2**
(45) **Date of Patent: Aug. 19, 2025**

(54) **GENERAL CONSTRAINTS INFORMATION FOR VIDEO CODING**

(71) Applicants: **Beijing Bytedance Network Technology Co., Ltd.**, Beijing (CN); **Bytedance Inc.**, Los Angeles, CA (US)

(72) Inventors: **Weijia Zhu**, San Diego, CA (US); **Jizheng Xu**, San Diego, CA (US); **Li Zhang**, San Diego, CA (US); **Kai Zhang**, San Diego, CA (US); **Yue Wang**, Beijing (CN)

(73) Assignees: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.**, Beijing (CN); **BYTEDANCE INC.**, Los Angeles, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **18/524,994**

(22) Filed: **Nov. 30, 2023**

(65) **Prior Publication Data**

US 2024/0114156 A1 Apr. 4, 2024

Related U.S. Application Data

(63) Continuation of application No. 17/857,874, filed on Jul. 5, 2022, now Pat. No. 11,838,523, which is a (Continued)

(30) **Foreign Application Priority Data**

Jan. 5, 2020 (WO) PCT/CN2020/070368

(51) **Int. Cl.**
H04N 19/196 (2014.01)
H04N 11/04 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04N 19/196*** (2014.11); ***H04N 11/044*** (2013.01); ***H04N 19/105*** (2014.11);
(Continued)

(58) **Field of Classification Search**
CPC H04N 19/105; H04N 19/196; H04N 19/12; H04N 11/044; H04N 19/186; H04N 19/176; H04N 19/126
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,512,953 A 4/1996 Nahumi
6,016,163 A 1/2000 Rodriguez
(Continued)

FOREIGN PATENT DOCUMENTS

CN 104137545 A 11/2014
CN 104205836 A 12/2014
(Continued)

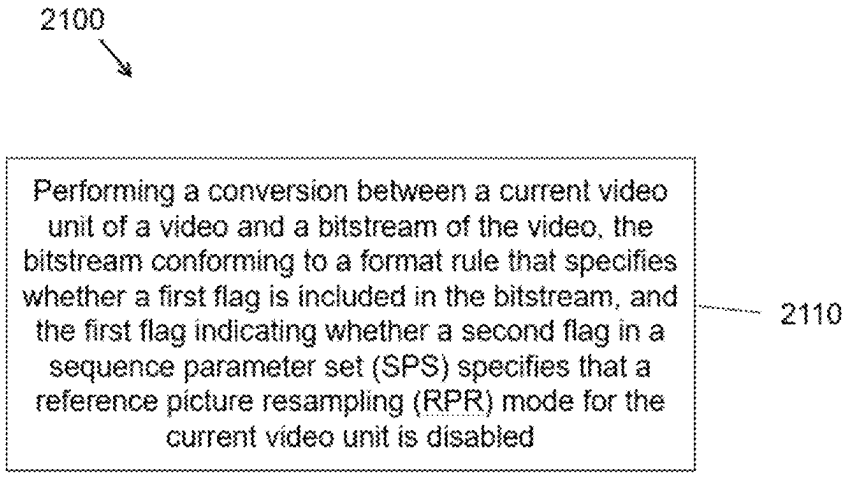
OTHER PUBLICATIONS

Abdoli et al. "Intra Block-DPCM with Layer Separation of Screen Content in VVC," 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, Sep. 25, 2019, retrieved on Feb. 21, 2021 from https://ieeexplore.ieee.org/abstract/Document/8803389.
(Continued)

Primary Examiner — Shawn S An
(74) *Attorney, Agent, or Firm* — Conley Rose, P.C.

(57) **ABSTRACT**

A method for implementing an adaptive color transform (ACT) during image/video encoding and decoding, includes determining, for a conversion between a video including a block and a bitstream of the video, that a size of the block is greater than a maximum allowed size for an ACT mode, and performing, based on the determining, the conversion, wherein, in response to the size of the block being greater than the maximum allowed size for the ACT mode, the block
(Continued)



is partitioned into multiple sub-blocks, and wherein each of the multiple sub-blocks share a same prediction mode, and the ACT mode is enabled at a sub-block level.

18 Claims, 24 Drawing Sheets

Related U.S. Application Data

continuation of application No. PCT/CN2021/070279, filed on Jan. 5, 2021.

(51) **Int. Cl.**

H04N 19/105	(2014.01)
H04N 19/12	(2014.01)
H04N 19/126	(2014.01)
H04N 19/176	(2014.01)
H04N 19/186	(2014.01)

(52) **U.S. Cl.**

CPC ***H04N 19/12*** (2014.11); ***H04N 19/126*** (2014.11); ***H04N 19/176*** (2014.11); ***H04N 19/186*** (2014.11)

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,474,357	B1	1/2009	Murdock	
9,137,547	B2	9/2015	Van der Auwera	
9,538,200	B2	1/2017	Van der Auwera	
9,591,302	B2	3/2017	Sullivan	
9,723,331	B2	8/2017	Van der Auwera	
9,807,403	B2	10/2017	Chong	
10,057,574	B2	8/2018	Li	
10,085,024	B2	9/2018	Yu	
10,097,832	B2	10/2018	Sullivan	
10,319,100	B2	6/2019	Dai	
10,321,130	B2	6/2019	Dong	
10,506,230	B2	12/2019	Zhang	
10,582,213	B2	3/2020	Li	
10,708,591	B2	7/2020	Zhang	
10,708,592	B2	7/2020	Dong	
10,778,974	B2	9/2020	Karczewicz	
10,855,985	B2	12/2020	Zhang	
11,184,643	B2	11/2021	Korta et al.	
11,539,956	B2	12/2022	Li	
11,863,715	B2 *	1/2024	Zhu	H04N 19/103
11,973,959	B2	4/2024	Zhu et al.	
11,991,390	B2 *	5/2024	Yoo	H04N 19/157
2002/0106025	A1 *	8/2002	Tsukagoshi	H04N 19/577 375/E7.091
2003/0206585	A1	11/2003	Kerofsky	
2004/0062313	A1	4/2004	Schoenblum	
2004/0101059	A1	5/2004	Joch	
2004/0146108	A1	7/2004	Hsia	
2007/0201564	A1	8/2007	Joch	
2009/0219994	A1	9/2009	Tu	
2009/0262798	A1	10/2009	Chiu	
2009/0289320	A1	11/2009	Cohen	
2012/0183052	A1	7/2012	Lou	
2012/0192051	A1	7/2012	Rothschiller	
2013/0016772	A1	1/2013	Matsunobu	
2013/0022107	A1	1/2013	Van der Auwera	
2013/0077676	A1	3/2013	Sato	
2013/0101018	A1	4/2013	Chong	
2013/0101025	A1	4/2013	Van der Auwera	
2013/0101031	A1	4/2013	Van der Auwera	
2013/0107970	A1	5/2013	Wang	
2013/0107973	A1	5/2013	Wang	
2013/0188731	A1	7/2013	Kim	
2013/0259141	A1	10/2013	Van der Auwera	
2013/0329785	A1	12/2013	Lim	
2014/0003497	A1	1/2014	Sullivan	
2014/0003498	A1	1/2014	Sullivan	
2014/0192904	A1	7/2014	Rosewarne	
2014/0321552	A1	10/2014	He	
2014/0355689	A1	12/2014	Tourapis	
2014/0362917	A1	12/2014	Joshi	
2014/0376611	A1	12/2014	Kim	
2015/0016506	A1 *	1/2015	Fu	H04N 19/174 375/240.02
2015/0043637	A1	2/2015	Morigami	
2015/0071345	A1	3/2015	Tourapis	
2015/0181976	A1 *	7/2015	Cooper	A43B 13/223 12/146 B
2015/0264354	A1	9/2015	Zhang	
2015/0264374	A1	9/2015	Xiu	
2015/0264402	A1 *	9/2015	Zhang	H04N 19/157 375/240.18
2015/0319438	A1	11/2015	Shima	
2015/0358631	A1	12/2015	Zhang	
2015/0365671	A1	12/2015	Pu	
2015/0365695	A1	12/2015	Pu	
2015/0373327	A1	12/2015	Zhang	
2016/0065991	A1	3/2016	Chen	
2016/0100167	A1	4/2016	Rapaka	
2016/0100168	A1	4/2016	Rapaka	
2016/0100175	A1	4/2016	Laroche	
2016/0179521	A1	6/2016	Jha	
2016/0212373	A1	7/2016	Aharon	
2016/0227224	A1	8/2016	Hsieh	
2016/0241853	A1	8/2016	Lim	
2016/0241858	A1	8/2016	Li	
2016/0261864	A1	9/2016	Samuelsson	
2016/0261884	A1	9/2016	Li	
2016/0286226	A1	9/2016	Ridge	
2016/0286235	A1	9/2016	Yamamoto	
2016/0337661	A1	11/2016	Pang et al.	
2017/0034532	A1	2/2017	Yamamoto	
2017/0048525	A1	2/2017	Lim	
2017/0070754	A1	3/2017	Lim	
2017/0099487	A1	4/2017	Jun	
2017/0105014	A1	4/2017	Lee	
2017/0127077	A1 *	5/2017	Chuang	H04N 19/103
2017/0127090	A1	5/2017	Rosewarne	
2017/0134728	A1	5/2017	Sullivan	
2017/0150151	A1 *	5/2017	Samuelsson	H04N 19/186
2017/0150157	A1	5/2017	Yamori	
2017/0238020	A1	8/2017	Karczewicz	
2017/0310969	A1 *	10/2017	Chen	H04N 19/12
2017/0318301	A1	11/2017	Li	
2017/0332075	A1	11/2017	Karczewicz	
2018/0027246	A1	1/2018	Liu	
2018/0041778	A1	2/2018	Zhang	
2018/0041779	A1	2/2018	Zhang	
2018/0048901	A1 *	2/2018	Zhang	H04N 19/124
2018/0063527	A1	3/2018	Chen	
2018/0084284	A1	3/2018	Rosewarne	
2018/0091812	A1	3/2018	Guo	
2018/0091829	A1	3/2018	Liu	
2018/0109794	A1	4/2018	Wang	
2018/0115787	A1	4/2018	Koo	
2018/0146197	A1	5/2018	Yi	
2018/0160112	A1	6/2018	Gamei	
2018/0199057	A1	7/2018	Chuang	
2018/0205949	A1	7/2018	Hsiang	
2018/0278934	A1	9/2018	Andersson	
2018/0338161	A1	11/2018	Zhai	
2018/0352246	A1	12/2018	Laroche	
2018/0352264	A1	12/2018	Guo	
2019/0020875	A1	1/2019	Liu	
2019/0089984	A1	3/2019	He	
2019/0098307	A1	3/2019	Sullivan	
2019/0116358	A1	4/2019	Zhang	
2019/0124330	A1	4/2019	Chien	
2019/0158831	A1	5/2019	Jung	
2019/0208206	A1	7/2019	Nakagami	
2019/0230353	A1	7/2019	Gadde	
2019/0238845	A1	8/2019	Zhang	
2019/0238849	A1	8/2019	Fang	
2019/0273923	A1	9/2019	Huang	
2019/0306502	A1	10/2019	Gadde	

(56) **References Cited**

U.S. PATENT DOCUMENTS

2020/0021845	A1	1/2020	Lin	
2020/0029072	A1*	1/2020	Xu	H04N 19/105
2020/0177910	A1	6/2020	Li	
2020/0213570	A1	7/2020	Shih	
2020/0236381	A1	7/2020	Chujoh	
2020/0267381	A1	8/2020	Vanam	
2020/0329239	A1	10/2020	Hsiao	
2020/0396467	A1*	12/2020	Lai	H04N 19/103
2020/0413100	A1*	12/2020	Hsiang	H04N 19/105
2021/0006792	A1	1/2021	Han	
2021/0021841	A1*	1/2021	Xu	H04N 19/157
2021/0021863	A1	1/2021	Kalva	
2021/0044820	A1	2/2021	Furht	
2021/0051320	A1*	2/2021	Tourapis	H04N 19/182
2021/0058643	A1	2/2021	Zhao	
2021/0076032	A1	3/2021	Hu	
2021/0099732	A1	4/2021	Ray	
2021/0120239	A1	4/2021	Zhu	
2021/0160479	A1	5/2021	Hsiang	
2021/0176464	A1	6/2021	Ray	
2021/0176479	A1	6/2021	Liao	
2021/0176501	A1	6/2021	Chen	
2021/0195201	A1	6/2021	Li	
2021/0211700	A1	7/2021	Li	
2021/0266550	A1	8/2021	Li	
2021/0266552	A1	8/2021	Kotra	
2021/0266556	A1	8/2021	Choi	
2021/0314628	A1	10/2021	Zhang	
2021/0321121	A1	10/2021	Zhang	
2021/0337239	A1	10/2021	Zhang	
2021/0344903	A1	11/2021	Yu	
2021/0368171	A1	11/2021	Zhang	
2021/0377524	A1	12/2021	Zhang	
2021/0385446	A1	12/2021	Liu	
2021/0385454	A1	12/2021	Fleureau	
2021/0392381	A1	12/2021	Wang	
2021/0409701	A1	12/2021	Zhu	
2022/0014782	A1	1/2022	Chon	
2022/0078433	A1*	3/2022	Yoo	H04N 19/176
2022/0141495	A1*	5/2022	Kim	H04N 19/159 375/240.26
2022/0159282	A1	5/2022	Sim	
2022/0182641	A1	6/2022	Nam et al.	
2022/0191496	A1	6/2022	Kotra	
2022/0201294	A1	6/2022	Nam	
2022/0210408	A1	6/2022	Zhu	
2022/0210448	A1	6/2022	Zhu	
2022/0217410	A1	7/2022	Wang	
2022/0264122	A1	8/2022	Zhu	
2022/0272347	A1	8/2022	Zhu	
2022/0295091	A1	9/2022	Chujoh	
2022/0303567	A1*	9/2022	Jung	H04N 19/46
2022/0312042	A1	9/2022	Deshpande	
2022/0329794	A1	10/2022	Kotra	
2022/0337836	A1*	10/2022	Zhao	H04N 19/186
2022/0345697	A1	10/2022	Zhoi et al.	
2022/0345726	A1	10/2022	Zhao	
2023/0007255	A1	1/2023	Tsukuba	
2023/0018055	A1	1/2023	Hendry et al.	
2023/0130131	A1	4/2023	Chiang	
2023/0156194	A1	5/2023	Sullivan	

FOREIGN PATENT DOCUMENTS

CN	104221376	A	12/2014
CN	104221378	A	12/2014
CN	104584559	A	4/2015
CN	104584560	A	4/2015
CN	105960802	A	9/2016
CN	105979271	A	9/2016
CN	106105203	A	11/2016
CN	106105205	A	11/2016
CN	106416249	A	2/2017
CN	106797465	A	5/2017
CN	107079150	A	8/2017
CN	107079157	A	8/2017
CN	107211122	A	9/2017
CN	107409215	A	11/2017
CN	107431826	A	12/2017
CN	107534782	A	1/2018
CN	107534783	A	1/2018
CN	107846591	A	3/2018
CN	108293124	A	7/2018
CN	109076210	A	12/2018
CN	109196862	A	1/2019
CN	110121884	A	8/2019
CN	114946180	B	8/2022
EP	3425911	A1	1/2019
EP	3507984	A1	7/2019
GB	2506852	A	4/2014
GB	2575090	A	1/2020
JP	2015512600	A	4/2015
JP	2019525679	A	9/2019
JP	2020017970	A	1/2020
JP	2022542851	A	10/2022
JP	2022544164	A	10/2022
JP	2023011955	A	1/2023
JP	2023500644	A	1/2023
JP	7508558	B2	7/2024
RU	2636103	C2	11/2017
TW	201424378	A	6/2014
WO	2008016219	A1	2/2008
WO	2012117744	A1	9/2012
WO	2012177202	A1	12/2012
WO	2014008212	A1	1/2014
WO	2014039547	A1	3/2014
WO	2015015058	A1	2/2015
WO	2015187978	A1	12/2015
WO	2016040865	A1	3/2016
WO	2016049894	A1	4/2016
WO	2016057665	A1	4/2016
WO	2016123232	A1	8/2016
WO	2017052440	A1	3/2017
WO	2018237146	A1	12/2018
WO	2019009776	A1	1/2019
WO	2019069950	A1	4/2019
WO	2019147910	A1	8/2019

OTHER PUBLICATIONS

Document: JVET-P2001-vE, "Versatile Video Coding (Draft 7)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 491 pages.

Gao et al. "AVS—The Chinese Next-Generation Video Coding Standard," National Association Broadcasters, 2004, retrieved on Dec. 16, 2020 from http://www.avs.org.cn/reference/AVS%20NAB%Paper%20Final03.pdf.

Gordon et al. "Mismatch on BDPCM Luma/Chroma Contect Indices Between VTM7 and Spec," in Fraunhofer.de [online] Nov. 14, 2019, retrieved Mar. 1, 2021, retrieved from the internet URL: https://jvet.hhi.fraunhofer.de/trac/vvc/ticket/708.

"High Efficiency Video Coding," Series H: Audiovisual and Multimedia Systems: Infrastructure of Audiovisual Services—Coding of Moving Video, ITU-T Telecommunication Standardization Sector of ITU, H.265, Feb. 2018.

Marpe et al. "An Adaptive Color Transform Approach and its Application in 4:4:4 Video Coding," 2006 14th European Signal Processing Conference, IEEE, 2006, Sep. 8, 2006, retrieved on Feb. 21, 2021 from https:// ieeexplore.ieee.org/abstract/Document/7071266.

Misra et al. "On Cross Component Adaptive Loop Filter for Video Compression," 2019 Picture Coding Symposium (PCS), Nov. 12-15, 2019, Ningbo, China, retrieved from the internet URL:https://ieeexplore.ieee.org/Document/8954547.

Document JVET-Q0058, Misra et al. "CE5 Common Base: Cross Component Adaptive Loop Filter," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, Document JVET-Q0058, 2020.

(56) **References Cited**

OTHER PUBLICATIONS

Norkin et al. "HEVC Deblocking Filter," IEEE Transactions on Circuits and Systems for Video Technology, Dec. 2012, 22(12):1746-1754.

JVET-O0650-v4, Ramasubramonian et al. "AHG15: On Signalling of Chroma QP Tables," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, Document JVET-O0650, 2019.

Sullivan et al. "Standardized Extensions of High Efficiency Video Coding (HEVC)," IEEE Journal of Selected Topics in Signal Processing, Dec. 2013, 7(6):1001-1016.

Bossen, F., et al., Retrieved from the Internet: https://vcgit.hhi.fraunhofer.de/jvetAA/CSoftware_VTM/tagsA/TM-6-0, Sep. 19, 2022, 2 pages.

JEM-7.0: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/ HM-16.6-JEM-7.0, Sep. 19, 2022, 1 page.

Wan W et al: "Consistent chroma QP derivation in deblocking and inverse quantization processes", 102.MPEG Meeting; Oct. 15, 2012-Oct. 19, 2012;Shanghai; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/ WG11), No. m26439, Oct. 9, 2012(Oct. 9, 2012), XP030054772.

JCTVC-K0220, Kanumuri et al. "Use of Chroma QP Offsets in Deblocking," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting, Shanghai, CN, Oct. 10-19, 2012, Document JCTVC-K0220, 2012.

JVET-P1002, Xu et al. "Non-CE5: Chroma QP Derivation Fix for Deblocking Filter (Combination of JVET-P0105 and JVET-P0539)," Joint Video Experts Team (JVET) of ITU-T SG WP 3 and SI/IEC JTC 1/SC 29/WG 11 16th Meeting, Geneva, CH, Oct. 1-11, 2019, Document JVET-P1002, 2019.

JVET-O2002-v2, Chen, J., et al., "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, 89 pages.

JVET-O2001-vE, "Versatile Video Coding (Draft 6), "Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/ IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, 455 pages.

JVET-O1168, Han et al. "Cu Level Chroma QP Control for VVC," Joint Video Experts Team (JVET) of ITU-T SG WP 3 and SI/IEC JTC 1/SC 29/WG 11 15th Meeting, Gothenburg, SE, Jul. 3-12, 2019, Document JVET-O1168, 2019.

Document JCTVC-O0089, Kim et al. "AhG5: Deblocking Filter in 4:4:4 Chroma Format," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Geneva, CH, Oct. 23-Nov. 1, 2013. Document JCTVC-O0089, 2013.

Document: JVET-P1006-v2, Wan, W., et al., "AHG17: Text for picture header," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 5 pages. p. 136.

Document JVET-N0473-r2, Misra et al. "Non-CE11: On ISP Transform Boundary Deblocking," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, Mar. 19-27, 2019, Document JVET-N0473-r2, 2019.

Sjoberg et al. "HEVC High-Level Syntax" In: "High Efficiency Video Coding (HEVC) : Algorithms and Architectures," Aug. 23, 2014 (Aug. 23, 2014), Springer, ISBN: 978-3-319-06895-4 pp. 13-48, DOI:10.1007/978-3-319-06895-4.2.

Document JVET-P0517, Xiu et al. "Support of Adaptive Color Transform for 444 Video Coding in VVC," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting, Geneva, CH, Oct. 1-11, 2019, Document JVET-P0517, 2019.

Document JVET-N0401-v5, De-Luxian-Hernandez et al. "Non-CE3/Non-CE8: Enable Transform Skip in CUs Using ISP," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting, Geneva, CH, Mar. 19-27, 2019, Document JVET-N0401-v5, 2019.

Document JCTVC-U0106, Xiu et al. "On Signaling Adaptive Color Transform at TU Level," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 21st Meeting, Warsaw, PL, Jun. 19-26, 2015, Document JCTVC-U0106, 2015.

Document JVET-O0566, Xu et al. "Non-CE5: Consistent Deblocking for Chroma Components," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, Document JVET-O0566, 2019.

Document JVET-P0105-v2, Kotra et al. "Non-CE5: Modified Chroma QP Derivation for Deblocking Filter ," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, Document JVET-P0105-v2, 2019.

Kotra priority Document PCT/EP2019/072643, 2019.

JVET-O0413-v3 Morigami et al. "Chroma Deblocking Filter Adjustments for 4:4:4 and 4:2:2 Format," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, Document JVET-O0413, 2019.

JVET-Q0253-v1, Meher, A., et al., "CE5-related: High level syntax modifications for CCALF," Joint Video Experts Team (JVET)of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020.

JVET-Q0190-v1, Chubach, O., et al., "CE5-related: On CC-ALF modifications related to coefficients and signalling," Joint Video Experts Team (JVET)of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020.

JCTVC-Y1002—Rosewarne et al. "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 7," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG1125th Meeting: Chengdu, CN, Oct. 14-21, 2016, document JCTVC-Y1002, 2016.

JVET-G1001—Chen et al. "Algorithm Description of Joint Exploration Test Model 7 (JEM 7)," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 7th Meeting: Torino, IT, Jul. 13-21, 2017, document JVET-G1001, 2017.

Document: JVET-P0436, Zhao, J., et al., "AHG15: On CU Adaptive Chroma QP Offset Signalling," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 5 pages.

JVET-P1001-v1—Kotra et al. "Non-CE5: Chroma QP Derivation Fix for Deblocking Filter (Combination of JVET-P0105 and JVET-P0539)," Joint Video Experts Team (JVET of ITU-T SG 16 WP 3 an ISO/IEC JTC 1/SC 29/WG 1 16th Meeting: Geneva, CH, Oct. 1-11, 2019, document JVET-P1001, 2019.

Notice of Allowance from U.S. Appl. No. 17/694,305 dated Feb. 23, 2023, 13 pages.

Non-Final Office Action from U.S. Appl. No. 17/694,253 dated Mar. 24, 2023, 36 pages.

Non Final Office Action from U.S. Appl. No. 17/810,187 dated Nov. 16, 2023, 26 pages.

Final Office Action from U.S. Appl. No. 17/694,305 dated Dec. 11, 2023, 13 pages.

Non-Final Office Action from U.S. Appl. No. 17/856,631 dated Nov. 24, 2023, 32 pages.

Non-Final Office Action from U.S. Appl. No. 17/694,305 dated Mar. 27, 2024, 16 pages.

Notice of Allowance from U.S. Appl. No. 17/694,305 dated Apr. 7, 2023, 14 pages.

Advisory Action from U.S. Appl. No. 17/720,582 dated Jun. 1, 2023, 13 pages.

Non Final Office Action from U.S. Appl. No. 17/720,582 dated Nov. 14, 2022, 47 pages.

Non Final Office Action from U.S. Appl. No. 17/694,305 dated Aug. 10, 2023, 13 pages.

(56) **References Cited**

OTHER PUBLICATIONS

International Search Report and Written Opinion from International Patent Application No. PCT/US2020/050638 dated Nov. 30, 2020 (8 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/050649 dated Feb. 12, 2021 (11 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/054959 dated Feb. 12, 2021 (11 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/055329 dated Jan. 19, 2021 (10 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/055332 dated Jan. 29, 2021 (9 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/063746 dated Apr. 8, 2021 (14 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/067264 dated Mar. 23, 2021 (11 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/067651 dated Mar. 30, 2021 (10 pages).
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/067655 dated Mar. 30, 2021 (10 pages).
Non-Final Office Action from U.S. Appl. No. 17/694,305 dated Jun. 9, 2022, 8 pages.
Extended European Search Report from European Patent Application No. 20862153.2 dated Oct. 5, 2022 (12 pages).
Extended European Search Report from European Patent Application No. 20876422.5 dated Oct. 20, 2022 (7 pages).
Partial Supplementary European Search Report from European Patent Application No. 20877874.6 dated Oct. 31, 2022 (15 pages).
Non Final Office Action from U.S. Appl. No. 17/856,631 dated Nov. 10, 2022, 28 pages.
Non Final Office Action from U.S. Appl. No. 17/852,934 dated Jan. 12, 2023, 37 pages.
Extended European Search Report from European Patent Application No. 20899673.6 dated Dec. 15, 2022 (10 pages).
Non Final Office Action from U.S. Appl. No. 17/810,187 dated Jan. 9, 2023, 33 pages.
Extended European Search Report from European Application No. 20909161.0 dated Feb. 7, 2023, 10 pages.
Extended European Search Report from European Application No. 20877874.6 dated Feb. 10, 2023, 17 pages.
Chinese Office Action from Chinese Application No. 202080088208.7 dated May 31, 2024, 17 pages. With English Translation.
Japanese Office Action from Japanese Application No. 2022-540684 dated Aug. 22, 2023, 18 pages. With English Translation.
Japanese Notice of Allowance from Japanese Application No. 2022-540684 dated Jan. 30, 2024, 6 pages. With English Translation.
Non Final Office Action from U.S. Appl. No. 17/835,647 dated Aug. 17, 2022, 11 pages.
Japanese Office Action from Japanese Application No. 2022-534625 dated May 16, 2023, 27 pages. With English Translation.
Chinese Notice of Allowance from Chinese Application No. 202080072432.7 dated Mar. 30, 2023, 62 pages. With English Translation.
Chinese Notice of Allowance from Chinese Application No. 202080072432.7 dated Oct. 28, 2023, 6 pages. With English Translation.
Chinese Office Action from Chinese Application No. 202080072268.X dated Mar. 30, 2023, 48 pages. With English Translation.
Chinese Office Action from Chinese Application No. 202080072268.X dated Oct. 31, 2023, 7 pages. With English Translation.
Non Final Office Action from U.S. Appl. No. 17/716,447 dated Aug. 4, 2022, 27 pages.
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/51689 dated Dec. 17, 2020 (9 pages).
Chinese Office Action from Chinese Application No. 202080066216.1 dated Jun. 29, 2023, 17 pages. With English Translation.
Non Final Office Action from U.S. Appl. No. 17/720,634 dated Jul. 19, 2022, 25 pages.
Final Office Action from U.S. Appl. No. 17/694,305 dated Sep. 30, 2022, 31 pages.
International Search Report and Written Opinion from International Patent Application No. PCT/US2020/050644 dated Dec. 18, 2020 (9 pages).
Non-Final Office Action from U.S. Appl. No. 17/694,253 dated Jul. 12, 2022, 43 pages.
Russian Office Action from Russian Application No. 2022108234 dated Dec. 19, 2023, 11 pages. With English Translation.
Japanese Office Action from Japanese Application No. 2022-515999 dated Feb. 21, 2023, 10 pages. With English Translation.
Indian Office Action from Indian Application No. 202247013800 dated Jul. 21, 2022, 7 pages.
Document: JVET-P2001-vE, Bross, B., et al., "Versatile Video Coding (Draft 7)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 491 pages.
Suehring, K., et al., Retrieved from the Internet: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-7.0, Sep. 25, 2022, 2 pages.
Document: JVET-M0413, Said, A., et al., "CE5: Per-context CABAC initialization with single window (Test 5.1.4)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, Jan. 9-18, 2019, 3 pages.
"Series H: Audiovisual and Multimedia Systems Infrastructure of audiovisual services—Coding of moving video High efficiency video coding," ITU-T and ISO/IEC, "High efficiency video coding", Rec. ITU-T H.265 | ISO/IEC 23008-2, Feb. 2018, 692 pages.
Document: JCTVC-Y1002, Rosewarne, C., et al., "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 7," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 25th Meeting: Chengdu, CN, Oct. 14-21, 2016, 70 pages.
Document: JVET-G1001, Chen, J., et al., "Algorithm description of JointExploration Test Model 7 (JEM7)," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 7th Meeting: Torino, IT, Jul. 13-21, 2017, 50 pages.
JEM-7.0 Retrieved from the internet: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/ HM-16.6-JEM-7.0. Sep. 19, 2022, 1 page.
Document: JVET-O2001, Bross, B., et al., "Versatile Video Coding (Draft 6)", Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, 455 pages.
Document: JVET-Q0041-v2, Wang, Y.K., et al. "AHG2: Editorial input on VVC draft text," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 2 pages.
Document: JVET-Q0513_r2, Xiu, X., et al., "AHG16: Clipping residual samples for ACT," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 11 pages.
Document: JVET-Q0820-v3, Lagrange P., et al., "ACT: common text for bug fixes and transform change," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 7 pages.
Document: JVET-L0553, Li, X., et al., "Fix of Initial QP Signaling," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, Oct. 3-12, 2018, 2 pages.
Document: JVET-N0368, Zhao, X., et al., "An implementation of adaptive color transform in VVC,"Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, Mar. 19-27, 2019, 3 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Document: JVET-O0166, Clare, G., et al., "CE8-related: BDPCM for chroma," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, Jul. 3-12, 2019, 5 pages.
Document: JVET-P0517-R1, Xiu, X., et al., "Support of adaptive color transform for 444 video coding in VVC," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 4 pages.
Document: JVET-Q0305-v2, Dou, L., et al., "Disallowing JCCR mode for ACT coded CUs," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 4 pages.
Document: JVET-Q0423-V1, Li, L., et al., "Interaction between ACT and BDPCM chroma," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/[EC JTC I/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 3 pages.
Document: JVET-Q0521-V1, Zhu, et al., "Alignment of BDPCM for ACT," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 6 pages.
Document: JVET-QO520-v1, Wang, Y., et al., "AHG9: Cleanups on signaling for CC-ALF, BDPCM, ACT and Palette," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 6 pages.
Document: JVET-R0058-v5, Wang, Y-K., et al., "AHG8/AHG9/AHG12: On the combination of RPR, subpictures, and scalability," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 18th Meeting: by teleconference, Apr. 15-24, 2020, 10 pages.
Document: JVET-S0105 McCarthy, S., et al., "AHG9: Modification of general constraint information," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC I/SC 29/WG 11 19th Meeting: by teleconference, Jun. 22-Jul. 1, 2020, 16 pages.
Document: JVET-P1001, Kotra, A., et al., "Non-CE5: Chroma QP derivation fix for deblocking filter (Combination of JVET-P0105 and JVET-P0539)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/ WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 7 pages.
Document: JVET-P2002, Chen, J., et al., "Algorithm description for Versatile Video Coding and Test Model 7 (VTM 7)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 90 pages.
Document: JVET-Q0420-V1, Li, L., et al., "AHG12: Signaling of chroma presence in PPS and APS," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 3 pages.
Document: JVET-Q0815, Zhang, L., et al., "BoG on ACT," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/[EC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 3 pages.
Document: JVET-Q0241-v3, Jung, J., et al., "On QP adjustment in adaptive color transform," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/[EC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 5 pages.
Document: JVET-Q0114-v1, Wang, Y.K., et al., "AHG9: A few more general contraints flags," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 3 pages.
Document: JVET-R0286, Chang, Y., et al., "AhG9: On general constraint information syntax," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 18th Meeting: by teleconference: Apr. 15-24, 2020, 5 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2020/137941, English Translation of International Search Report dated Mar. 22, 2021, 10 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2020/137946, English Translation of International Search Report dated Mar. 22, 2021, 9 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/070265, English Translation of International Search Report dated Mar. 26, 2021, 14 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/070279, English Translation of International Search Report dated Apr. 1, 2021, 11 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/070282, English Translation of International Search Report dated Mar. 23, 2021, 13 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/070580, English Translation of International Search Report dated Mar. 22, 2021, 12 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/071659, English Translation of International Search Report dated Apr. 6, 2021, 14 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/071660, English Translation of International Search Report dated Mar. 31, 2021, 11 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/072017, English Translation of International Search Report dated Apr. 19, 2021, 14 pages.
Foreign Communication From A Related Counterpart Application, International Application No. PCT/CN2021/072396, English Translation of International Search Report dated Mar. 29, 2021, 14 pages.
Office Action dated Sep. 13, 2022, 11 pages, U.S. Appl. No. 17/842,025, filed Jun. 16, 2022.
Extended European Search Report from European Application No. 21736195.5 dated Mar. 15, 2023, 14 pages.
Final Office Action from U.S. Appl. No. 17/842,025 dated Jan. 13, 2023, 23 pages.
Notice of Allowance from U.S. Appl. No. 17/857,874 dated Jul. 6, 2023, 9 pages.
Non-Final Office Action from U.S. Appl. No. 17/857,924 dated Feb. 3, 2023, 23 pages.
Notice of Allowance from U.S. Appl. No. 17/842,025 dated Aug. 10, 2023, 14 pages.
Notice of Allowance from U.S. Appl. No. 17/857,874 dated Jan. 26, 2023, 23 pages.
Non-Final Office Action from U.S. Appl. No. 17/856,631 dated Sep. 26, 2024, 26 pages.
Non-Final Office Action from U.S. Appl. No. 17/810,187 dated Aug. 1, 2024, 34 pages.
Non-Final Office Action from U.S. Appl. No. 18/498,665 dated Sep. 26, 2024, 106 pages.
Non-Final Office Action from U.S. Appl. No. 18/519,994 dated Aug. 14, 2024, 125 pages.
Vietnamese Office Action from Vietnamese Patent Application No. 1-2022-01641 dated Aug. 30, 2024, 4 pages.
Non-Final Office Action from U.S. Appl. No. 17/864,011 dated Apr. 25, 2024, 38 pages.
Non-Final Office Action from U.S. Appl. No. 18/524,994 dated Jun. 21, 2024, 56 pages.
Final Office Action from U.S. Appl. No. 17/856,631 dated Jun. 4, 2024, 45 pages.
Document JVET-N0276, Jill Boyce, AHG15: On interoperability point signalling, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, Mar. 19-27, 2019, 16 pages.
Chinese Office Action from Chinese Patent Application No. 202180008254.6 dated May 23, 2025, 10 pages.

* cited by examiner

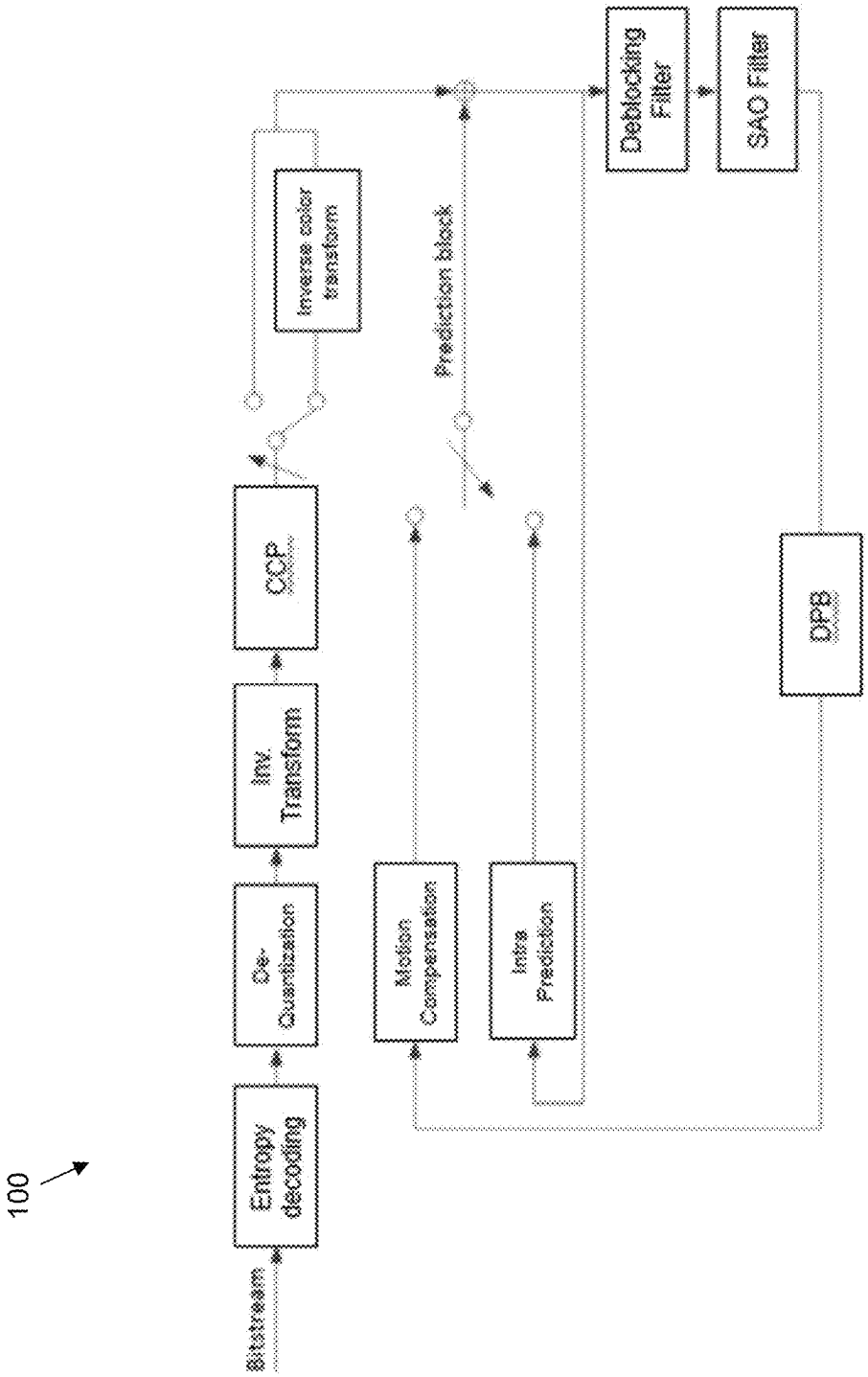

100
Bitstream
Entropy decoding
De-Quantization
Inv. Transform
CCP
Inverse color transform
Prediction block
Motion Compensation
Intra Prediction
Deblocking Filter
SAO Filter
DPB

FIG. 1

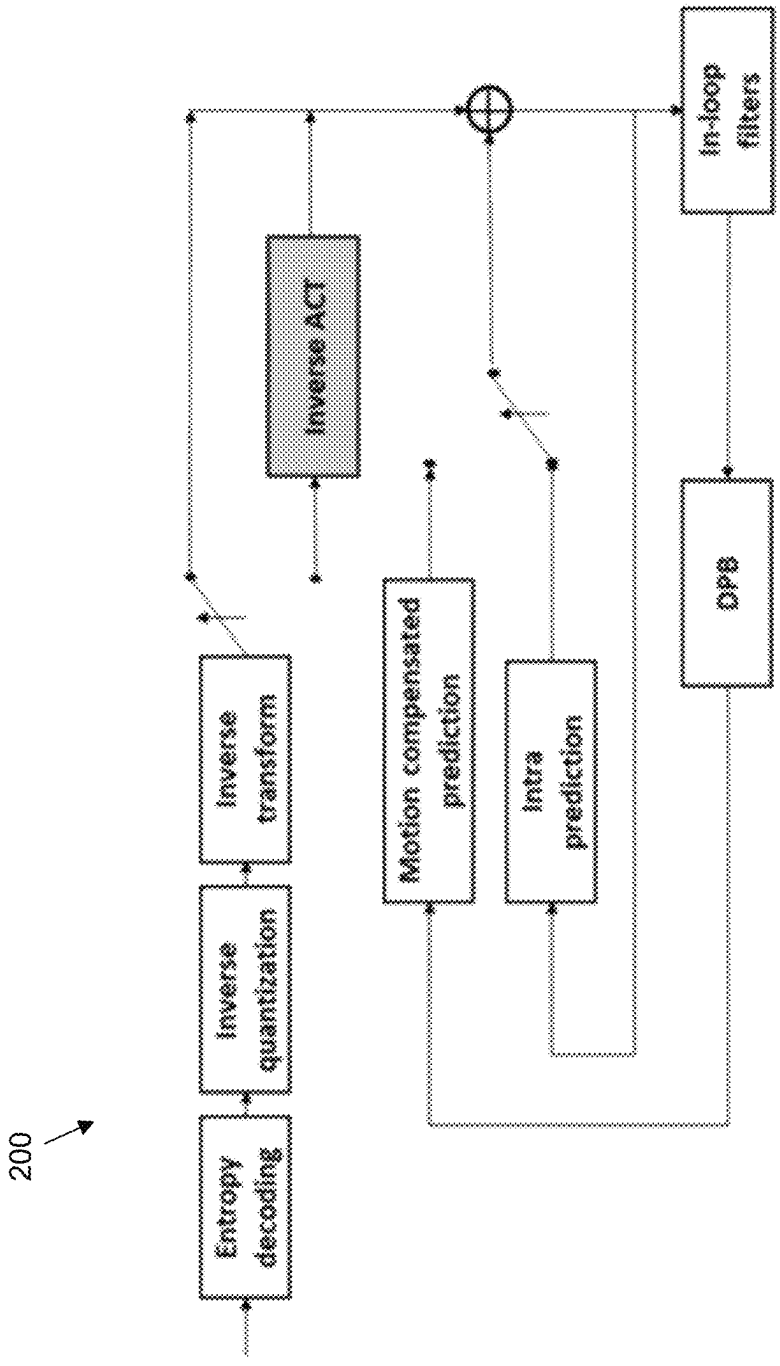

200
Entropy decoding
Inverse quantization
Inverse transform
Inverse ACT
Motion compensated prediction
Intra prediction
DPB
In-loop filters

FIG. 2

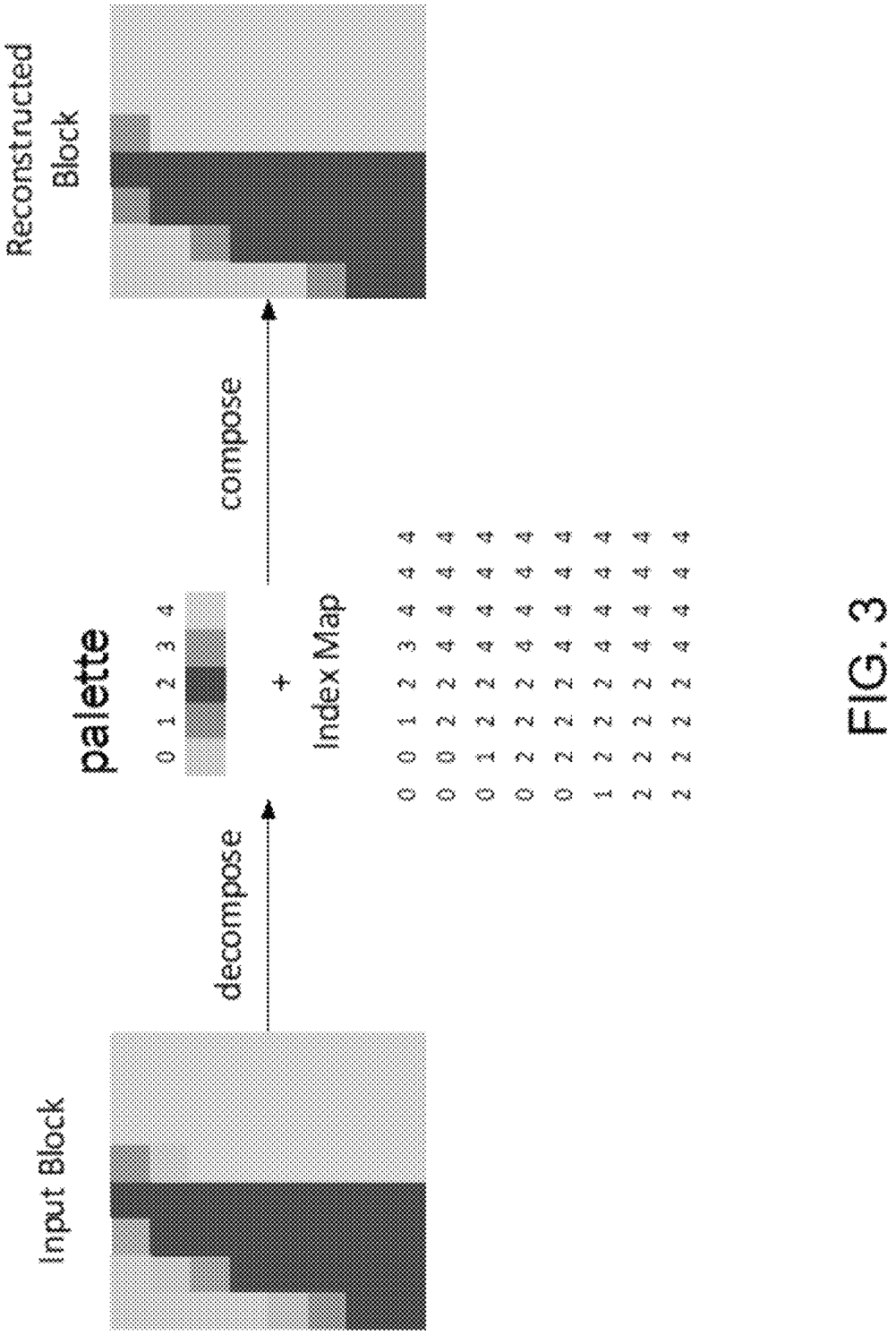

Reconstructed Block
compose
palette
0 1 2 3 4
+
Index Map
0 0 1 2 3 4 4 4 4
0 0 2 2 4 4 4 4 4
0 1 2 2 4 4 4 4 4
0 2 2 2 4 4 4 4 4
0 2 2 2 4 4 4 4 4
1 2 2 2 4 4 4 4 4
2 2 2 2 4 4 4 4 4
2 2 2 2 4 4 4 4 4
decompose
Input Block

FIG. 3

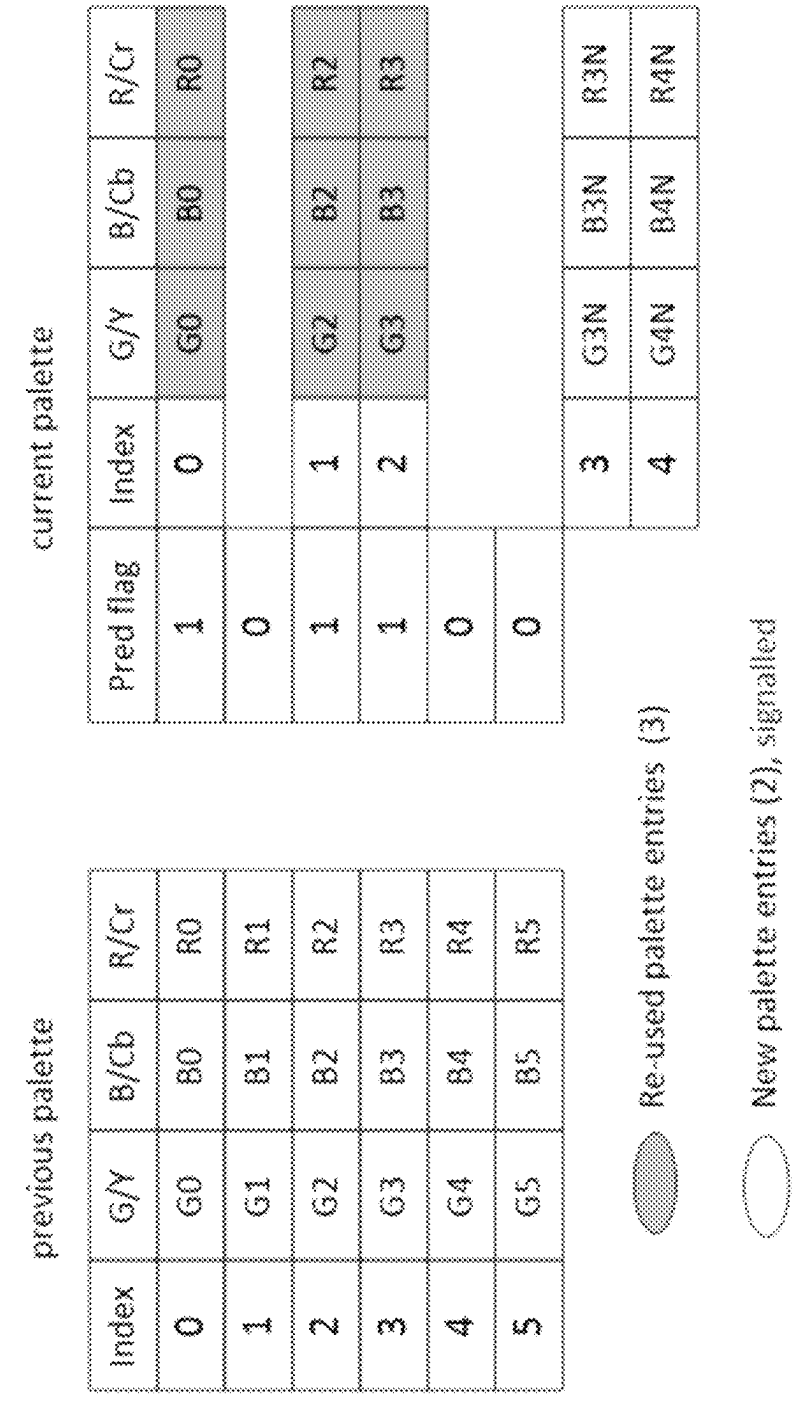
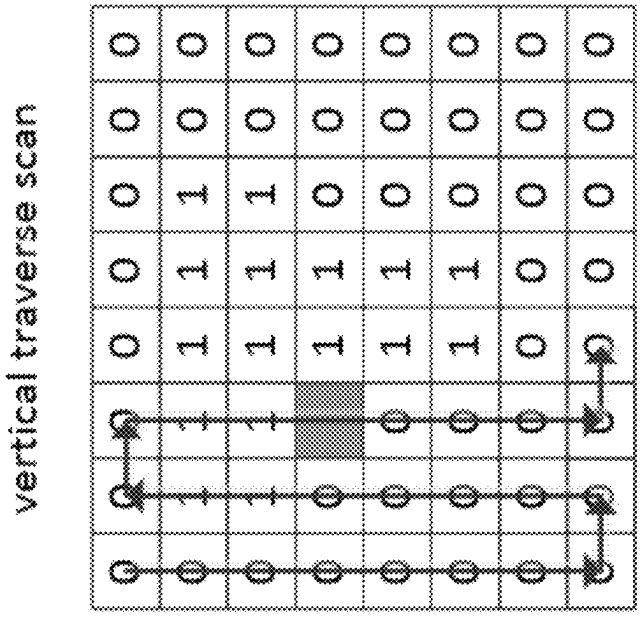

previous palette
Index
G/Y
B/Cb
R/Cr
0
G0
B0
R0
1
G1
B1
R1
2
G2
B2
R2
3
G3
B3
R3
4
G4
B4
R4
5
G5
B5
R5
current palette
Pred flag
Index
G/Y
B/Cb
R/Cr
1
0
G0
B0
R0
0
1
G2
B2
R2
1
2
G3
B3
R3
0
0
0
3
G3N
B3N
R3N
4
G4N
B4N
R4N
Re-used palette entries (3)
New palette entries (2), signalled

FIG. 4



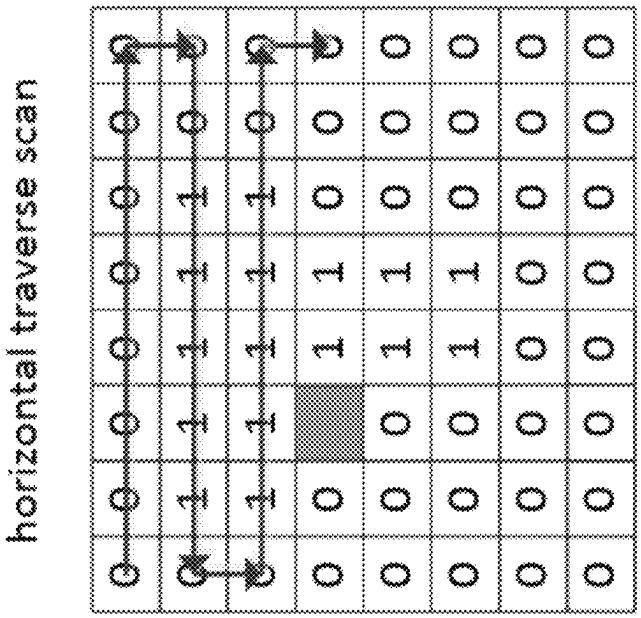


FIG. 5

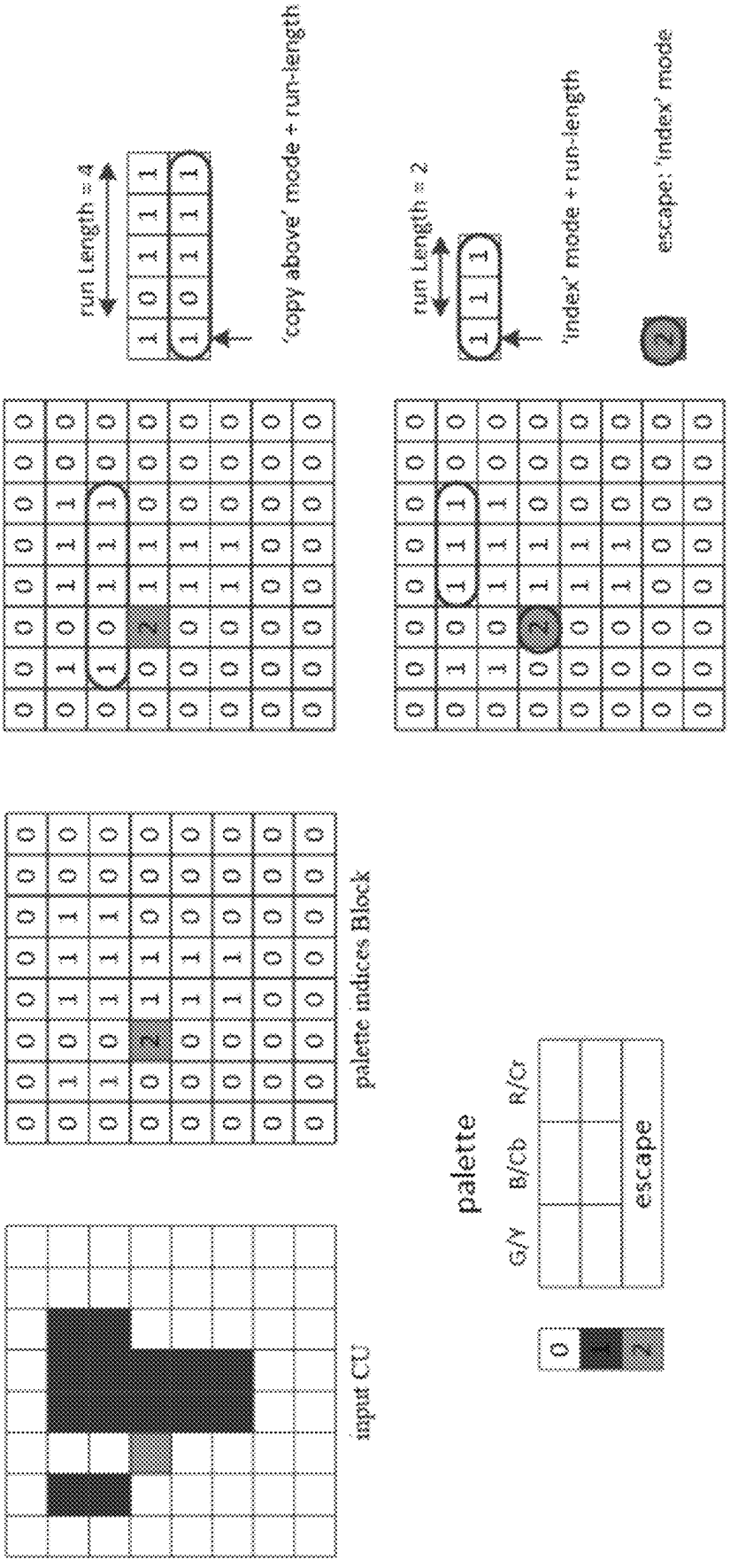

input CU
palette indices Block
palette
G/Y
B/Cb
R/Cr
escape
'index' mode + run-length
run Length = 2
escape: 'index' mode
'copy above' mode + run-length
run Length = 4

FIG. 6

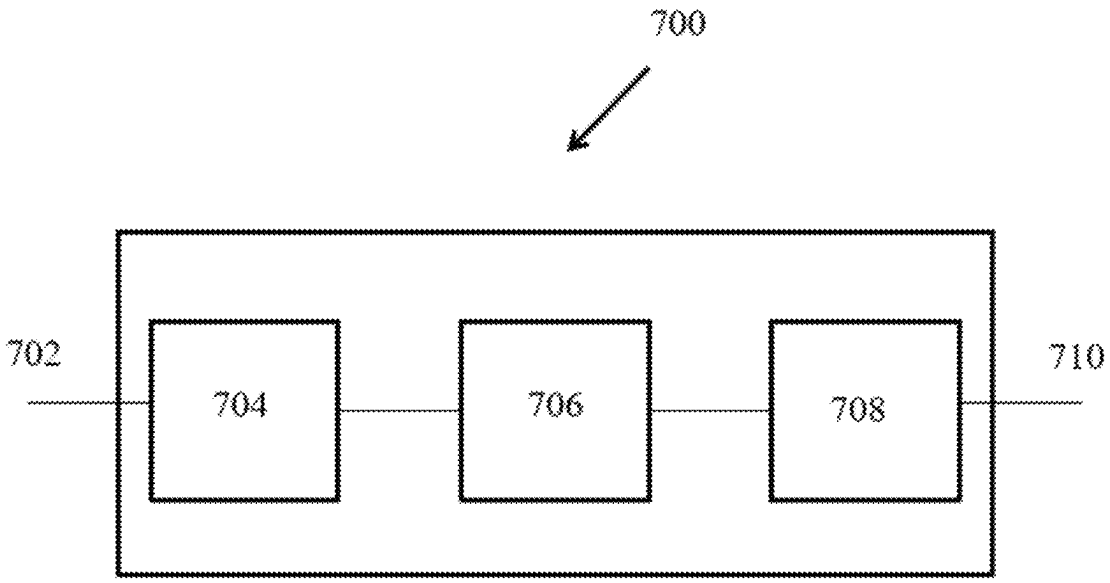


FIG. 7

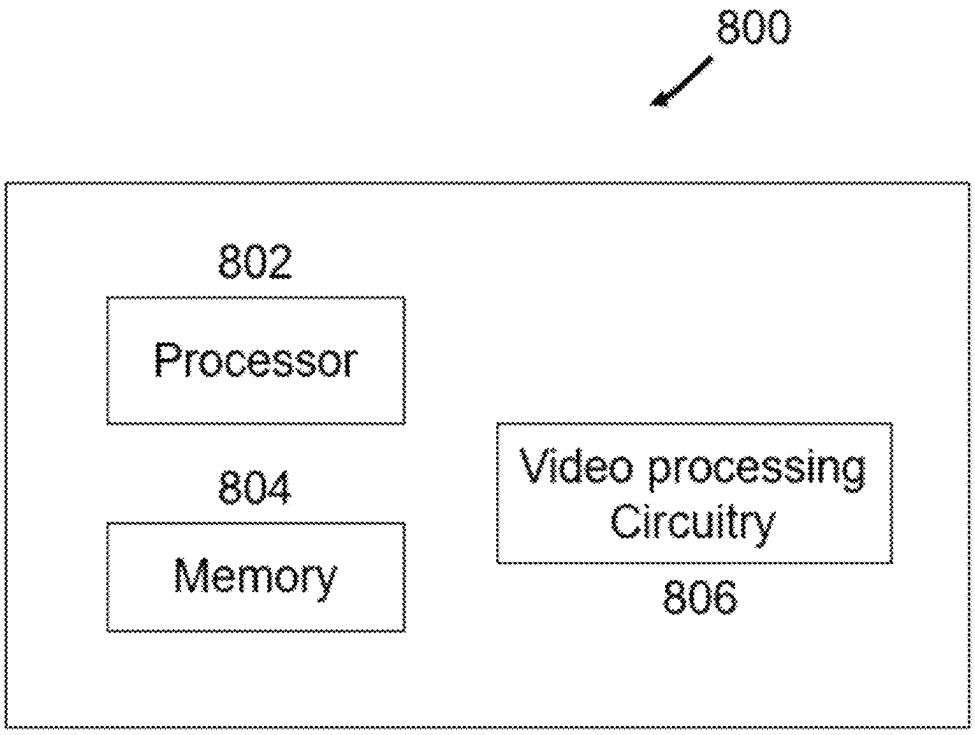

800
802
Processor
804
Memory
Video processing Circuitry
806

FIG. 8

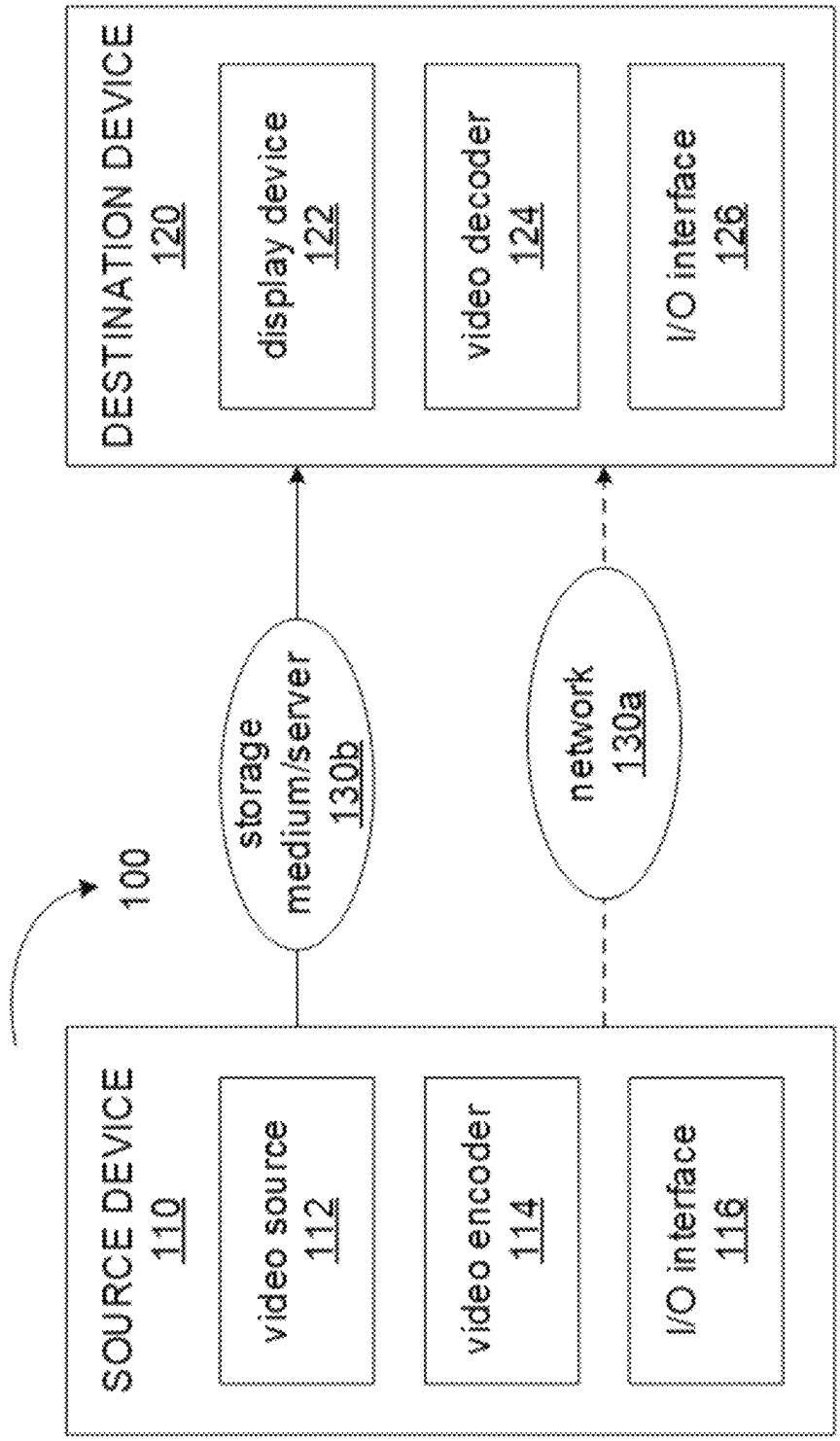

SOURCE DEVICE
110
video source
112
video encoder
114
I/O interface
116
100
storage medium/server
130b
network
130a
DESTINATION DEVICE
120
display device
122
video decoder
124
I/O interface
126

FIG. 9

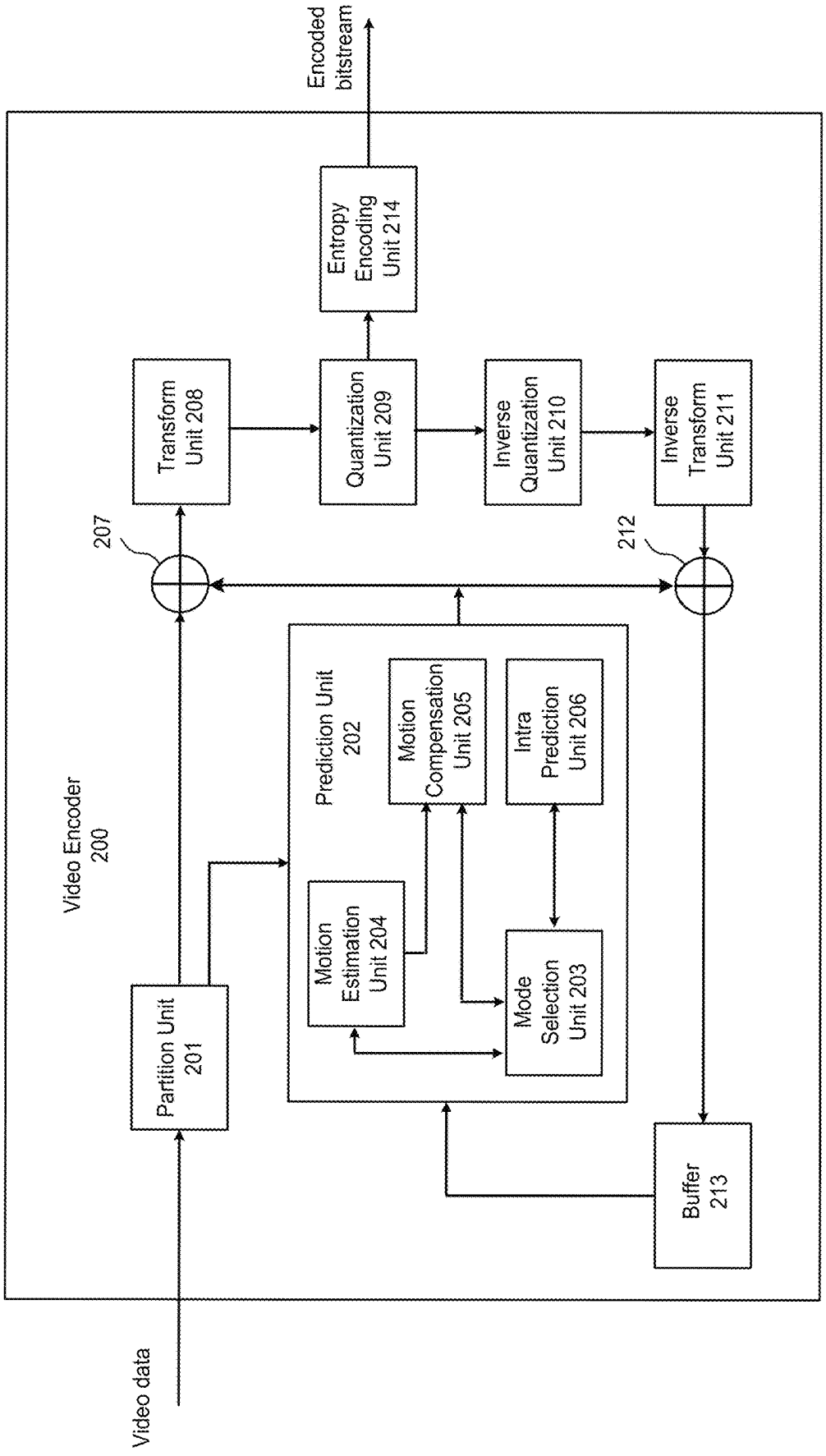

Encoded bitstream
Entropy Encoding Unit 214
Transform Unit 208
Quantization Unit 209
Inverse Quantization Unit 210
Inverse Transform Unit 211
207
212
Video Encoder 200
Prediction Unit 202
Motion Compensation Unit 205
Intra Prediction Unit 206
Motion Estimation Unit 204
Mode Selection Unit 203
Partition Unit 201
Buffer 213
Video data

FIG. 10

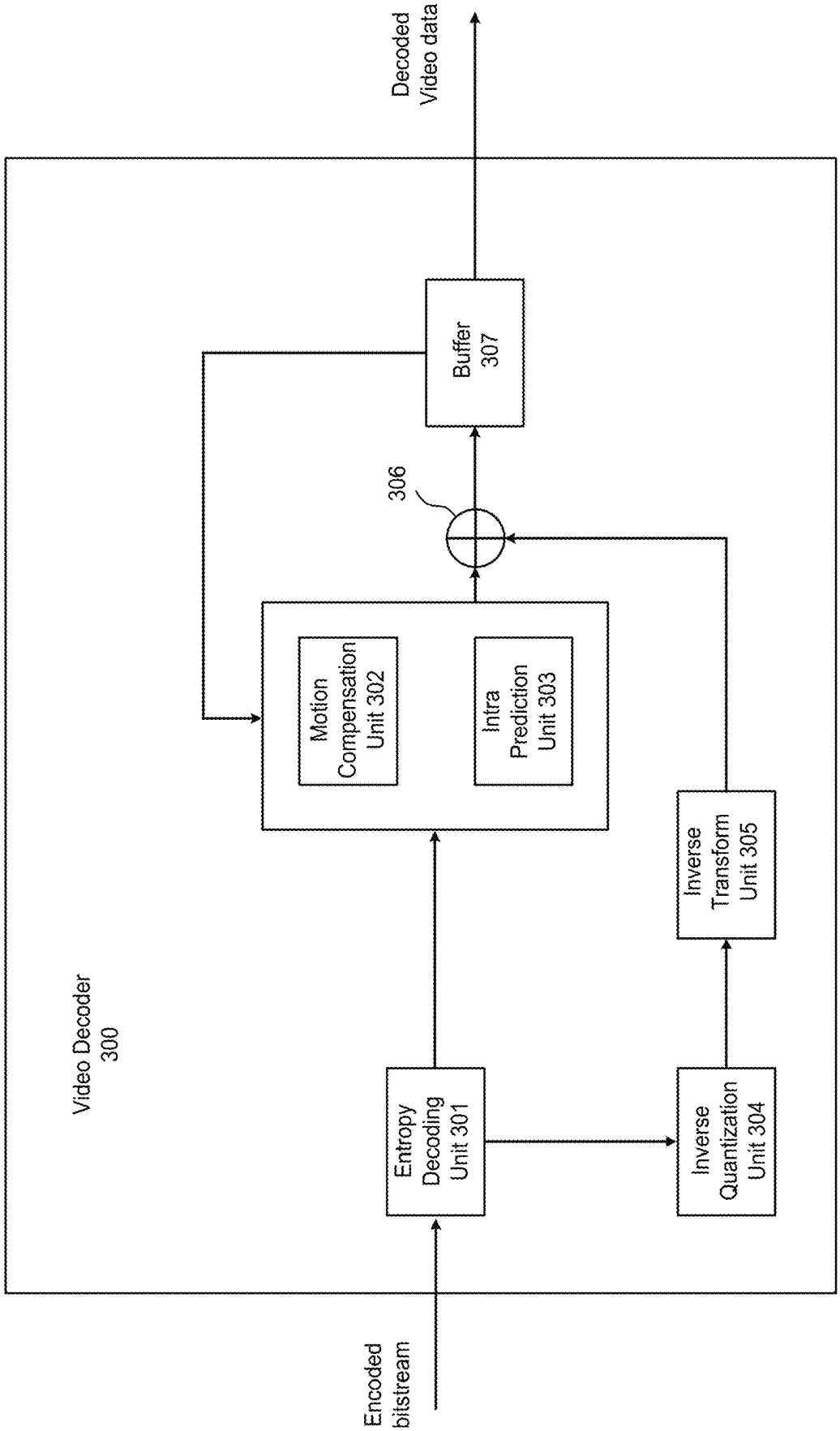

Decoded Video data
Buffer 307
306
Motion Compensation Unit 302
Intra Prediction Unit 303
Inverse Transform Unit 305
Video Decoder 300
Entropy Decoding Unit 301
Inverse Quantization Unit 304
Encoded bitstream

FIG. 11

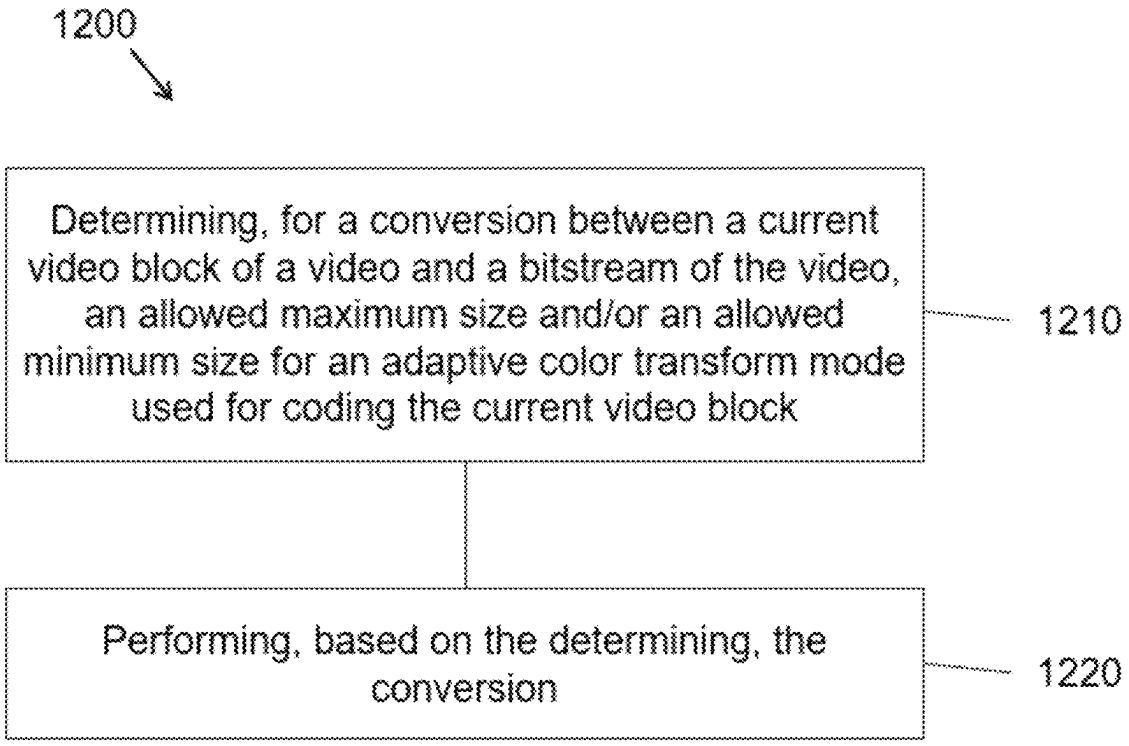

1200
Determining, for a conversion between a current video block of a video and a bitstream of the video, an allowed maximum size and/or an allowed minimum size for an adaptive color transform mode used for coding the current video block
1210
Performing, based on the determining, the conversion
1220

FIG. 12

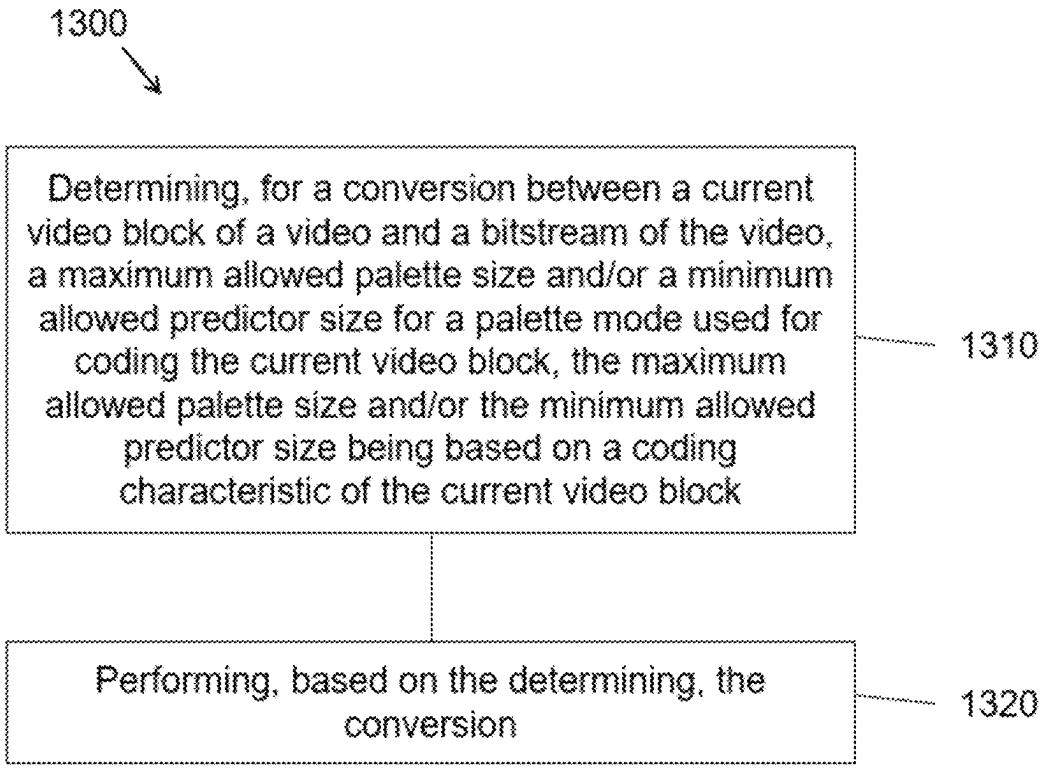

1300
Determining, for a conversion between a current video block of a video and a bitstream of the video, a maximum allowed palette size and/or a minimum allowed predictor size for a palette mode used for coding the current video block, the maximum allowed palette size and/or the minimum allowed predictor size being based on a coding characteristic of the current video block
1310
Performing, based on the determining, the conversion
1320

FIG. 13

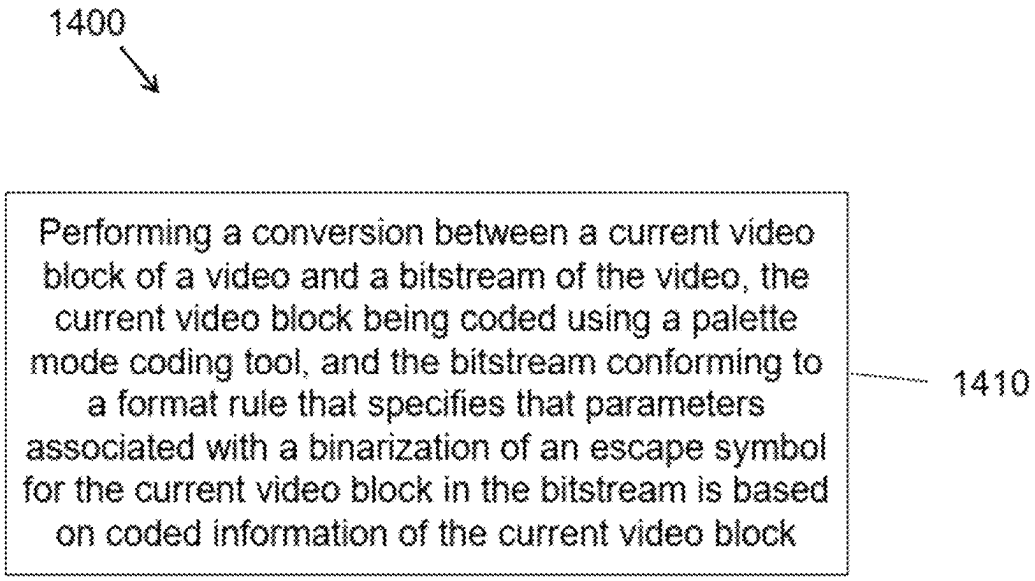

1400
Performing a conversion between a current video block of a video and a bitstream of the video, the current video block being coded using a palette mode coding tool, and the bitstream conforming to a format rule that specifies that parameters associated with a binarization of an escape symbol for the current video block in the bitstream is based on coded information of the current video block
1410

FIG. 14

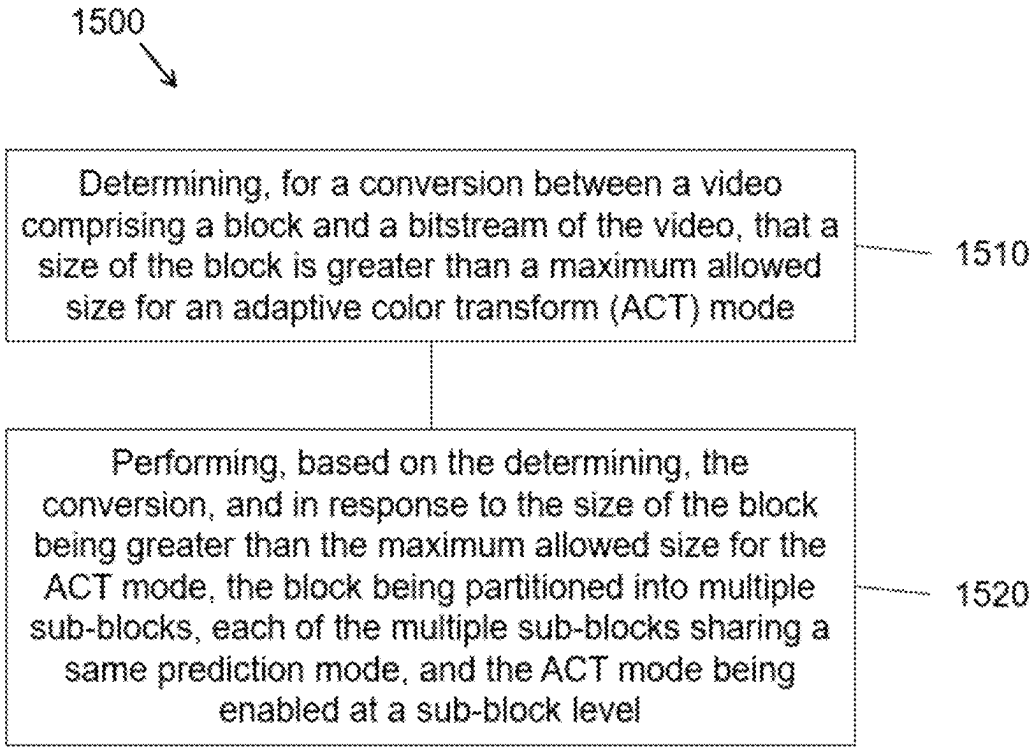

1500
Determining, for a conversion between a video comprising a block and a bitstream of the video, that a size of the block is greater than a maximum allowed size for an adaptive color transform (ACT) mode
1510
Performing, based on the determining, the conversion, and in response to the size of the block being greater than the maximum allowed size for the ACT mode, the block being partitioned into multiple sub-blocks, each of the multiple sub-blocks sharing a same prediction mode, and the ACT mode being enabled at a sub-block level
1520

FIG. 15

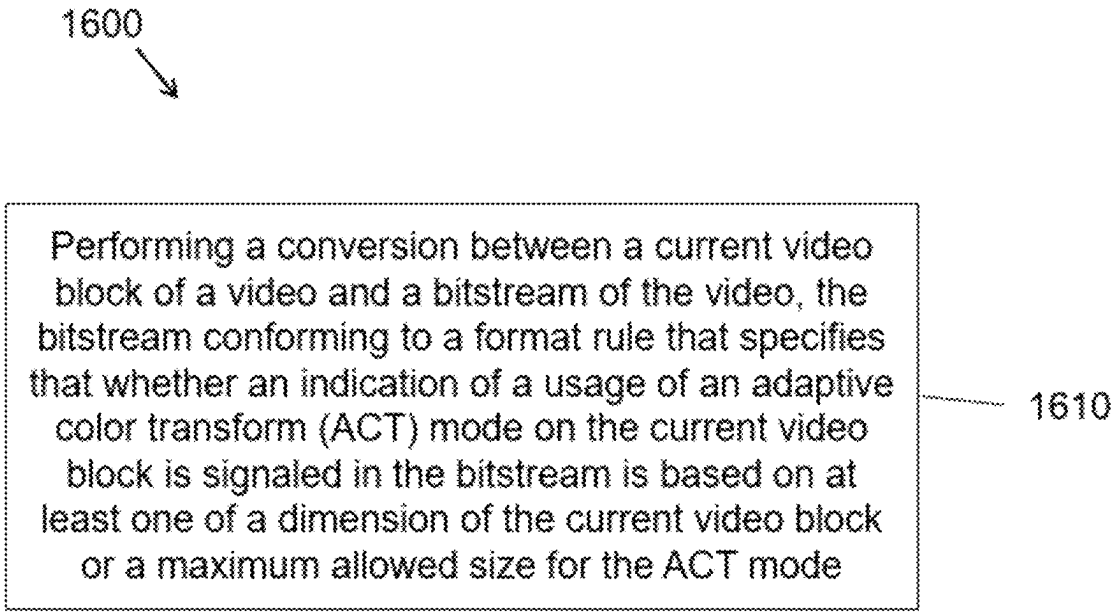

1600
Performing a conversion between a current video block of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies that whether an indication of a usage of an adaptive color transform (ACT) mode on the current video block is signaled in the bitstream is based on at least one of a dimension of the current video block or a maximum allowed size for the ACT mode
1610

FIG. 16

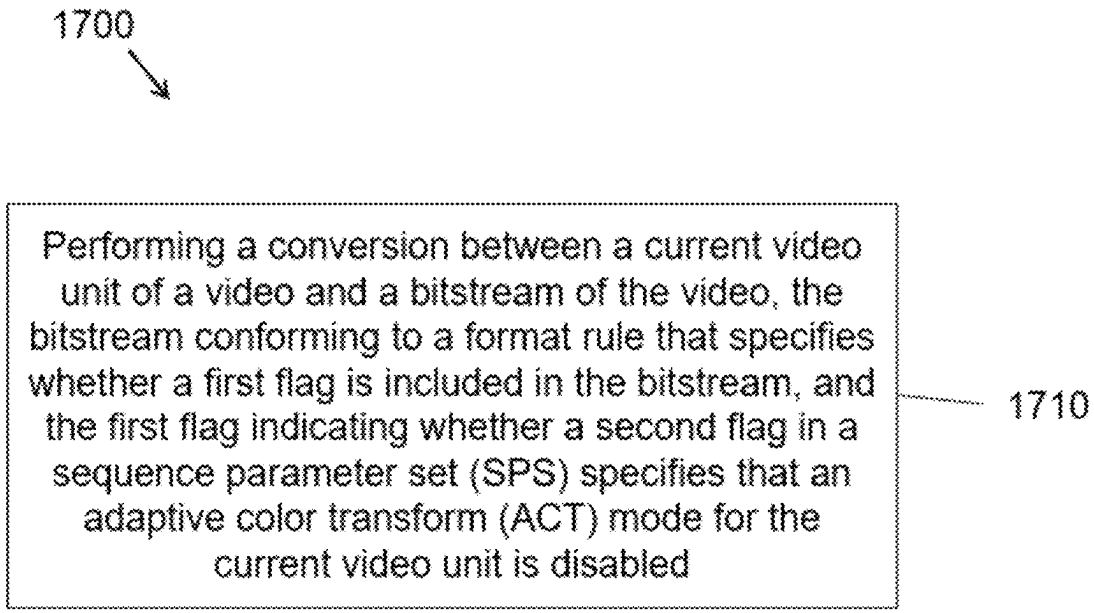

1700
Performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that an adaptive color transform (ACT) mode for the current video unit is disabled
1710

FIG. 17

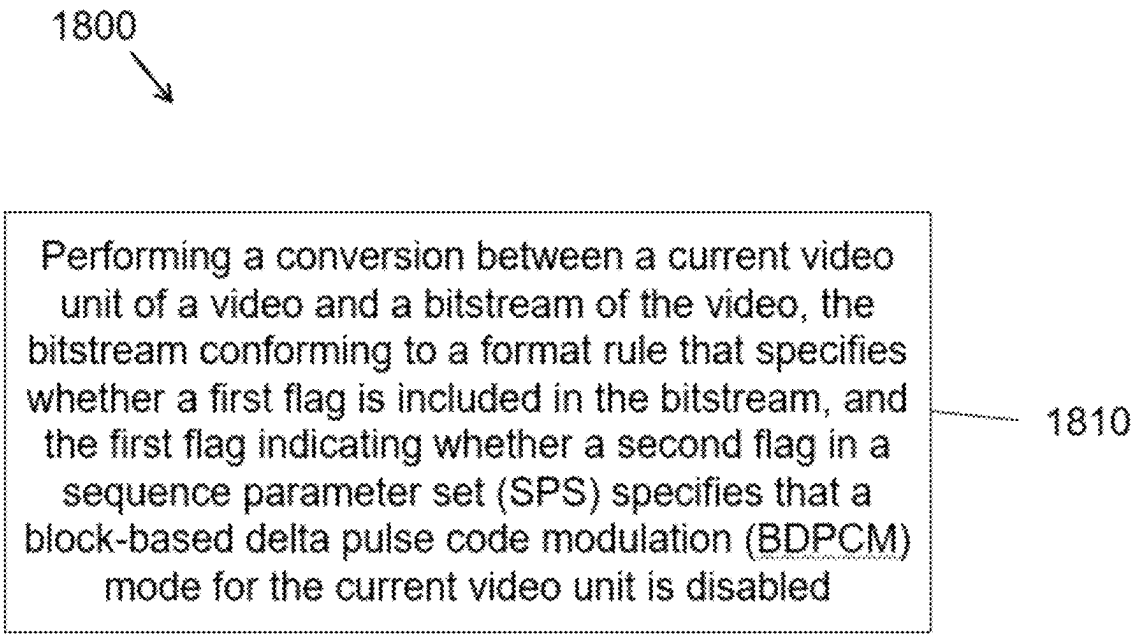

1800
Performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for the current video unit is disabled
1810

FIG. 18

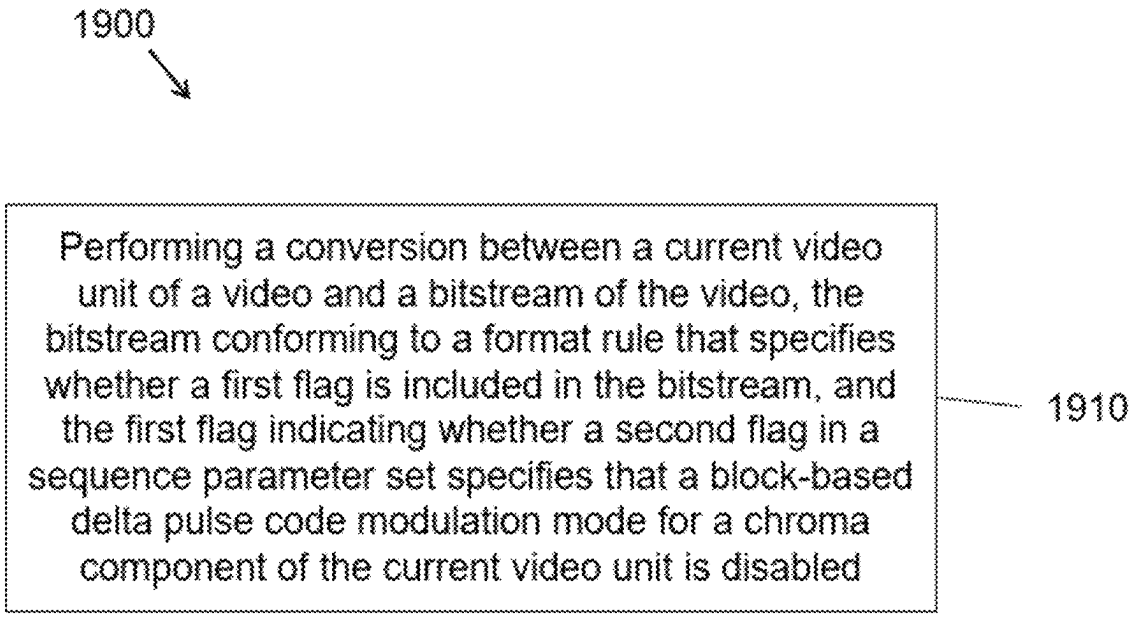

1900
Performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set specifies that a block-based delta pulse code modulation mode for a chroma component of the current video unit is disabled
1910

FIG. 19

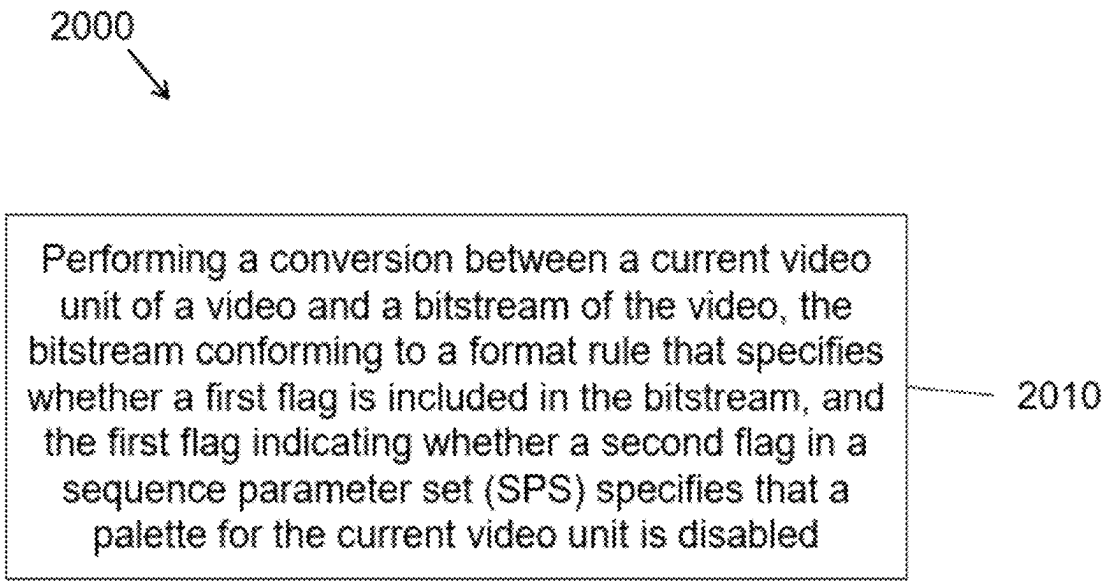

2000
Performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a palette for the current video unit is disabled
2010

FIG. 20

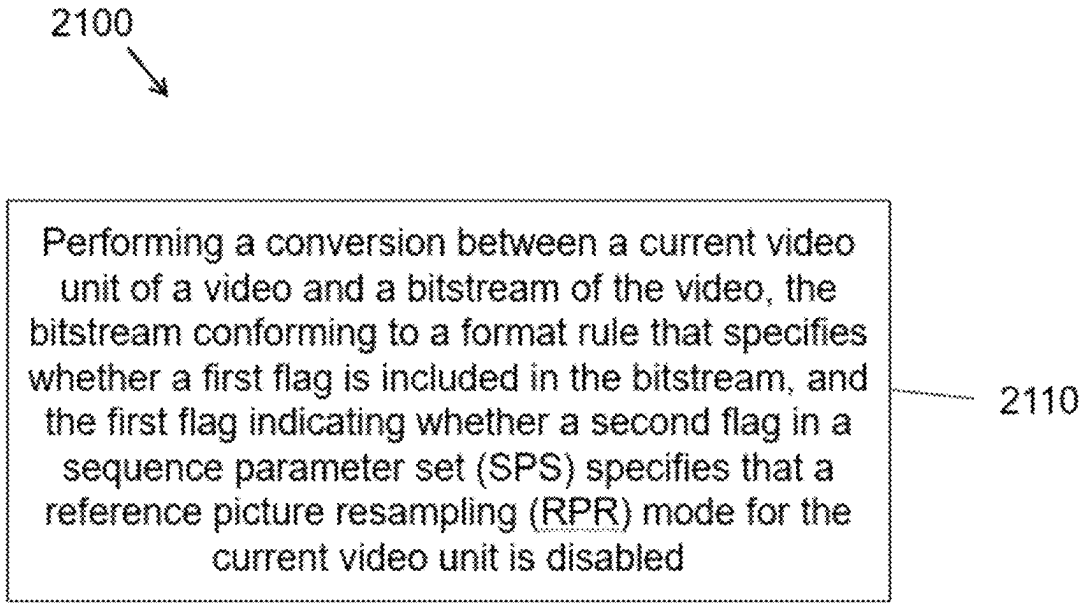

2100
Performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a reference picture resampling (RPR) mode for the current video unit is disabled
2110

FIG. 21

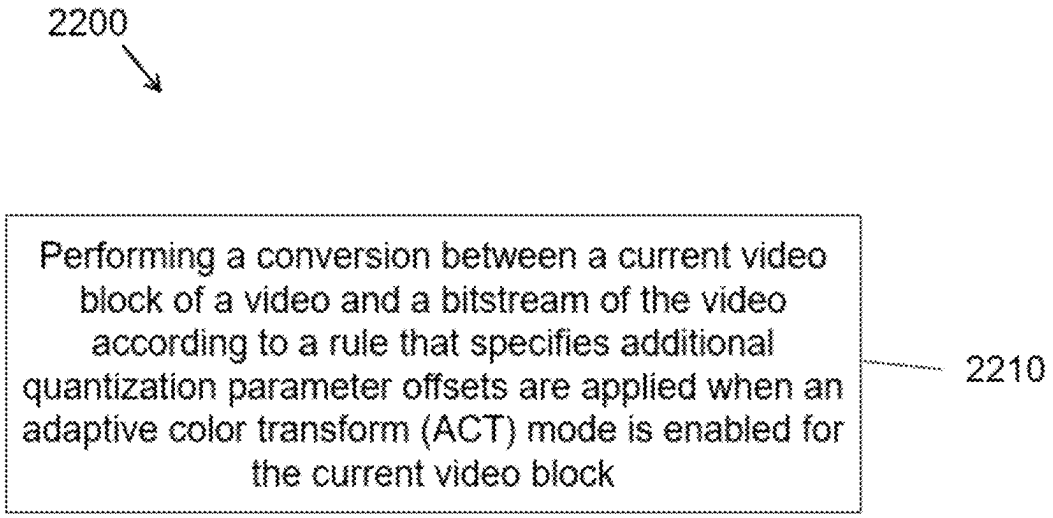

2200
Performing a conversion between a current video block of a video and a bitstream of the video according to a rule that specifies additional quantization parameter offsets are applied when an adaptive color transform (ACT) mode is enabled for the current video block
2210

FIG. 22

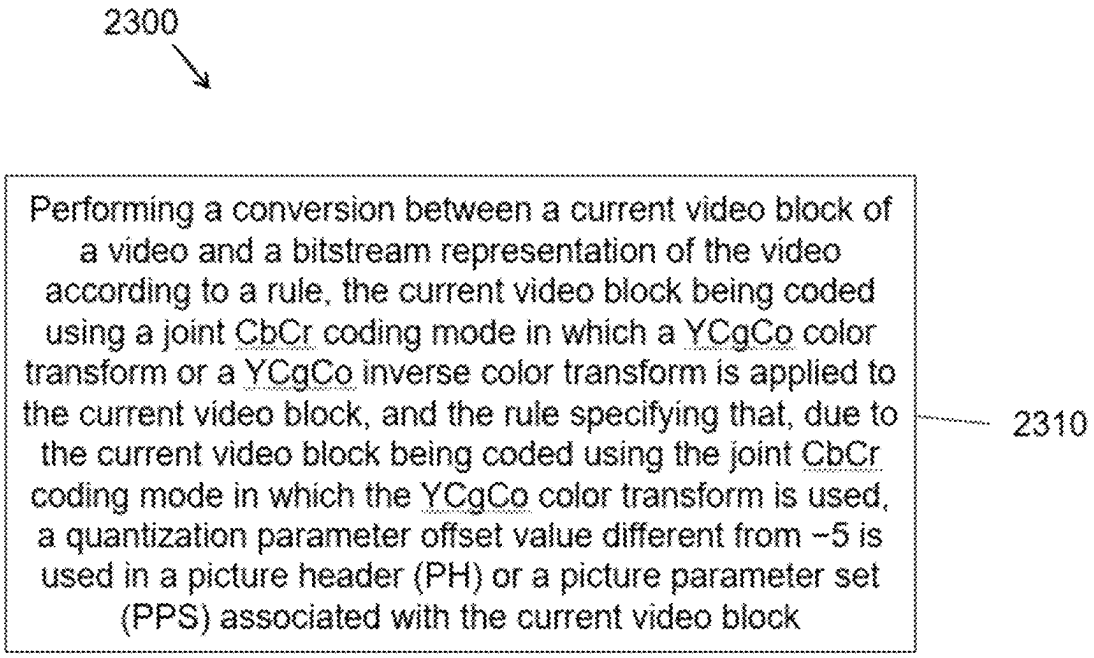

2300
Performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, the current video block being coded using a joint CbCr coding mode in which a YCgCo color transform or a YCgCo inverse color transform is applied to the current video block, and the rule specifying that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo color transform is used, a quantization parameter offset value different from −5 is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block
2310

FIG. 23

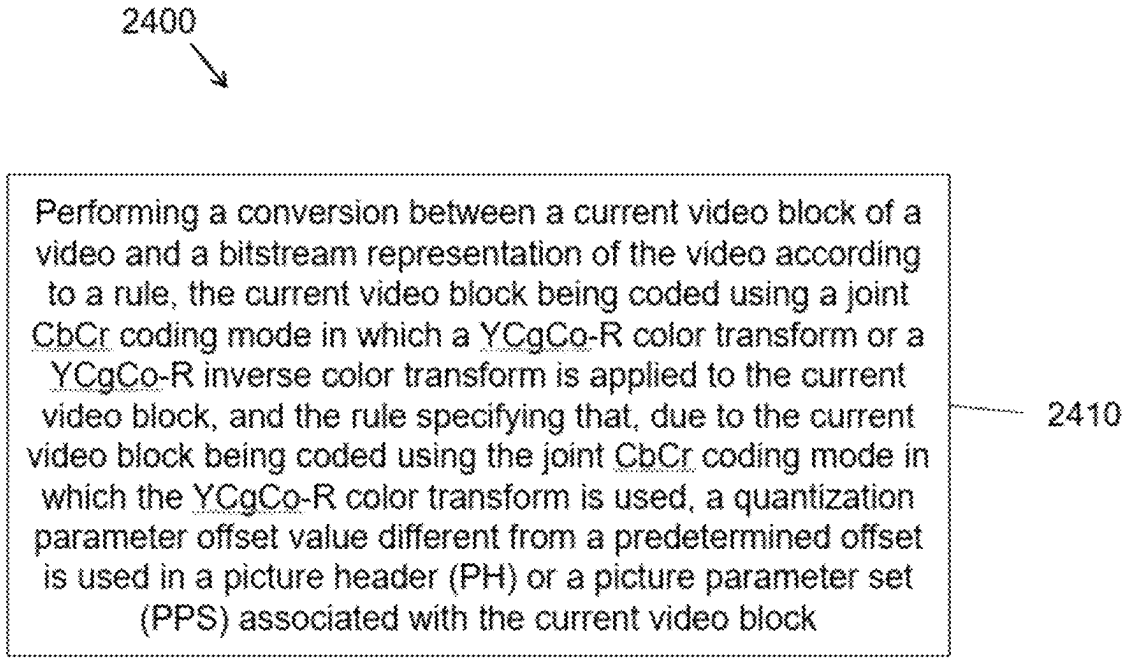

2400
Performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, the current video block being coded using a joint CbCr coding mode in which a YCgCo-R color transform or a YCgCo-R inverse color transform is applied to the current video block, and the rule specifying that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo-R color transform is used, a quantization parameter offset value different from a predetermined offset is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block
2410

FIG. 24

GENERAL CONSTRAINTS INFORMATION FOR VIDEO CODING

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 17/857,874, filed on Jul. 5, 2022, which is a continuation of International Patent Application No. PCT/CN2021/070279, filed on Jan. 5, 2021 which claims the priority to and benefits of International Patent Application No. PCT/CN2020/070368 filed on Jan. 5, 2020. All the aforementioned patent applications are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

The present disclosure relates to image and video coding and decoding.

BACKGROUND

Digital video accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

SUMMARY

The present disclosure discloses system, methods and devices for video encoding and decoding that use, in addition to other coding tools, the adaptive color transform (ACT) mode.

In one example aspect, a method of video processing is disclosed. The method includes determining, for a conversion between a current video block of a video and a bitstream of the video, an allowed maximum size and/or an allowed minimum size for an adaptive color transform (ACT) mode used for coding the current video block; and performing, based on the determining, the conversion.

In another example aspect, a method of video processing is disclosed. The method includes determining, for a conversion between a current video block of a video and a bitstream of the video, a maximum allowed palette size and/or a minimum allowed predictor size for a palette mode used for coding the current video block; and performing, based on the determining, the conversion, wherein the maximum allowed palette size and/or the minimum allowed predictor size is based on a coding characteristic of the current video block.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video block of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the current video block is coded using a palette mode coding tool, and wherein the format rule specifies that parameters associated with a binarization of an escape symbol for the current video block in the bitstream is based on coded information of the current video block.

In yet another example aspect, a method of video processing is disclosed. The method includes determining, for a conversion between a video comprising a block and a bitstream of the video, that a size of the block is greater than a maximum allowed size for an adaptive color transform (ACT) mode; and performing, based on the determining, the conversion, wherein, in response to the size of the block being greater than the maximum allowed size for the ACT mode, the block is partitioned into multiple sub-blocks, and wherein each of the multiple sub-blocks share a same prediction mode, and the ACT mode is enabled at a sub-block level.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video block of a video and a bitstream of the video, wherein the bitstream conforms to a format rule that specifies that whether an indication of a usage of an adaptive color transform (ACT) mode on the current video block is signaled in the bitstream is based on at least one of a dimension of the current video block or a maximum allowed size for the ACT mode.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that an adaptive color transform (ACT) mode for the current video unit is disabled.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for the current video unit is disabled.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for a chroma component of the current video unit is disabled.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a palette for the current video unit is disabled.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a reference picture resampling (RPR) mode for the current video unit is disabled.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video block of a video and a bitstream of the video according to a rule that specifies

additional quantization parameter offsets are applied when an adaptive color transform (ACT) mode is enabled for the current video block.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, wherein the current video block is coded using a joint CbCr coding mode in which a YCgCo color transform or a YCgCo inverse color transform is applied to the current video block, and wherein the rule that specifies that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo color transform is used, a quantization parameter offset value different from −5 is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

In yet another example aspect, a method of video processing is disclosed. The method includes performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, wherein the current video block is coded using a joint CbCr coding mode in which a YCgCo-R color transform or a YCgCo-R inverse color transform is applied to the current video block, and wherein the rule that specifies that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo-R color transform is used, a quantization parameter offset value different from a predetermined offset is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

In yet another example aspect, a video encoder apparatus is disclosed. The video encoder comprises a processor configured to implement the above-described methods.

In yet another example aspect, a video decoder apparatus is disclosed. The video decoder comprises a processor configured to implement the above-described methods.

In yet another example aspect, a non-transitory computer readable medium having code stored thereon is disclosed. The code embodies one of the methods described herein in the form of processor-executable code.

These, and other, features are described throughout the present disclosure.

BRIEF DESCRIPTION OF DRAWINGS

FIG. **1** shows a screen content coding (SCC) decoder flow of in-loop adaptive color transform (ACT).

FIG. **2** illustrates a decoding process with the ACT.

FIG. **3** shows an example of a block coded in palette mode.

FIG. **4** shows an example of using a palette predictor to signal palette entries.

FIG. **5** shows an example of horizontal and vertical traverse scans.

FIG. **6** shows an example of coding of palette indices.

FIG. **7** is a block diagram showing an example video processing system according to various embodiments of the disclosure.

FIG. **8** is a block diagram of an example hardware platform used for video processing.

FIG. **9** is a block diagram that illustrates a video coding system according to various embodiments of the disclosure.

FIG. **10** is a block diagram that illustrates an encoder according to various embodiments of the disclosure.

FIG. **11** is a block diagram that illustrates a decoder according to various embodiments of the disclosure.

FIGS. **12-24** show flowcharts for example methods of video processing.

DETAILED DESCRIPTION

Section headings are used in the present disclosure for ease of understanding and do not limit the applicability of embodiments disclosed in each section only to that section. Furthermore, H.266 terminology is used in some description only for ease of understanding and not for limiting scope of the disclosed embodiments. As such, the embodiments described herein are applicable to other video codec protocols and designs also.

1. Summary

The present disclosure is related to image/video coding technologies. Specifically, it is related to adaptive color transform in image/video coding. It may be applied to the standard under development, e.g., Versatile Video Coding (VVC). It may be also applicable to future video coding standards or video codec.

2. Brief Description

Video coding standards have evolved primarily through the development of the well-known International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by video coding experts group (VCEG) and moving pictures experts group (MPEG) jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

The latest version of VVC draft, i.e., Versatile Video Coding (Draft 7) could be found at: http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/16_Geneva/wg11/JVET-P2001-v14.zip.

The latest reference software of VVC, named VTM, could be found at: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-7.0.

2.1. Adaptive Color Transform (ACT) in HEVC-SCC

The Adaptive Color Transform (ACT) was adopted into the HEVC Screen Content Coding (SCC) test model 2 at the 18th JCT-VC meeting (Jun. 30 to Jul. 9, 2014, Sapporo, Japan). ACT performs in-loop color space conversion in the prediction residual domain using color transform matrices based on the YCoCg and YCoCg-R color spaces. ACT is turned on or off adaptively at the CU level using the flag cu_residual_act_flag. ACT can be combined with Cross Component Prediction (CCP), which is another inter com-

ponent de-correlation method already supported in HEVC. When both are enabled, ACT is performed after CCP at the decoder, as shown in the system **100** of FIG. **1**.

2.1.1. Color Space Conversion in ACT

The color space conversion in ACT is based on the YCoCg-R transform. Both lossy coding and lossless coding (cu_transquant_bypass_flag=0 or 1) use the same inverse transform, but an additional 1-bit left shift is applied to the Co and Cg components in the case of lossy coding. Specifically, the following color space transforms are used for forward and backward conversion for lossy and lossless coding:

Forward Transform for Lossy Coding (Non-Normative):

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} /4$$

Forward Transform for Lossless Coding (Non-Normative):

$Co=R-B$

$t=B+(Co>>1)$

$Cg=(G-t)$

$Y=t+(Cg>>1)$

Backward Transform (Normative):

```
if (lossy){
   Co = Co << 1
   Cg = Cg << 1
}
t = Y − (Cg >> 1)
G = Cg + t
B = t − (Co >> 1)
R = Co + b
```

The forward color transform is not normalized, with its norm being roughly equal to $\sqrt{6}/4$ for Y and Cg and equal to $\sqrt{2}/2$ for Co. In order to compensate for the non-normalized nature of the forward transform, delta QPs of (−5, −3, −5) are applied to (Y, Co, Cg), respectively. In other words, for a given "normal" QP for the CU, if ACT is turned on, then the quantization parameter is set equal to (QP−5, QP−3, QP−5) for (Y, Co, Cg), respectively. The adjusted quantization parameter only affects the quantization and inverse quantization of the residuals in the CU. For deblocking, the "normal" QP value is still used. Clipping to 0 is applied to the adjusted QP values to ensure that they will not become negative. Note that this QP adjustment is only applicable to lossy coding, as quantization is not performed in lossless coding (cu_transquant_bypass_flag=1). In SCM 4, PPS/slice-level signalling of additional QP offset values is introduced. These QP offset values may be used instead of (−5, −3, −5) for CUs when adaptive color transform is applied.

When the input bit-depths of the color components are different, appropriate left shifts are applied to align the sample bit-depths to the maximal bit-depth during ACT, and appropriate right shifts are applied to restore the original sample bit-depths after ACT.

2.2. ACT in VVC

FIG. **2** depicts a flowchart **200** of VVC with the ACT being applied. As illustrated in FIG. **2**, the color space conversion is carried out in residual domain. Specifically, one additional decoding module, namely inverse ACT, is introduced after inverse transform to convert the residuals from YCgCo domain back to the original domain.

In the VVC, unless the maximum transform size is smaller than the width or height of one coding unit (CU), one CU leaf node is also used as the unit of transform processing. Therefore, in the proposed implementation, the ACT flag is signaled for one CU to select the color space for coding its residuals. Additionally, following the HEVC ACT design, for inter and intra block copy (IBC) CUs, the ACT is only enabled when there is at least one non-zero coefficient in the CU. For intra CUs, the ACT is only enabled when chroma components select the same intra prediction mode of luma component, i.e., derived mode (DM).

The core transforms used for the color space conversions are kept the same as that used for the HEVC. Additionally, same with the ACT design in HEVC, to compensate the dynamic range change of residuals signals before and after color transform, the QP adjustments of (−5, −5, −3) are applied to the transform residuals.

On the other hand, the forward and inverse color transforms need to access the residuals of all three components. Correspondingly, in the proposed implementation, the ACT is disabled in the following two scenarios where not all residuals of three components are available.

1. Separate-tree partition: when separate-tree is applied, luma and chroma samples inside one coding tree unit (CTU) are partitioned by different structures. This results in that the CUs in the luma-tree only contains luma component and the CUs in the chroma-tree only contains two chroma components.
2. Intra sub-partition prediction (ISP): the ISP sub-partition is only applied to luma while chroma signals are coded without splitting. In the current ISP design, except the last ISP sub-partitions, the other sub-partitions only contain luma component.

The texts of a coding unit in the VVC draft are shown as below.

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	
chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0	
if(slice_type != I | | sps_ibc_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && ((!(cbWidth = = 4 && cbHeight = = 4) && modeType != MODE_TYPE_INTRA) | | (sps_ibc_enabled_flag && cbWidth <= 64 && cbHeight <= 64)))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] = = 0 && slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && modeType = = MODE_TYPE_ALL)	
pred_mode_flag	ae(v)

-continued

	Descriptor
if(((slice type = = I && cu_skip_flag[x0][y0] = =0) | | (slice type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA | | (((cbWidth = = 4 && cbHeight = = 4) | | modeType = = MODE_TYPE_INTRA) && cu_skip_flag[x0][y0] = = 0)))) && cbWidth <=64 && cbHeight <=64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[x0][y0] = = 0 && modeType!= MODE_TYPE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_act_enabled_flag && treeType = = SINGLE_TREE)	
cu_act_enabled_flag	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA | | CuPredMode[chType][x0][y0] = = MODE_PLT) {	
if(treeType = = SINGLE_TREE | | treeType = = DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
} else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
if(sps_mip_enabled_flag)	
intra_mip_flag[x0][y0]	ae(v)
if(intra_mip_flag[x0][y0]) {	
intra_mip_transposed[x0][y0]	ae(v)
intra_mip_mode[x0][y0]	ae(v)
} else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)
if(sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] = = 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && !cu_act_enabled_flag)	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] = = 1)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] = = 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] = = 0)	
intra_luma_not_planar_flag[x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
} else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
}	
}	
}	
if((treeType = = SINGLE_TREE | | treeType = = DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType = = DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, treeType)	
else {	
if(!cu_act_enabled_flag) {	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)

-continued

	Descriptor
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] = = 0)	
general_merge_flag[x0][y0]	ae(v)
if(general_merge_flag[x0][y0])	
merge_data(x0, y0, cbWidth, cbHeight, chType)	
else if(CuPredMode[chType][x0][y0] = = MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
if(MaxNumIbcMergeCand > 1)	
mvp_l0_flag[x0][y0]	ae(v)
if(sps_amvr_enabled_flag &&	
(MvdL0[x0][y0][0] != 0 | | MvdL0[x0][y0][1] != 0))	
amvr_precision_idx[x0][y0]	ae(v)
} else {	
if(slice_type = = B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16 {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !mvd_l1_zero_flag &&	
inter_pred_idc[x0][y0] = = PRED_BI &&	
!inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] >1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] = = PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	

-continued

	Descriptor
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
}	
mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if((sps_amvr_enabled_flag && inter_affine_flag[x0][y0] = = 0 &&	
(MvdL0[x0][y0][0] != 0 | | MvdL0[x0][y0][1] != 0 | |	
MvdL1[x0][y0][0] != 0 | | MvdL1[x0][y0][1] != 0)) | |	
(sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] = = 1 &&	
(MvdCpL0[x0][y0][0][0] != 0 | | MvdCpL0[x0][y0][0][1] != 0 | |	
MvdCpL1[x0][y0][0][0] != 0 | | MvdCpL1[x0][y0][0][1] != 0 | |	
MvdCpL0[x0][y0][1][0] != 0 | | MvdCpL0[x0][y0][1][1] != 0 | |	
MvdCpL1[x0][y0][1][0] != 0 | | MvdCpL1[x0][y0][1][1] != 0 | |	
MvdCpL0[x0][y0][2][0] != 0 | | MvdCpL0[x0][y0][2][1] != 0 | |	
MvdCpL1[x0][y0][2][0] !=0 | | MvdCpL1[x0][y0][2][1] != 0)) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_idx[x0][y0]	ae(v)
}	
if(sps_bcw_enabled_flag && inter_pred_idc[x0][y0] = = PRED_BI &&	
luma_weight_l0_flag[ref_idx_l0 [x0][y0]] = = 0 &&	
luma_weight_l1_flag[ref_idx_l1 [x0][y0]] = = 0 &&	
chroma_weight_l0_flag[ref_idx_l0 [x0][y0]] = = 0 &&	
chroma_weight_l1_flag[ref_idx_l1 [x0][y0]] = = 0 &&	
cbWidth * cbHeight >= 256)	
bcw_idx[x0][y0]	ae(v)
}	
}	
if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred mode_plt flag && general_merge_flag[x0][y0] = = 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[chType][x0][y0] = = MODE_INTER && sps_sbt_enabled_flag	
&& !ciip_flag[x0][y0] && !MergeTriangleFlag[x0][y0]	
&& cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH | | allowSbtHorH)	
cu_sbt_flag	ae(v)
if(cu_sbt_flag) {	
if((allowSbtVerH | | allowSbtHorH) && (allowSbtVerQ | | allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) | |	
(!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
if(sps_act_enabled_flag && CuPredMode[chType][x0][y0] != MODE_INTRA &&	
treeType = = SINGLE_TREE)	
cu_act_enabled_flag ae(v)	
LfnstDcOnly = 1	
LfnstZeroOutSigCoeffFlag = 1	
MtsZeroOutSigCoeffFlag = 1	
transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)	
lfnstWidth = (treeType = = DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType = = ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)	

-continued

```
                                                                    Descriptor
    lfnstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC
        : ( ( IntraSubPartitionsSplitType == ISP_HOR_SPLIT) ? cbHeight /
            NumIntraSubPartitions : cbHeight )
    if( Min( lfnstWidth, lfnstHeight ) >= 4 && sps_lfnst_enabled_flag == 1
&&
      CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&
      transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 &&
      ( treeType != DUAL_TREE_CHROMA | | !infra_mip_flag[ x0 ][ y0 ] | |
        Min( lfnstWidth, lfnstHeight ) >= 16 ) &&
      Max( cbWidth, cbHeight) <= MaxTbSizeY) {
      if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT | | LfnstDcOnly ==
0 ) &&
        LfnstZeroOutSigCoeffFlag == 1 )
        lfnst_idx                                                   ae(v)
    }
    if( treeType != DUAL_TREE_CHROMA && lfnst_idx == 0 &&
      transform skip flag[ x0 ][ y0 ][ 0 ] == 0 && Max( cbWidth, cbHeight )
<= 32 &&
      IntraSubPartitionsSplit[ x0 ][ y0 ] == ISP_NO_SPLIT && cu_sbt_flag
== 0 &&
      MtsZeroOutSigCoeffFlag == 1 && tu_cbf_luma[ x0 ][ y0 ] ) {
      if( ( ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTER &&
        sps_explicit_mts_inter_enabled_flag ) | |
        ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&
        sps_explicit_mts_intra_enabled_flag ) ) )
        mts_idx                                                     ae(v)
    }
  }
```

cu_act_enabled_flag equal to 1 specifies that the residuals of the current coding unit are coded in YC_gC_o color space. cu_act_enabled_flag equal to 0 specifies that the residuals of the current coding unit are coded in original color space. When cu_act_enabled_flag is not present, it is inferred to be equal to 0.

2.3. Transform Skip Mode in VVC

As in HEVC, the residual of a block can be coded with transform skip mode which completely skip the transform process for a block. In addition, for transform skip blocks, a minimum allowed Quantization Parameter (QP) signaled in SPS is used, which is set equal to 6*(internalBitDepth−inputBitDepth)+4 in VTM7.0.

2.4. Block-based Delta Pulse Code Modulation (BDPCM)

In JVET-M0413, a block-based Delta Pulse Code Modulation (BDPCM) is proposed to code screen contents efficiently and then adopted into VVC.

The prediction directions used in BDPCM can be vertical and horizontal prediction modes. The intra prediction is done on the entire block by sample copying in prediction direction (horizontal or vertical prediction) similar to intra prediction. The residual is quantized and the delta between the quantized residual and its predictor (horizontal or vertical) quantized value is coded. This can be described by the following: For a block of size M (rows)×N (cols), let $r_{i,j}$, 0≤i≤M−1, 0≤j≤N−1 be the prediction residual after performing intra prediction horizontally (copying left neighbor pixel value across the predicted block line by line) or vertically (copying top neighbor line to each line in the predicted block) using unfiltered samples from above or left block boundary samples. Let $Q(r_{i,j})$, 0≤i≤M−1, 0≤j≤N−1 denotes the quantized version of the residual $r_{i,j}$, where residual is difference between original block and the predicted block values. Then the block DPCM is applied to the quantized residual samples, resulting in modified M×N array $\tilde{R}$ with elements $\tilde{r}_{i,j}$. When vertical BDPCM is signaled:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \le j \le (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \le i \le (M-1), \quad 0 \le j \le (N-1) \end{cases}.$$

For horizontal prediction, similar rules apply, and the residual quantized samples are obtained by

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \le i \le (M-1), \quad j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \le i \le (M-1), \quad 1 \le j \le (N-1) \end{cases}.$$

The residual quantized samples $\tilde{r}_{i,j}$ are sent to the decoder.

On the decoder side, the above calculations are reversed to produce $Q(r_{i,j})$, 0≤i≤M−1, 0≤j≤N−1.

For vertical prediction case,

$$Q(r_{i,j}) = \Sigma_{k=0}^{i} \tilde{r}_{k,j}, \; 0 \le i \le (M-1), 0 \le j \le (N-1).$$

For horizontal case,

$$Q(r_{i,j}) = \Sigma_{k=0}^{j} \tilde{r}_{i,k}, 0 \le i \le (M-1), 0 \le j \le (N-1).$$

The inverse quantized residuals, $Q^{-1}(Q(r_{i,j}))$, are added to the intra block prediction values to produce the reconstructed sample values.

The main benefit of this scheme is that the inverse BDPCM can be done on the fly during coefficient parsing simply adding the predictor as the coefficients are parsed or it can be performed after parsing.

In VTM7.0, the BDPCM also can be applied on chroma blocks and the chroma BDPCM has a separate flag and BDPCM direction from the luma BDPCM mode.

2.5. Scaling Process for Transform Coefficients

The texts related to scaling process for transform coefficients in JVET-P2001-vE is given as follows.

Inputs to this process are:

- a luma location (xTbY, yTbY) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable nTbW specifying the transform block width,
- a variable nTbH specifying the transform block height,
- a variable predMode specifying the prediction mode of the coding unit,
- a variable cIdx specifying the color component of the current block.

Output of this process is the (nTbW)×(nTbH) array d of scaled transform coefficients with elements d[x][y].

The quantization parameter qP is derived as follows:

If cIdx is equal to 0, the following applies:

$$qP=Qp'_Y \tag{1129}$$

Otherwise, if TuCResMode[xTbY][yTbY] is equal to 2, the following applies:

$$qP=QP'_{CbCr} \tag{1130}$$

Otherwise, if cIdx is equal to 1, the following applies:

$$qP=Qp'_{Cb} \tag{1131}$$

Otherwise (cIdx is equal to 2), the following applies:

$$qP=Qp'_{Cr} \tag{1132}$$

The quantization parameter qP is modified and the variables rectNonTsFlag and bdShift are derived as follows:

If transform_skip_flag[xTbY][yTbY][cIdx] is equal to 0, the following applies:

$$qP=qP-(\text{cu_act_enabled_flag}[xTbY][yTbY]?5:0) \tag{1133}$$

$$\text{rectNonTsFlag}=(((\text{Log 2}(nTbW)+\text{Log 2}(nTbH))\ \&\ 1)==1)?1:0 \tag{1134}$$

$$\text{bdShift}=\text{BitDepth}+\text{rectNonTsFlag}+((\text{Log 2}(nTbW)+\text{Log 2}(nTbH))/2)-5+\text{pic_dep_quant_enabled_flag} \tag{1135}$$

Otherwise (transform_skip_flag[xTbY][yTbY][cIdx] is equal to 1), the following applies:

$$qP=\text{Max}(\text{QpPrimeTsMin},qP)-(\text{cu_act_enabled_flag}[xTbY][yTbY]?5:0) \tag{1136}$$

$$\text{rectNonTsFlag}=0 \tag{1137}$$

$$\text{bdShift}=10 \tag{1138}$$

The variable bdOffset is derived as follows:

$$\text{bdOffset}=(1<<\text{bdShift})>>1 \tag{1139}$$

The list levelScale[ ][ ] is specified as levelScale[j][k]={{40, 45, 51, 57, 64, 72}, {57, 64, 72, 80, 90, 102}} with j=0 . . . 1, k=0 . . . 5.

The (nTbW)×(nTbH) array dz is set equal to the (nTbW)×(nTbH) array TransCoeffLevel[xTbY][yTbY][cIdx].

For the derivation of the scaled transform coefficients d[x][y] with x=0 . . . nTbW−1, y=0 . . . nTbH−1, the following applies:

The intermediate scaling factor m[x][y] is derived as follows:

If one or more of the following conditions are true, m[x][y] is set equal to 16:

- sps_scaling_list_enabled_flag is equal to 0.
- pic_scaling_list_present_flag is equal to 0.
- transform_skip_flag[xTbY][yTbY][cIdx] is equal to 1.
- scaling_matrix_for_lfnst_disabled_flag is equal to 1 and lfnst_idx[xTbY][yTbY] is not equal to 0.

Otherwise, the following applies:

The variable id is derived based on predMode, cIdx, nTbW, and nTbH as specified in Table 36 and the variable log 2MatrixSize is derived as follows:

$$\text{log 2MatrixSize}=(id<2)?1:(id<8)?2:3 \tag{1140}$$

The scaling factor m[x][y] is derived as follows:

$$m[x][y]=\text{ScalingMatrixRec}[id][i][j]$$

$$\text{with } i=(x<<\text{log 2MatrixSize})>>\text{Log 2}(nTbW),$$

$$j=(y<<\text{log 2MatrixSize})>>\text{Log 2}(nTbH) \tag{1141}$$

If id is greater than 13 and both x and y are equal to 0, m[0][0] is further modified as follows:

$$m[0][0]=\text{ScalingMatrixDCRec}[id-14] \tag{1142}$$

NOTE—A quantization matrix element m[x][y] can be zeroed out when any of the following conditions is true:

- x is greater than 32,
- y is greater than 32,
- The decoded TU is not coded by default transform mode (i.e., transform type is not equal to 0) and x is greater than 16,
- The decoded TU is not coded by default transform mode (i.e., transform type is not equal to 0) and y is greater than 16.

The scaling factor ls[x][y] is derived as follows:

If pic_dep_quant_enabled_flag is equal to 1 and transform_skip_flag[xTbY][yTbY][cIdx] is equal to 0, the following applies:

$$ls[x][y]=(m[x][y]*\text{levelScale}[\text{rectNonTsFlag}][(qP+1)\ \%\ 6])<<((qP+1)/6) \tag{1143}$$

Otherwise (pic_dep_quant_enabled_flag is equal to 0 or transform_skip_flag[xTbY][yTbY][cIdx] is equal to 1), the following applies:

$$ls[x][y]=(m[x][y]*\text{levelScale}[\text{rectNonTsFlag}][qP\%6])<<(qP/6) \tag{1144}$$

When BdpcmFlag[xTbY][yTbY][cIdx] is equal to 1, dz[x][y] is modified as follows:

If BdpcmDir[xTbY][yTbY][cIdx] is equal to 0 and x is greater than 0, the following applies:

$$dz[x][y]=\text{Clip3}(\text{CoeffMin},\text{CoeffMax},dz[x-1][y]+dz[x][y]) \tag{1145}$$

Otherwise, if BdpcmDir[xTbY][yTbY][cIdx] is equal to 1 and y is greater than 0, the following applies:

$$dz[x][y]=\text{Clip3}(\text{CoeffMin},\text{CoeffMax},dz[x][y-1]+dz[x][y]) \tag{1146}$$

The value dnc[x][y] is derived as follows:

$$dnc[x][y]=(dz[x][y]*ls[x][y]+\text{bdOffset})>>\text{bdShift} \tag{1147}$$

The scaled transform coefficient d[x][y] is derived as follows:

$$d[x][y]=\text{Clip3}(\text{CoeffMin},\text{CoeffMax},dnc[x][y]) \tag{1148}$$

TABLE 36

Specification of the scaling matrix identifier variable id according to predMode, cIdx, nTbW, and nTbH

max(nTbW, nTbH)		2	4	8	16	32	64
predMode = MODE_INTRA	cIdx = 0 (Y)		2	8	14	20	26
	cIdx = 1 (Cb)		3	9	15	21	21
	cIdx = 2 (Cr)		4	10	16	22	22
predMode = MODE_INTER (INTER, IBC)	cIdx = 0 (Y)		5	11	17	23	27
	cIdx = 1 (Cb)	0	6	12	18	24	24
	cIdx = 2 (Cr)	1	7	13	19	25	25

2.6. Palette Mode

2.6.1. Concept of Palette Mode

The basic idea behind a palette mode is that the pixels in the CU are represented by a small set of representative color values. This set is referred to as the palette. And it is also possible to indicate a sample that is outside the palette by signalling an escape symbol followed by (possibly quantized) component values. This kind of pixel is called escape pixel. The palette mode is illustrated in FIG. **3**. As depicted in FIG. **3**, for each pixel with three color components (luma, and two chroma components), an index to the palette is founded, and the block could be reconstructed based on the founded values in the palette.

2.6.2. Coding of the Palette Entries

For coding of the palette entries, a palette predictor is maintained. The maximum size of the palette as well as the palette predictor is signaled in the SPS. In HEVC-SCC, a palette_predictor_initializer_present_flag is introduced in the PPS. When this flag is 1, entries for initializing the palette predictor are signaled in the bitstream. The palette predictor is initialized at the beginning of each CTU row, each slice and each tile. Depending on the value of the palette_predictor_initializer_present_flag, the palette predictor is reset to 0 or initialized using the palette predictor initializer entries signaled in the PPS. In HEVC-SCC, a palette predictor initializer of size 0 was enabled to allow explicit disabling of the palette predictor initialization at the PPS level.

For each entry in the palette predictor, a reuse flag is signaled to indicate whether it is part of the current palette. This is illustrated in FIG. **4**. The reuse flags are sent using run-length coding of zeros. After this, the number of new palette entries are signaled using Exponential Golomb (EG) code of order 0, i.e., EG-0. Finally, the component values for the new palette entries are signaled.

2.6.3. Coding of Palette Indices

The palette indices are coded using horizontal and vertical traverse scans as shown in FIG. **5**. The scan order is explicitly signaled in the bitstream using the palette_transpose_flag. For the rest of the subsection it is assumed that the scan is horizontal.

The palette indices are coded using two palette sample modes: 'COPY_LEFT' and 'COPY_ABOVE'. In the 'COPY_LEFT' mode, the palette index is assigned to a decoded index. In the 'COPY_ABOVE' mode, the palette index of the sample in the row above is copied. For both "COPY_LEFT" and 'COPY_ABOVE' modes, a run value is signaled which specifies the number of subsequent samples that are also coded using the same mode.

In the palette mode, the value of an index for the escape symbol is the number of palette entries. And, when escape symbol is part of the run in 'COPY_LEFT' or 'COPY_ABOVE' mode, the escape component values are signaled for each escape symbol. The coding of palette indices is illustrated in FIG. **6**.

This syntax order is accomplished as follows. First the number of index values for the CU is signaled. This is followed by signaling of the actual index values for the entire CU using truncated binary coding. Both the number of indices as well as the index values are coded in bypass mode. This groups the index-related bypass bins together. Then the palette sample mode (if necessary) and run are signaled in an interleaved manner. Finally, the component escape values corresponding to the escape symbols for the entire CU are grouped together and coded in bypass mode. The binarization of escape symbols is EG coding with 3rd order, i.e., EG-3.

An additional syntax element, last_run_type_flag, is signaled after signaling the index values. This syntax element, in conjunction with the number of indices, eliminates the need to signal the run value corresponding to the last run in the block.

In HEVC-SCC, the palette mode is also enabled for 4:2:2, 4:2:0, and monochrome chroma formats. The signaling of the palette entries and palette indices is almost identical for all the chroma formats. In case of non-monochrome formats, each palette entry consists of 3 components. For the monochrome format, each palette entry consists of a single component. For subsampled chroma directions, the chroma samples are associated with luma sample indices that are divisible by 2. After reconstructing the palette indices for the CU, if a sample has only a single component associated with it, only the first component of the palette entry is used. The only difference in signaling is for the escape component values. For each escape symbol, the number of escape component values signaled may be different depending on the number of components associated with that symbol.

2.6.4. Palette in Dual Tree

In VVC, the dual tree coding structure is used on coding the intra slices, so the luma component and two chroma components may have different palette and palette indices. In addition, the two chroma component shares same palette and palette indices.

2.6.5. Line Based Coefficient Group (CG) Palette Mode

Line based CG palette mode was adopted to VVC. In this method, each CU of palette mode is divided into multiple segments of m samples (m=16 in this test) based on the traverse scan mode. The encoding order for palette run coding in each segment is as follows: For each pixel, 1 context coded bin ***run_copy_flag*** =0 is signaled indicating if the pixel is of the same mode as the previous pixel, i.e., if the previous scanned pixel and the current pixel are both of run type COPY_ABOVE or if the previous scanned pixel and the current pixel are both of run type INDEX and the same index value. Otherwise, ***run_copy_flag*** =1 is signaled. If the pixel and the previous pixel are of different mode, one context coded bin ***copy_above_palette_indices_flag*** is signaled indicating the run type, i.e., INDEX or COPY_ABOVE, of the pixel. Same as the palette mode in VTM6.0,

decoder doesn't have to parse run type if the sample is in the first row (horizontal traverse scan) or in the first column (vertical traverse scan) since the INDEX mode is used by default. Also, decoder doesn't have to parse run type if the previously parsed run type is COPY_ABOVE. After palette run coding of pixels in one segment, the index values (for INDEX mode) and quantized escape colors are bypass coded and grouped apart from encoding/parsing of context coded bins to improve throughput within each line CG. Since the index value is now coded/parsed after run coding, instead of processed before palette run coding as in VTM, encoder doesn't have to signal the number of index values ***num_palette_indices_minus1*** and the last run type ***copy_above_indices_for_final_run_flag***. .

3. Technical Problems Solved by Embodiments and Solutions Described Herein

In the current design, the ACT and luma BDPCM modes can be enabled for one block. However, the chroma BDPCM mode may always be disabled on blocks coded with ACT mode. Therefore, the prediction signal may be derived differently for luma and chroma blocks in the same coding unit which may be less efficient.

The quantization parameter (QP) of a block may become a minus number when ACT is enabled.

The current design of ACT does not support lossless coding.

The signalling of usage of ACT is not block size dependent.

The maximal palette size and the maximal predictor size are fixed numbers, which may limit the flexibility of palette mode.

Escape samples employs Exponential-Golomb (EG) with 3th order as the binarization method, but the binarization for escape samples is not dependent on Quantization Parameter (QP).

4. Technical Solutions

The technical solutions described below should be considered as examples to explain general concepts. These technical solutions should not be interpreted in a narrow way. Furthermore, these technical solutions can be combined in any manner.

In the following description, the term 'block' may represent a video region, such as a coding unit (CU), a prediction unit (PU), a transform unit (TU), which may contain samples in three color components. The term 'BDPCM' is not limited to the design in VVC, but it may present the technologies that coding residuals using different prediction signal generation methods.

Interaction Between ACT and BDPCM (Items 1-4)

1. Whether to enable chroma BDPCM mode may depend on the usage of ACT and/or luma BDPCM mode.
   a. In one example, when ACT is enabled on a block, the indication of the usage of chroma BDPCM mode (e.g., intra_bdpcm_chroma_flag) may be inferred to be the indication of the usage of the luma BDPCM mode (e.g., intra_bdpcm_luma_flag).
      i. In one example, the inferred value of chroma BDPCM mode is defined as (ACT and luma BDPCM modes are enabled? true: false).
         1. In one example, the intra_bdpcm_chroma_flag may be set equal to false when intra_bdpcm_luma_flag is false.
            a. Alternatively, the intra_bdpcm_chroma_flag may be set equal to true when intra_bdpcm_luma_flag is true.
      ii. Alternatively, in one example, the indication of the usage of chroma BDPCM mode may be inferred to be true if the indication of the usage of the luma BDPCM mode and ACT for the block are true.
   b. Alternatively, whether to signal the usage of ACT for a block may be conditionally checked, such as the same BDPCM prediction direction is used for both luma and chroma samples in the block.
      i. Alternatively, furthermore, indication of usage of ACT is signaled after usage of BDPCM modes.

2. When ACT is enabled on a block, the indication of the prediction direction of chroma BDPCM mode (e.g., intra_bdpcm_chroma_dir_flag) may be inferred to be the indication of the prediction direction of the usage of the luma BDPCM mode (e.g., intra_bdpcm_luma_dir_flag).
   a. In one example, the inferred value of intra_bdpcm_chroma_dir_flag is defined as (ACT is enabled? intra_bdpcm_luma_dir_flag: 0).
      i. In one example, the indication of the prediction direction of chroma BDPCM mode may be inferred to horizontal if the indication of the prediction direction of the luma BDPCM mode is horizontal.
      ii. Alternatively, in one example, the indication of the prediction direction of chroma BDPCM mode may be inferred to vertical if the indication of the prediction direction of the luma BDPCM mode is vertical.

3. The ACT and BDPCM modes may be exclusively applied.
   a. In one example, when ACT mode is enabled on a block, the BDPCM mode may be disabled on the block.
      i. Alternatively, furthermore, the indication of usage of BDPCM mode may be signaled after the signalling of the indication of usage of ACT mode.
      ii. Alternatively, furthermore, the indication of usage of BDPCM mode may be not signaled and inferred to false (0).
   b. In one example, when BDPCM mode is enabled on a block, the ACT mode may be disabled on the block.
      i. Alternatively, furthermore, the indication of usage of ACT mode may be signaled after the signalling of the indication of usage of BDPCM mode.
      ii. Alternatively, furthermore, the indication of usage of ACT mode may be not signaled and inferred to false (0).
   c. In one example, the BDPCM mode in the above examples may denote luma BDPCM mode and/or chroma BDPCM mode.

4. Inverse ACT may be applied before reverse BDPCM at the decoder.
   a. In one example, ACT may be applied even when luma and chroma BDPCM have a different prediction mode.
   b. Alternatively, at the encoder, forward ACT may be applied after BDPCM.

QP Setting When ACT is Enabled (Item 5)

5. It is proposed to clip the QP when ACT is enabled.
   a. In one example, the clipping function may be defined as (l, h, x), where l is the lowest possible value of the input x and h is the highest possible value of the input x.
      i. In one example, l may be set equal to 0.
      ii. In one example, h may be set equal to 63.
   b. In one example, the QP may be the qP given in Section 2.5.
   c. In one example, the clipping may be performed after the QP adjustment for ACT mode.

d. In one example, when transform skip is applied, l may be set equal to the minimal allowed QP for transform skip mode.

Palette Mode Related (Item 6-7)

6. The values of maximum allowed palette size and/or maximum allowed predictor size may depend on a coding characteristic. Assume S_1 is the maximal palette size (or palette predictor size) associated with a first coding characteristic; and S_2 is the maximal palette size (or palette predictor size) associated with a second coding characteristic.

a. In one example, the coding characteristic may be color component.
   i. In one example, the maximum allowed palette size and/or maximum allowed predictor size for different color components may have different values.
   ii. In one example, the values of maximum allowed palette size and/or maximum allowed predictor size for a first color components (e.g., Y in YCbCr, G in RGB) may be different from those for the other two-color components (e.g., Cb and Cr in YCbCr, B and R in RGB) excluding the first color component.
b. In one example, the coding characteristics may be the quantization parameters (QPs).
   i. In one example, S_1 and/or S_2 for QP_1 may be smaller than the S_1 and/or S_2 for QP_2 if QP_1 is greater than QP_2
   ii. In one example, the QP may be the slice level QP or block level QP.
c. In one example, S_2 may be greater than or equal to S_1.
d. For a first coding characteristic and a second coding characteristic, indications of maximum palette sizes/palette predictor sizes may be signaled separately or inferred from one to another.
   i. In one example, S_1 may be signaled and S_2 may be derived based on S_1.
      1. In one example, S_2 may be inferred to S_1-n.
      2. In one example, S_2 may be inferred to $S_1>>n$.
      3. In one example, S_2 may be inferred to floor (S_1/n), where floor(x) denotes the maximum integer no larger than x.
e. In one example, S_1 and/or S_2 may be signaled at a high level (e.g., SPS/PPS/PH/slice header) and adjusted at a lower level (e.g., CU/block).
   i. How to adjust S_1 and/or S_2 may depend on the coded information.
      1. How to adjust S_1 and/or S_2 may depend on the current QP.
         a. In one example, the S_1 and/or S_2 may be reduced if the current QP is increased.
      2. How to adjust S_1 and/or S_2 may depend on block dimension.
         a. In one example, the S_1 and/or S_2 may be increased if the current blocks size is increased.
f. S_1 and/or S_2 may be dependent on whether LMCS is used.

7. The parameters associated with a binarization method for escape samples/pixels may depend on coded information, e.g., the quantization parameters (QPs).

a. In one example, EG binarization method may be utilized, and the order of the EG binarization, denoted by k, may be dependent on the coded information.
   i. In one example, k may be reduced when the current QP is increased.

Signaling of ACT Mode (Item 8-10)

8. Indications of the allowed maximum and/or minimum ACT size may be signaled in sequence/video/slice/tile/subpicture/brick/other video processing unit level or derived based on coded information.

a. In one example, they may be signaled in SPS/PPS/Picture header/slice header.
b. In one example, they may be conditionally signaled, such as according to the ACT is enabled.
c. In one example, N-level of allowed maximum and/or minimum ACT size may be signaled/defined, e.g., N=2.
   i. In one example, the allowed maximum and/or minimum ACT size may be set to either K0 or K1 (e.g., K0=64, K1=32).
   ii. Alternatively, furthermore, indication of the level may be signaled, e.g., when N=2, a flag may be signaled.
d. In one example, indications of the differences between the allowed maximum and/or minimum ACT size and the allowed maximum and/or minimum transform (or transform skip) sizes (e.g., for the luma component) may be signaled.
e. In one example, the allowed maximum and/or minimum ACT size may be derived from the allowed maximum and/or minimum (or transform skip) sizes (e.g., for the luma component).
f. Alternatively, furthermore, whether to and/or how to signal indications of ACT usage and other side information related to ACT may be dependent on the allowed maximum and/or minimum.

9. When a block is greater than the allowed maximum ACT size (or allowed maximum transform size), the block may be automatically split to multiple sub-blocks wherein all sub-blocks share the same prediction mode (e.g., all of them are intra coded), and the ACT may be enabled in the sub-block level, instead of the block level.

10. The indication of the usage of ACT mode may be conditionally signaled based on the block dimensions (e.g., block width and/or height, block width times height, ratios between block width and height, maximum/minimum values of block width and height) and/or maximumly allowed ACT sizes.

a. In one example, the indication of the usage of ACT mode may be signaled when certain conditions, e.g., according to block dimension, are satisfied.
   i. In one example, the conditions are if the current block width is smaller than or equal to m and/or the current block height is smaller than or equal to n.
   ii. In one example, the conditions are if the current block width times height is smaller than or no larger than m.
   iii. In one example, the conditions are if the current block width times height is larger than or no smaller than m.
b. Alternatively, in one example, the indication of the usage of ACT mode may be not signaled when certain conditions, e.g., according to block dimension, are NOT satisfied.
   i. In one example, the conditions are if the current block width is larger than m and/or the current block height is larger than n.

ii. In one example, the conditions are if the current block width times height is smaller than or no larger than m.

iii. In one example, the conditions are if the current block width times height is larger than or no smaller than m.

iv. Alternatively, furthermore, the indication of the usage of ACT mode may be inferred to 0.

c. In the above examples, the variables m, n may be pre-defined (e.g., 4, 64, 128), or signaled, or derived on-the-fly.

i. In one example, m and/or n may be derived based on a decoded message in the SPS/PPS/adaptation parameter set (APS)/CTU row/group of CTUs/CU/block.

1. In one example, m and/or n may be set equal to the maximally allowed transform size (e.g., MaxTbSizeY).

Signaling of Constraint Flags in General Constraint Information Syntax (Item 11-16)

The constraint flags below may be signaled in a video unit other than SPS. For example, they may be signaled in the general constraint information syntax specified in JVET-P2001-vE.

11. It is proposed to have a constraint flag to specify whether the SPS ACT enabling flag (e.g., sps_act_enabled_flag) may be equal to 0.

a. In one example, the flag may be denoted as no_act_constraint_flag

i. When this flag is equal to 1, the SPS ACT enabling flag (e.g., sps_act_enabled_flag) may be equal to 0.

ii. When this flag is equal to 0, it does not impose such a constraint.

12. It is proposed to have a constraint flag to specify whether the SPS BDPCM enabling flag (e.g., sps_bdpcm_enabled_flag) may be equal to 0.

a. In one example, the flag may be denoted as no_bdpcm_constraint_flag.

i. When this flag is equal to 1, the SPS BDPCM enabling flag (e.g., sps_bdpcm_enabled_flag) may be equal to 0.

ii. When this flag is equal to 0, it does not impose such a constraint.

13. It is proposed to have a constraint flag to specify whether the SPS chroma BDPCM enabling flag (e.g., sps_bdpcm_chroma_enabled_flag) may be equal to 0.

a. In one example, the flag may be denoted as no_bdpcm_chroma_constraint_flag.

i. When this flag is equal to 1, the SPS chroma BDPCM enabling flag (e.g., sps_bdpcm_chroma_enabled_flag) may be equal to 0.

ii. When this flag is equal to 0, it does not impose such a constraint.

14. It is proposed to have a constraint flag to specify whether the SPS palette enabling flag (e.g., sps_palette_enabled_flag) may be equal to 0.

a. In one example, the flag may be denoted as no_palette_constraint_flag.

i. When this flag is equal to 1, the SPS palette enabling flag (e.g., sps_palette_enabled_flag) may be equal to 0.

ii. When this flag is equal to 0, it does not impose such a constraint.

15. It is proposed to have a constraint flag to specify whether the SPS RPR enabling flag (e.g., ref_pic_resampling_enabled_flag) may be equal to 0.

a. In one example, the flag may be denoted as no_ref_pic_resampling_constraint_flag.

i. When this flag is equal to 1, the SPS RPR enabling flag (e.g., ref_pic_resampling_enabled_flag) may be equal to 0.

ii. When this flag is equal to 0, it does not impose such a constraint.

16. In the above examples (bullet 11-15), such constraint flags may be conditionally signaled, e.g., according to chroma format (e.g., chroma_format_idc) and/or separate plane coding or ChromaArrayType.

ACT QP Offset (Item 17-19)

17. It is proposed that ACT offsets may be applied after applying other chroma offsets (e.g., those in PPS and/or picture header (PH) and/or slice header (SH)) when ACT is applied on a block.

18. It is proposed to have PPS and/or PH offsets other than −5 for JCbCr mode 2 when YCgCo color transform is applied on a block.

a. In one example, the offset may be other than −5.

b. In one example, the offset may be indicated in PPS (e.g., as pps_act_cbcr_qp_offset_plus6) and the offset may be set as pps_act_cbcr_qp_offset_plus6-6.

c. In one example, the offset may be indicated in PPS (e.g., as pps_act_cbcr_qp_offset_plus7) and the offset may be set as pps_act_cbcr_qp_offset_plus7-7.

19. It is proposed to have PPS and/or PH offsets other than 1 for JCbCr mode 2 when YCgCo-R is applied on a block.

a. In one example, the offset may be other than −1.

b. In one example, the offset may be indicated in PPS (e.g., as pps_act_cbcr_qp_offset) and the offset may be set as pps_act_cbcr_qp_offset.

c. In one example, the offset may be indicated in PPS (e.g., as pps_act_cbcr_qp_offset_plus1) and the offset may be set as pps_act_cbcr_qp_offset_plus1-1.

General Techniques ((Items 20-21)

20. In the above examples, S_1, S_2, l, h, m, n and/or k are integer numbers and may depend on

a. A message signaled in the decoding parameter set (DPS)/SPS/video parameter set (VPS)/PPS/APS/picture header/slice header/tile group header/Largest coding unit (LCU)/Coding unit (CU)/LCU row/group of LCUs/TU/PU block/Video coding unit

b. Position of CU/PU/TU/block/Video coding unit

c. Coded modes of blocks containing the samples along the edges

d. Transform matrices applied to the blocks containing the samples along the edges

e. Block dimension/Block shape of current block and/or its neighboring blocks

f. Indication of the color format (such as 4:2:0, 4:4:4, RGB or YUV)

g. Coding tree structure (such as dual tree or single tree)

h. Slice/tile group type and/or picture type

i. Colour component (e.g., may be only applied on Cb or Cr)

j. Temporal layer ID

k. Profiles/Levels/Tiers of a standard

l. Alternatively, S_1, S_2, l, h, m, n and/or k may be signaled to the decoder.

21. The above proposed methods may be applied under certain conditions.

a. In one example, the condition is the color format is 4:2:0 and/or 4:2:2.

b. In one example, indication of usage of the above methods may be signaled in sequence/picture/slice/tile/brick/a video region-level, such as SPS/PPS/picture header/slice header.

c. In one example, the usage of the above methods may depend on

i. Video contents (e.g., screen contents or natural contents)

ii. A message signaled in the DPS/SPS/VPS/PPS/APS/picture header/slice header/tile group header/Largest coding unit (LCU)/Coding unit (CU)/LCU row/group of LCUs/TU/PU block/Video coding unit

iii. Position of CU/PU/TU/block/Video coding unit

iv. Coded modes of blocks containing the samples along the edges

v. Transform matrices applied to the blocks containing the samples along the edges

vi. Block dimension of current block and/or its neighboring blocks

vii. Block shape of current block and/or its neighboring blocks

viii. Indication of the color format (such as 4:2:0, 4:4:4, RGB or YUV)

ix. Coding tree structure (such as dual tree or single tree)

x. Slice/tile group type and/or picture type

xi. Colour component (e.g., may be only applied on Cb or Cr)

xii. Temporal layer ID

xiii. Profiles/Levels/Tiers of a standard

xiv. Alternatively, m and/or n may be signaled to the decoder.

5. Embodiments

The embodiments are based on JVET-P2001-vE. The newly added texts are highlight by ***bold italicized underlined text*** . The deleted texts are marked by italicized text.

5.1. Embodiment #1

This embodiment is related to interaction between ACT and BDPCM modes.

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	
chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0	
...	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_palette_enabled_flag &&	
cbWidth <=64 && cbHeight <=64 && cu_skip_flag[x0][y0] = = 0 &&	
modeType != MODE_TYPE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_act_enabled_flag &&	
treeType = = SINGLE_TREE)	
cu_act_enabled_flag	ae(v)
if(CuPredMode[chType][x0][y0] = = MODE_INTRA ||	
CuPredMode[chType][x0][y0] = = MODE_PLT) {	
if(treeType = = SINGLE_TREE || treeType = = DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
} else {	
if(sps_bdpcm_enabled_flag &&	
cbWidth <= MaxTsSize && cbHeight <= MaxTsSize &&	
!cu act enabled flag)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
...	
}	
}	
}	
...	
}	

5.2. Embodiment #2

This embodiment is related to interaction between ACT and BDPCM modes.

intra_bdpcm_chroma_flag equal to 1 specifies that BDPCM is applied to the current chroma coding blocks at the location (x0, y0), i.e., the transform is skipped, the intra chroma prediction mode is specified by intra_bdpcm_chroma_dir_flag. intra_bdpcm_chroma_flag equal to 0 specifies that BDPCM is not applied to the current chroma coding blocks at the location (x0, y0).

When intra_bdpcm_chroma_flag is not present and ***and cu act enabled flag is false***, it is inferred to be equal to 0.

When intra_bdpcm_chroma_flag is not present and cu_act_enabled_flag is true, it is inferred to be equal to intra_bdpcm luma flag.

The variable BdpcmFlag[x][y][cIdx] is set equal to intra_bdpcm_chroma_flag for x=x0 . . . x0+cbWidth−1, y=y0 . . . y0+cbHeight−1 and cIdx=1 . . . 2.

intra_bdpcm_chroma_dir_flag equal to 0 specifies that the BDPCM prediction direction is horizontal. intra_bdpcm_chroma_dir_flag equal to 1 specifies that the BDPCM prediction direction is vertical.

When cu_act_enabled_flag is true, intra_bdpcm_chroma_dir_flag is set equal to intra_bdpcm_luma_dir_flag.

The variable BdpcmDir[x][y][cIdx] is set equal to intra_bdpcm_chroma_dir_flag for x=x0 . . . x0+cbWidth−1, y=y0 . . . y0+cbHeight−1 and cIdx=1 . . . 2.

5.3. Embodiment #3

This embodiment is related to QP setting.

8.7.3 Scaling process for Transform Coefficients

Inputs to this process are:

- a luma location (xTbY, yTbY) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable nTbW specifying the transform block width,
- a variable nTbH specifying the transform block height,
- a variable predMode specifying the prediction mode of the coding unit,
- a variable cIdx specifying the color component of the current block.

Output of this process is the (nTbW)×(nTbH) array d of scaled transform coefficients with elements d[x][y].

The quantization parameter qP is modified and the variables rectNonTsFlag and bdShift are derived as follows:

If transform_skip_flag[xTbY][yTbY][cIdx] is equal to 0, the following applies:

$qP=qP-(\text{cu_act_enabled_flag}[xTbY][yTbY]?5:0)$ (1133)

qP = Clip3(0, 63, qP)

$\text{rectNonTsFlag}=(((\text{Log}2(nTbW)+\text{Log}2(nTbH))\ \&\ 1)==1)?1:0$ (1134)

$\text{bdShift}=\text{BitDepth}+\text{rectNonTsFlag}+((\text{Log}2(nTbW)+\text{Log}2(nTbH))/2)-5+\text{pic_dep_quant_enabled_flag}$ (1135)

Otherwise (transform_skip_flag[xTbY][yTbY][cIdx] is equal to 1), the following applies:

$qP=\text{Max}(\text{QpPrimeTsMin},qP)-(\text{cu_act_enabled_flag}[xTbY][yTbY]?5:0)$ (1136)

qP = Max(QpPrimeTsMin, qP - (cu_act_enabled_flag[xTbY][yTbY] ? 5 : 0))

$\text{rectNonTsFlag}=0$ (1137)

$\text{bdShift}=10$ (1138)

5.4. Embodiment #4

8.7.1 Derivation Process for Quantization Parameters

The chroma quantization parameters for the Cb and Cr components, Qp'Cb and Qp'Cr, and joint Cb-Cr coding Qp'CbCr are derived as follows:

```
Qp'Cb = Clip3( -QpBdOffset, 63, qPCb + pps_cb_qp_offset +
slice_cb_qp_offset + CuQpOffsetCb )
        + QpBdOffset                                        (1122)
Qp'Cr = Clip3( -QpBdOffset, 63, qPCr + pps_cr_qp_offset +
slice_cr_qp_offset + CuQpOffsetCr )
        + QpBdOffset                                        (1123)
Qp'CbCr = Clip3( -QpBdOffset, 63, qPCbCr + pps_joint_cbcr_qp_offset +
          slice_joint_cbcr_qp_offset
+CuQpOffsetCbCr ) + QpBdOffset                              (1124)
```

5.5. Embodiment #5

7.3.9.5 Coding Unit Syntax

```
                                                                    Descriptor
coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {
    chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0
    ...
    if( CuPredMode[ chType ][ x0 ][ y0 ] = = MODE_INTRA &&
sps_act_enabled_flag                                        &&
      treeType = = SINGLE_TREE && cbWidth <= K0 && cbHeight <= K1)
      cu_act_enabled_flag                                   ae(v)
    ...
```

-continued

	Descriptor
if(cu_cbf) {	
if(sps_act_enabled_flag && CuPredMode[chType][x0][y0] != MODE_INTRA && treeType = = SINGLE_TREE ***&& cbWidth <= K0 && cbHeight <= K1***)	
cu_act_enabled_flag	ae(v)
...	
}	
}	

K0 and K1 are set equal to 32.

5.6. Embodiment #6

7.3.9.5 Coding Unit Syntax

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	
chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0	
...	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_act_enabled_flag && treeType = = SINGLE_TREE ***&& cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY***)	
cu_act_enabled_flag	ae(v)
...	
if(cu_cbf) {	
if(sps_act_enabled_flag && CuPredMode[chType][x0][y0] != MODE_INTRA && treeType = = SINGLE_TREE ***&& cbWidth <=MaxTbSizeY && cbHeight <= MaxTbSizeY***)	
cu_act_enabled_flag	ae(v)
...	
}	
}	

FIG. **7** is a block diagram showing an example video processing system **700** in which various embodiments disclosed herein may be implemented. Various implementations may include some or all of the components of the system **700**. The system **700** may include input **702** for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8- or 10-bit multi-component pixel values, or may be in a compressed or encoded format. The input **702** may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

The system **700** may include a coding component **704** that may implement the various coding or encoding methods described in the present disclosure. The coding component **704** may reduce the average bitrate of video from the input **702** to the output of the coding component **704** to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component **704** may be either stored, or transmitted via a communication connected, as represented by the component **706**. The stored or communicated bitstream (or coded) representation of the video received at the input **702** may be used by the component **708** for generating pixel values or displayable video that is sent to a display interface **710**. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as "coding" operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or DisplayPort, and so on. Examples of storage interfaces include serial advanced technology attachment (SATA), peripheral component interconnect (PCI), integrated drive electronics (IDE) interface, and the like. The embodiments described in the present disclosure may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

FIG. **8** is a block diagram of a video processing apparatus **800**. The apparatus **800** may be used to implement one or more of the methods described herein. The apparatus **800** may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus **800** may include one or more processors **802**, one or more memories **804** and video processing hardware **806**. The memory (memories) **804** may be used for storing data and code used for implementing the methods and embodiments described herein. The video processing hardware **806** may be used to implement, in hardware circuitry, some embodiments described in the present disclosure. In some embodiments, the hardware **806** may be partly or entirely in the processors **802**, e.g., a graphics processor.

FIG. **9** is a block diagram that illustrates an example video coding system **100** that may utilize the embodiments of this disclosure. As shown in FIG. **9**, video coding system **100** may include a source device **110** and a destination device

120. Source device 110 generates encoded video data which may be referred to as a video encoding device. Destination device 120 may decode the encoded video data generated by source device 110 which may be referred to as a video decoding device. Source device 110 may include a video source 112, a video encoder 114, and an input/output (I/O) interface 116.

Video source 112 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 114 encodes the video data from video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 116 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination device 120 via I/O interface 116 through network 130*a*. The encoded video data may also be stored onto a storage medium/server 130*b* for access by destination device 120.

Destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122.

I/O interface 126 may include a receiver and/or a modem. I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/server 130*b*. Video decoder 124 may decode the encoded video data. Display device 122 may display the decoded video data to a user. Display device 122 may be integrated with the destination device 120, or may be external to destination device 120, which may be configured to interface with an external display device.

Video encoder 114 and video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or further standards.

FIG. 10 is a block diagram illustrating an example of video encoder 200, which may be video encoder 114 in the system 100 illustrated in FIG. 9.

Video encoder 200 may be configured to perform any or all of the embodiments of this disclosure. In the example of FIG. 10, video encoder 200 includes a plurality of functional components. The embodiments described in this disclosure may be shared among the various components of video encoder 200. In some examples, a processor may be configured to perform any or all of the embodiments described in this disclosure.

The functional components of video encoder 200 may include a partition unit 201; a prediction unit 202, which may include a mode select unit 203, a motion estimation unit 204, a motion compensation unit 205, and an intra prediction unit 206; a residual generation unit 207; a transform unit 208; a quantization unit 209; an inverse quantization unit 210; an inverse transform unit 211; a reconstruction unit 212; a buffer 213; and an entropy encoding unit 214.

In other examples, video encoder 200 may include more, fewer, or different functional components. In an example, prediction unit 202 may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

Furthermore, some components, such as motion estimation unit 204 and motion compensation unit 205 may be highly integrated, but are represented in the example of FIG. 10 separately for purposes of explanation.

Partition unit 201 may partition a picture into one or more video blocks. Video encoder 200 and video decoder 300 may support various video block sizes.

Mode select unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra- or inter-coded block to a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some examples, mode select unit 203 may select a combination of intra and inter prediction (CIIP) mode in which the prediction is based on an inter prediction signal and an intra prediction signal. Mode select unit 203 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-prediction.

To perform inter prediction on a current video block, motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. Motion compensation unit 205 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 213 other than the picture associated with the current video block.

Motion estimation unit 204 and motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

In some examples, motion estimation unit 204 may perform uni-directional prediction for the current video block, and motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

In other examples, motion estimation unit 204 may perform bi-directional prediction for the current video block, motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

In some examples, motion estimation unit **204** may output a full set of motion information for decoding processing of a decoder.

In some examples, motion estimation unit **204** may not output a full set of motion information for the current video. Rather, motion estimation unit **204** may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit **204** may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

In one example, motion estimation unit **204** may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder **300** that the current video block has the same motion information as the other video block.

In another example, motion estimation unit **204** may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder **300** may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

As discussed above, video encoder **200** may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder **200** include advanced motion vector prediction (AMVP) and merge mode signaling.

Intra prediction unit **206** may perform intra prediction on the current video block. When intra prediction unit **206** performs intra prediction on the current video block, intra prediction unit **206** may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

Residual generation unit **207** may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

In other examples, there may be no residual data for the current video block, for example in a skip mode, and residual generation unit **207** may not perform the subtracting operation.

Transform processing unit **208** may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

After transform processing unit **208** generates a transform coefficient video block associated with the current video block, quantization unit **209** may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

Inverse quantization unit **210** and inverse transform unit **211** may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit **212** may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the prediction unit **202** to produce a reconstructed video block associated with the current block for storage in the buffer **213**.

After reconstruction unit **212** reconstructs the video block, loop filtering operation may be performed reduce video blocking artifacts in the video block.

Entropy encoding unit **214** may receive data from other functional components of the video encoder **200**. When entropy encoding unit **214** receives the data, entropy encoding unit **214** may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

FIG. **11** is a block diagram illustrating an example of video decoder **300** which may be video decoder **124** in the system **100** illustrated in FIG. **9**.

The video decoder **300** may be configured to perform any or all of the embodiments of this disclosure. In the example of FIG. **11**, the video decoder **300** includes a plurality of functional components. The embodiments described in this disclosure may be shared among the various components of the video decoder **300**. In some examples, a processor may be configured to perform any or all of the embodiments described in this disclosure.

In the example of FIG. **11**, video decoder **300** includes an entropy decoding unit **301**, a motion compensation unit **302**, an intra prediction unit **303**, an inverse quantization unit **304**, an inverse transformation unit **305**, a reconstruction unit **306**, and a buffer **307**. Video decoder **300** may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder **200** (FIG. **10**).

Entropy decoding unit **301** may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit **301** may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit **302** may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit **302** may, for example, determine such information by performing the AMVP and merge mode.

Motion compensation unit **302** may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

Motion compensation unit **302** may use interpolation filters as used by video encoder **200** during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit **302** may determine the interpolation filters used by video encoder **200** according to received syntax information and use the interpolation filters to produce predictive blocks.

Motion compensation unit **302** may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence.

Intra prediction unit **303** may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit **304** inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and

decoded by entropy decoding unit **301**. Inverse transform unit **305** applies an inverse transform.

Reconstruction unit **306** may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit **305** or intra prediction unit **303** to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer **307**, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

FIGS. **12-24** show example methods that can implement the technical solution described above in, for example, the embodiments shown in FIGS. **7-11**.

FIG. **12** shows a flowchart for an example method **1200** of video processing. The method **1200** includes, at operation **1210**, determining, for a conversion between a current video block of a video and a bitstream of the video, an allowed maximum size and/or an allowed minimum size for an adaptive color transform (ACT) mode used for coding the current video block.

The method **1200** includes, at operation **1220**, performing, based on the determining, the conversion.

FIG. **13** shows a flowchart for an example method **1300** of video processing. The method **1300** includes, at operation **1310**, determining, for a conversion between a current video block of a video and a bitstream of the video, a maximum allowed palette size and/or a minimum allowed predictor size for a palette mode used for coding the current video block, the maximum allowed palette size and/or the minimum allowed predictor size being based on a coding characteristic of the current video block.

The method **1300** includes, at operation **1320**, performing, based on the determining, the conversion.

FIG. **14** shows a flowchart for an example method **1400** of video processing. The method **1400** includes, at operation **1410**, performing a conversion between a current video block of a video and a bitstream of the video, the current video block being coded using a palette mode coding tool, and the bitstream conforming to a format rule that specifies that parameters associated with a binarization of an escape symbol for the current video block in the bitstream is based on coded information of the current video block.

FIG. **15** shows a flowchart for an example method **1500** of video processing. The method **1500** includes, at operation **1510**, determining, for a conversion between a video comprising a block and a bitstream of the video, that a size of the block is greater than a maximum allowed size for an adaptive color transform (ACT) mode.

The method **1500** includes, at operation **1520**, performing, based on the determining, the conversion, and in response to the size of the block being greater than the maximum allowed size for the ACT mode, the block being partitioned into multiple sub-blocks, each of the multiple sub-blocks sharing a same prediction mode, and the ACT mode being enabled at a sub-block level.

FIG. **16** shows a flowchart for an example method **1600** of video processing. The method **1600** includes, at operation **1610**, performing a conversion between a current video block of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies that whether an indication of a usage of an adaptive color transform (ACT) mode on the current video block is signaled in the bitstream is based on at least one of a dimension of the current video block or a maximum allowed size for the ACT mode.

FIG. **17** shows a flowchart for an example method **1700** of video processing. The method **1700** includes, at operation **1710**, performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that an adaptive color transform (ACT) mode for the current video unit is disabled.

FIG. **18** shows a flowchart for an example method **1800** of video processing. The method **1800** includes, at operation **1810**, performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for the current video unit is disabled.

FIG. **19** shows a flowchart for an example method **1900** of video processing. The method **1900** includes, at operation **1910**, performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for a chroma component of the current video unit is disabled.

FIG. **20** shows a flowchart for an example method **2000** of video processing. The method **2000** includes, at operation **2010**, performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a palette for the current video unit is disabled.

FIG. **21** shows a flowchart for an example method **2100** of video processing. The method **2100** includes, at operation **2110**, performing a conversion between a current video unit of a video and a bitstream of the video, the bitstream conforming to a format rule that specifies whether a first flag is included in the bitstream, and the first flag indicating whether a second flag in a sequence parameter set (SPS) specifies that a reference picture resampling (RPR) mode for the current video unit is disabled.

FIG. **22** shows a flowchart for an example method **2200** of video processing. The method **2200** includes, at operation **2210**, performing a conversion between a current video block of a video and a bitstream of the video according to a rule that specifies additional quantization parameter offsets are applied when an adaptive color transform (ACT) mode is enabled for the current video block.

FIG. **23** shows a flowchart for an example method **2300** of video processing. The method **2300** includes, at operation **2310**, performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, the current video block being coded using a joint CbCr coding mode in which a YCgCo color transform or a YCgCo inverse color transform is applied to the current video block, and the rule specifying that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo color transform is used, a quantization parameter offset value different from −5 is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

FIG. **24** shows a flowchart for an example method **2400** of video processing. The method **2400** includes, at operation

2410, performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, the current video block being coded using a joint CbCr coding mode in which a YCgCo-R color transform or a YCgCo-R inverse color transform is applied to the current video block, and the rule specifying that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo-R color transform is used, a quantization parameter offset value different from a predetermined offset is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

A listing of solutions preferred by some embodiments is provided next.

A1. A method of video processing, comprising determining, for a conversion between a current video block of a video and a bitstream of the video, an allowed maximum size and/or an allowed minimum size for an adaptive color transform (ACT) mode used for coding the current video block; and performing, based on the determining, the conversion.

A2. The method of solution A1, wherein the allowed maximum size or the allowed minimum size for the ACT mode is based on an allowed maximum size or an allowed minimum size for a transform block or a transform skip block for a color component.

A3. The method of solution A2, wherein the color component is a luma component.

A4. The method of solution A1, wherein an indication of ACT mode is signaled in the bitstream based on the allowed maximum size or the allowed minimum size.

A5. The method of solution A1, wherein a side information related to the ACT mode is signaled in the bitstream based on the allowed maximum size or the allowed minimum size.

A6. The method of solution A1, wherein the allowed maximum size or the allowed minimum size is signaled in a sequence parameter set (SPS), a picture parameter set (PPS), a picture header, or a slice header.

A7. The method of solution A1, wherein the allowed maximum size or the allowed minimum size is signaled in the bitstream when the ACT mode is enabled.

A8. The method of solution A1, wherein the allowed maximum size is K0 and the allowed minimum size is K1, and wherein K0 and K1 are positive integers.

A9. The method of solution A8, wherein K0=64 and K1=32.

A10. The method of solution A1, wherein a difference between the allowed maximum size or the allowed minimum size for ACT coded blocks and corresponding sizes for transform blocks or transform skip blocks is signaled in the bitstream.

A11. A method of video processing, comprising determining, for a conversion between a current video block of a video and a bitstream of the video, a maximum allowed palette size and/or a minimum allowed predictor size for a palette mode used for coding the current video block; and performing, based on the determining, the conversion, wherein the maximum allowed palette size and/or the minimum allowed predictor size is based on a coding characteristic of the current video block.

A12. The method of solution A11, wherein the coding characteristic is a color component of the video, and wherein the maximum allowed palette sizes are different for different color components.

A13. The method of solution A11, wherein the coding characteristic is a quantization parameter.

A14. The method of solution A11, wherein S1 is the maximum allowed palette size or the minimum allowed predictor size associated with a first coding characteristic and S2 is the maximum allowed palette size or the minimum allowed predictor size associated with a second coding characteristic, and wherein S1 and S2 are positive integers.

A15. The method of solution A14, wherein S2 is greater than or equal to S1.

A16. The method of solution A14, wherein S1 and S2 are signaled separately in the bitstream.

A17. The method of solution A14, wherein S1 is signaled in the bitstream, and wherein S2 is inferred or derived from S1.

A18. The method of solution A17, wherein S2=S1−n, wherein n is a non-zero integer.

A19. The method of solution A14, wherein S1 and S2 are signaled at a first level and adjusted at a second level that is lower than the first level.

A20. The method of solution A19, wherein the first level is a sequence level, a picture level, or a slice level, and wherein the second level is a coding unit level or a block level.

A21. The method of solution A14, wherein S1 and S2 are based on whether luma mapping with chroma scaling (LMCS) is applied to the current video block.

A22. A method of video processing, comprising performing a conversion between a current video block of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the current video block is coded using a palette mode coding tool, and wherein the format rule specifies that parameters associated with a binarization of an escape symbol for the current video block in the bitstream is based on coded information of the current video block.

A23. The method of solution A22, wherein the coded information comprises one or more quantization parameters.

A24. The method of solution A22, wherein the binarization uses an exponential-Golomb binarization method with an order k, wherein k is a non-negative integer that is based on the coded information.

A25. A method of video processing, comprising determining, for a conversion between a video comprising a block and a bitstream of the video, that a size of the block is greater than a maximum allowed size for an adaptive color transform (ACT) mode; and performing, based on the determining, the conversion, wherein, in response to the size of the block being greater than the maximum allowed size for the ACT mode, the block is partitioned into multiple sub-blocks, and wherein each of the multiple sub-blocks share a same prediction mode, and the ACT mode is enabled at a sub-block level.

A26. A method of video processing, comprising performing a conversion between a current video block of a video and a bitstream of the video, wherein the bitstream conforms to a format rule that specifies that whether an indication of a usage of an adaptive color transform (ACT) mode on the current video block is signaled in the bitstream is based on at least one of a dimension of the current video block or a maximum allowed size for the ACT mode.

A27. The method of solution A26, wherein the indication is signaled due to a width of the current video block being smaller than or equal to m or a height of the current video block being smaller than or equal to n, wherein m and n are positive integers.

A28. The method of solution A26, wherein the indication is signaled due to a product of a width of the current video

block and a height of the current video block being smaller than or equal to m, wherein m is a positive integer.

A29. The method of solution A26, wherein the indication is not signaled due to a width of the current video block being greater than m or a height of the current video block being greater than n, wherein m and n are positive integers.

A30. The method of any of solutions A27 to A29, wherein m is predefined.

A31. The method of any of solutions A27 to A29, wherein m is derived based on a decoded message in a sequence parameter set (SPS), a picture parameter set (PPS), an adaptation parameter set (APS), a coding tree unit (CTU) row, a group of CTUs, a coding unit (CU), or the block.

A32. The method of solution A26, wherein the indication is not signaled and inferred to be zero.

Another listing of solutions preferred by some embodiments is provided next.

B1. A method of video processing, comprising performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that an adaptive color transform (ACT) mode for the current video unit is disabled.

B2. The method of solution B1, wherein the first flag comprises general constraints information for one or more pictures in a set of output layers of the video.

B3. The method of solution B1 or B2, wherein the first flag is no_act_constraint_flag, and wherein the second flag is sps_act_enabled_flag.

B4. The method of any of solutions B1 to B3, wherein the second flag is equal to zero due to the first flag being equal to one.

B5. The method of any of solutions B1 to B3, wherein the second flag equaling zero specifies that the ACT mode is disabled for the current video unit.

B6. The method of any of solutions B1 to B3, wherein the second flag can equal either zero or one due to the first flag being equal to zero.

B7. A method of video processing, comprising performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for the current video unit is disabled.

B8. The method of solution B7, wherein the first flag comprises general constraints information for one or more pictures in a set of output layers of the video.

B9. The method of solution B7 or B8, wherein the first flag is no_bdpcm_constraint_flag, and wherein the second flag is sps_bdpcm_enabled_flag.

B10. The method of any of solutions B7 to B9, wherein the second flag is equal to zero due to the first flag being equal to one.

B11. The method of any of solutions B7 to B9, wherein the second flag equaling zero specifies that the BDPCM mode is disabled for the current video unit.

B12. The method of any of solutions B7 to B9, wherein the second flag can equal either zero or one due to the first flag being equal to zero.

B13. A method of video processing, comprising performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a block-based delta pulse code modulation (BDPCM) mode for a chroma component of the current video unit is disabled.

B14. The method of solution B13, wherein the first flag comprises general constraints information for one or more pictures in a set of output layers of the video.

B15. The method of solution B13 or B14, wherein the first flag is no_bdpcm_chroma_constraint_flag, and wherein the second flag is sps_bdpcm_chroma_enabled_flag.

B16. The method of any of solutions B13 to B15, wherein the second flag is equal to zero due to the first flag being equal to one.

B17. The method of any of solutions B13 to B15, wherein the second flag equaling zero specifies that the BDPCM mode is disabled for the chroma component of the current video unit.

B18. The method of any of solutions B13 to B15, wherein the second flag can equal either zero or one due to the first flag being equal to zero.

B19. A method of video processing, comprising performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a palette mode for the current video unit is disabled.

B20. The method of solution B19, wherein the first flag comprises general constraints information for one or more pictures in a set of output layers of the video.

B21. The method of solution B19 or B20, wherein the first flag is no_palette_constraint_flag, and wherein the second flag is sps_palette_enabled_flag.

B22. The method of any of solutions B19 to B21, wherein the second flag is equal to zero due to the first flag being equal to one.

B23. The method of any of solutions B19 to B21, wherein the second flag equaling zero specifies that the palette mode is disabled for the current video unit.

B24. The method of any of solutions B19 to B21, wherein the second flag can equal either zero or one due to the first flag being equal to zero.

B25. A method of video processing, comprising performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream conforms to a format rule, wherein the format rule specifies whether a first flag is included in the bitstream, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) specifies that a reference picture resampling (RPR) mode for the current video unit is disabled.

B26. The method of solution B25, wherein the first flag comprises general constraints information for one or more pictures in a set of output layers of the video.

B27. The method of solution B25 or B26, wherein the first flag is no_ref_pic_resampling_constraint_flag, and wherein the second flag is ref_pic_resampling_enabled_flag.

B28. The method of any of solutions B25 to B27, wherein the second flag is equal to zero due to the first flag being equal to one.

B29. The method of any of solutions B25 to B27, wherein the second flag equaling zero specifies that the RPR mode is disabled for the current video unit.

B30. The method of any of solutions B25 to B27, wherein the second flag can equal either zero or one due to the first flag being equal to zero.

B31. The method of any of solutions B1 to B30, wherein the format rule further specifies that whether the first flag is signaled in the bitstream is based on a condition.

B32. The method of solution B31, wherein the condition is a type of a chroma format of the video.

Yet another listing of solutions preferred by some embodiments is provided next.

C1. A method of video processing, comprising performing a conversion between a current video block of a video and a bitstream of the video according to a rule that specifies additional quantization parameter offsets are applied when an adaptive color transform (ACT) mode is enabled for the current video block.

C2. The method of solution C1, wherein the additional quantization parameter offsets are applied after one or more chroma offsets are applied.

C3. The method of solution C2, wherein the one or more chroma offsets are specified in a picture parameter set (PPS), a picture header (PH), or a slice header (SH).

C4. A method of video processing, comprising performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, wherein the current video block is coded using a joint CbCr coding mode in which a YCgCo color transform or a YCgCo inverse color transform is applied to the current video block, and wherein the rule that specifies that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo color transform is used, a quantization parameter offset value different from −5 is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

C5. The method of solution C4, wherein the joint CbCr mode comprises a JCbCr mode 2.

C6. A method of video processing, comprising performing a conversion between a current video block of a video and a bitstream representation of the video according to a rule, wherein the current video block is coded using a joint CbCr coding mode in which a YCgCo-R color transform or a YCgCo-R inverse color transform is applied to the current video block, and wherein the rule that specifies that, due to the current video block being coded using the joint CbCr coding mode in which the YCgCo-R color transform is used, a quantization parameter offset value different from a predetermined offset is used in a picture header (PH) or a picture parameter set (PPS) associated with the current video block.

C7. The method of solution C6, wherein the joint CbCr mode comprises a JCbCr mode 2.

C8. The method of solution C6 or C7, wherein the predetermined offset is 1.

C9. The method of solution C6 or C7, wherein the predetermined offset is −1.

Yet another listing of solutions preferred by some embodiments is provided next.

P1. A method of video processing, comprising performing a determination whether to enable a chroma block-based delta pulse code modulation (BDPCM) mode for a video block of a video based on whether a usage of an adaptive color transform (ACT) mode and/or a luma BDPCM mode for the video block is enabled; and performing a conversion between the video block and a bitstream representation of the video according to the determinization.

P2. The method of solution P1, wherein a signaling for a first value for a first flag associated with enabling the chroma BDPCM mode is determined based on a signaling of the ACT mode being enabled for the video block and a signaling of a second value for a second flag associated with the usage of the luma BDPCM mode.

P3. The method of solution P2, wherein the first value for the first flag has a false value in response to the ACT mode being enabled and the second value for the second flag having a false value.

P4. The method of solution P2, wherein the first value for the first flag has a true value in response to the second value for the second flag having a true value.

P5. The method of solution P1, wherein a signaling of the ACT mode for the video block is conditionally based on a same BDPCM prediction direction being used for luma samples and chroma samples of the video block.

P6. The method of solution P5, wherein the signaling of the ACT mode is indicated after a signaling of the chroma BDPCM mode and the luma BDPCM mode.

P7. The method of solution P1, wherein in response to the usage of the ACT mode being enabled, a first value indicative of a first prediction direction of the chroma BDPCM mode is derived from a second value indicative of a second prediction direction of the luma BDPCM mode.

P8. The method of solution P7, wherein the first value indicative of the first prediction direction of the chroma BDPCM mode is same as the second value indicative of the second prediction direction of the luma BDPCM mode.

P9. The method of solution P8, wherein the first prediction direction of the chroma BDPCM mode and the second prediction direction of the luma BDPCM mode are in a horizontal direction.

P10. The method of solution P8, wherein the first prediction direction of the chroma BDPCM mode and the second prediction direction of the luma BDPCM mode are in a vertical direction.

P11. The method of solution P1, wherein in response to the usage of the ACT mode being disabled, a first value indicative of a first prediction direction of the chroma BDPCM mode is zero.

P12. A method of video processing, comprising performing a determination whether to enable a block-based delta pulse code modulation (BDPCM) mode for a video block of a video based on whether a usage of an adaptive color transform (ACT) mode for the video block is enabled; and performing a conversion between the video block and a bitstream representation of the video according to the determinization.

P13. The method of solution P12, wherein the BDPCM mode is disabled for the video block in response to the ACT mode being enabled for the video block.

P14. The method of solution P13, wherein a first flag indicative of the BDPCM mode is signaled after a second flag indicative of the ACT mode.

P15. The method of solution P13, wherein a flag indicative of the BDPCM mode is not signaled, wherein the flag is determined to be a false value or zero.

P16. The method of solution P12, wherein the ACT mode is disabled for the video block in response to the BDPCM mode being enabled for the video block.

P17. The method of solution P16, wherein a first flag indicative of the BDPCM mode is signaled before a second flag indicative of the ACT mode.

P18. The method of solution P16, wherein a flag indicative of the ACT mode is not signaled, wherein the flag is determined to be a false value or zero.

P19. The method of any of solutions P12 to P18, wherein the BDPCM mode includes a luma BDPCM mode and/or a chroma BDPCM mode.

P20. The method of solution P1, wherein the ACT mode is applied when the chroma BDPCM mode and the luma BDPCM mode are associated with different prediction modes.

P21. The method of solution P20, wherein a forward ACT mode is applied after the chroma BDPCM mode or the luma BDPCM mode.

P22. The method of any of solution P1 to P21, wherein a quantization parameter (QP) for the video block is clipped in response to the ACT mode being enabled.

P23. The method of solution P22, wherein a clipping function for clipping the QP is defined as (l, h, x), where l is a lowest possible value of an input x and h is a highest possible value of an input x.

P24. The method of solution P23, wherein l is equal to zero.

P25. The method of solution P23, wherein h is equal to 63.

P26. The method of solution P22, wherein the QP for the video block is clipped after the QP is adjusted for the ACT mode.

P27. The method of solution P23, wherein in response to a transform skip being applied to the video block, l is equal to a minimal allowed QP for a transform skip mode.

P28. The method of any of solution P23 to P26, wherein l, h, m, n and/or k are integer numbers that depend on (i) a message signaled in the DPS/SPS/VPS/PPS/APS/picture header/slice header/tile group header/Largest coding unit (LCU)/Coding unit (CU)/LCU row/group of LCUs/TU/PU block/Video coding unit, (ii) a position of CU/PU/TU/block/Video coding unit, (iii) coded modes of blocks containing the samples along the edges, (iv) transform matrices applied to the blocks containing the samples along the edges, (v) block dimension/Block shape of current block and/or its neighboring blocks, (vi) indication of the color format (such as 4:2:0, 4:4:4, RGB or YUV), (vii) a coding tree structure (such as dual tree or single tree), (viii) a slice/tile group type and/or picture type, (ix) a color component (e.g., may be only applied on Cb or Cr), (x) a temporal layer ID, or (xi) profiles/Levels/Tiers of a standard.

P29. The method of any of solution P23 to P26, wherein l, h, m, n and/or k are signaled to a decoder.

P30. The method of solution P30, wherein a color format is 4:2:0 or 4:2:2.

P31. The method of any of solutions P1 to P30, wherein an indication for the ACT mode or the BDPCM mode or the chroma BDPCM mode or the luma BDPCM mode is signaled in a sequence, a picture, a slice, a tile, a brick, or a video region-level.

The following technical solutions are applicable to any of the above solutions.

O1. The method of any of the preceding solutions, wherein the conversion comprises decoding the video from the bitstream representation.

O2. The method of any of the preceding solutions, wherein the conversion comprises encoding the video into the bitstream representation.

O3. The method of any of the preceding solutions, wherein the conversion comprises generating the bitstream from the current video unit, and the method further comprises storing the bitstream in a non-transitory computer-readable recording medium.

O4. A method of storing a bitstream representing a video to a computer-readable recording medium, comprising generating a bitstream from a video according to a method described in any of the preceding solutions; and writing the bitstream to the computer-readable recording medium.

O5. A video processing apparatus comprising a processor configured to implement a method recited in any of the preceding solutions.

O6. A computer-readable medium having instructions stored thereon, the instructions, when executed, causing a processor to implement a method recited in any of the preceding solutions.

O7. A computer readable medium that stores the bitstream representation generated according to any of the preceding solutions.

O8. A video processing apparatus for storing a bitstream representation, wherein the video processing apparatus is configured to implement a method recited in any of the preceding solutions.

O9. A bitstream that is generated using a method described herein, the bitstream being stored on a computer-readable medium.

In the present disclosure, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream.

The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this disclosure can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this disclosure and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored

in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

The processes and logic flows described in this disclosure can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically EPROM (EEPROM), and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and compact disc read-only memory (CD ROM) and digital versatile disc read-only memory (DVD-ROM) disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

While the present disclosure contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of the present disclosure. Certain features that are described in the present disclosure in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in the present disclosure should not be understood as requiring such separation in all embodiments.

Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in the present disclosure.

What is claimed is:

1. A method of processing video data, comprising:

performing a conversion between a current video unit of a video and a bitstream of the video,

wherein the bitstream conforms to a format rule,

wherein the format rule specifies whether a first flag is included in the bitstream, wherein the first flag comprises second general constraints information for one or more pictures in a set of output layers of the video, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) of the bitstream specifies that a pulse code modulation mode for the current video unit is disabled,

wherein the format rule further specifies whether a third flag is included in the bitstream, the third flag comprising general constraints information for the one or more pictures in the set of output layers of the video,

wherein the second flag is equal to zero in response to the third flag being equal to one, wherein the second flag being equal to zero specifies that the pulse code modulation mode is disabled for the current video unit,

wherein the third flag indicates whether a fourth flag in the SPS of the bitstream specifies that a resampling mode for the current video unit is disabled, and

wherein in response to the resampling mode being disabled, a current picture and a reference picture share same position offsets included in a picture parameter set (PPS), and the position offsets are used to calculate a scaling ratio.

2. The method of claim **1**, wherein the third flag is denoted as no_ref_pic_resampling_constraint_flag, and wherein the fourth flag is denoted as ref_pic_resampling_enabled_flag,

wherein the fourth flag is equal to zero in response to the third flag being equal to one, wherein the fourth flag being equal to zero specifies that the resampling mode is disabled for the current video unit, and

wherein the format rule further specifies whether the third flag is conditionally signaled in the bitstream.

3. The method of claim **1**, wherein the first flag is denoted as no_bdpcm_constraint_flag, and wherein the second flag is denoted as sps_bdpcm_enabled_flag.

4. The method of claim **1**, wherein the first flag indicates whether a fifth flag in the SPS specifies that the pulse code modulation mode for a chroma component of the current video unit is disabled.

5. The method of claim **1**, wherein the format rule further specifies whether a sixth flag is included in the bitstream, wherein the sixth flag comprises third general constraints information for the one or more pictures in the set of output layers of the video, and wherein the sixth flag indicates whether a seventh flag in the SPS specifies that a palette mode for the current video unit is disabled,

wherein the sixth flag is denoted as no_palette_constraint_flag, and wherein the seventh flag is denoted as sps_palette_enabled_flag, and

wherein the seventh flag is equal to zero in response to the sixth flag being equal to one, and wherein the seventh flag being equal to zero specifies that the palette mode is disabled for the current video unit.

6. The method of claim **1**, wherein the format rule further specifies whether an eighth flag is included in the bitstream, wherein the eighth flag comprises fourth general constraints information for the one or more pictures in the set of output

layers of the video, wherein the eighth flag indicates whether a ninth flag in the SPS specifies that a color transform mode for the current video unit is disabled, and wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the eighth flag is denoted as no_act_constraint_flag, and wherein the ninth flag is denoted as sps_act_enabled_flag, and

wherein the ninth flag is equal to zero in response to the eighth flag being equal to one, wherein the ninth flag being equal to zero specifies that the color transform mode is disabled for the current video unit.

7. The method of claim **1**, wherein the rule further specifies that additional quantization parameter offsets are further applied on intermedia quantization parameters, wherein the additional quantization parameter offsets are determined based on whether to apply a color transform mode for the current video block,

wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain, and

wherein the intermedia quantization parameters are derived from one or more chroma offsets specified in a picture parameter set (PPS) and a slice header (SH).

8. The method of claim **7**, wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cb component is defined as:

$$Qp'Cb=Clip3(-QpBdOffset,63,qPCb+pps_cb_qp_offset+sh_cb_qp_offset+CuQpOffsetCb)+QpBdOffset,$$

wherein the qPCb denotes an original quantization parameter for the Cb component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cb_qp_offset is the chroma offset specified in the PPS for the Cb component, sh_cb_qp_offset is the chroma offset specified in the SH for the Cb component, and CuQpOffsetCb denotes a variable derived in the PPS for the Cb component, and Clip3 is a clipping function.

9. The method of claim **7**, wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cr component is defined as:

$$Qp'Cr=Clip3(-QpBdOffset,63,qPCr+pps_cr_qp_offset+sh_cr_qp_offset+CuQpOffsetCr)+QpBdOffset,$$

wherein the qPCr denotes an original quantization parameter for the Cr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cr_qp_offset is the chroma offset specified in the PPS for the Cr component, sh_cr_qp_offset is the chroma offset specified in the SH for the Cr component, and CuQpOffsetCr denotes a variable derived in the PPS for the Cr component, and Clip3 is a clipping function.

10. The method of claim **7**, wherein an intermedia quantization parameter of the intermedia quantization parameters for a joint CbCr component is defined as:

$$Qp'CbCr=Clip3(-QpBdOffset,63,qPCbCr+pps_joint_cbcr_qp_offset+sh_joint_cbcr_qp_offset+CuQpOffsetCbCr)+QpBdOffset,$$

wherein the qPCbCr denotes an original quantization parameter for the joint CbCr component, QpBdOffset a bitdepth-based denotes quantization parameter offset, pps_joint_cbcr_qp_offset is the chroma offset specified in the PPS for the joint CbCr component, sh_joint_cbcr_qp_offset is the chroma offset specified in the SH for the joint CbCr component, and CuQpOffsetCbCr denotes a variable derived in the PPS for the joint CbCr component, and Clip3 is a clipping function,

wherein the rule further specifies that, due to the current video block being coded using a joint CbCr coding mode, the additional quantization parameter offset is not equal to −5,

wherein a mode index of the joint CbCr coding mode is equal to 2, and

wherein the color transform mode is applied on the current video block.

11. The method of claim **1**, wherein the conversion includes encoding the video into the bitstream.

12. The method of claim **1**, wherein the conversion includes decoding the video from the bitstream.

13. An apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to:

perform a conversion between a current video unit of a video and a bitstream of the video,

wherein the bitstream conforms to a format rule,

wherein the format rule specifies whether a first flag is included in the bitstream, wherein the first flag comprises second general constraints information for one or more pictures in a set of output layers of the video, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) of the bitstream specifies that a pulse code modulation mode for the current video unit is disabled,

wherein the format rule further specifies whether a third flag is included in the bitstream, the third flag comprising general constraints information for the one or more pictures in the set of output layers of the video,

wherein the second flag is equal to zero in response to the third flag being equal to one, wherein the second flag being equal to zero specifies that the pulse code modulation mode is disabled for the current video unit,

wherein the third flag indicates whether a fourth flag in the SPS of the bitstream specifies that a resampling mode for the current video unit is disabled, and

wherein in response to the resampling mode being disabled, a current picture and a reference picture share same position offsets included in a picture parameter set (PPS), and the position offsets are used to calculate a scaling ratio.

14. The apparatus of claim **13**,

wherein the third flag is denoted as no_ref_pic_resampling_constraint_flag, and wherein the fourth flag is denoted as ref_pic_resampling_enabled_flag,

wherein the fourth flag is equal to zero in response to the third flag being equal to one, wherein the fourth flag being equal to zero specifies that the resampling mode is disabled for the current video unit,

wherein the format rule further specifies whether the third flag is conditionally signaled in the bitstream,

wherein the first flag is denoted as no_bdpcm_constraint_flag, and wherein the second flag is denoted as sps_bdpcm_enabled_flag,

wherein the first flag indicates whether a fifth flag in the SPS specifies that the pulse code modulation mode for a chroma component of the current video unit is disabled,

wherein the format rule further specifies whether a sixth flag is included in the bitstream, wherein the sixth flag comprises third general constraints information for the one or more pictures in the set of output layers of the video, and wherein the sixth flag indicates whether a seventh flag in the SPS specifies that a palette mode for the current video unit is disabled,

wherein the sixth flag is denoted as no_palette_constraint_flag, and wherein the seventh flag is denoted as sps_palette_enabled_flag,

wherein the seventh flag is equal to zero in response to the sixth flag being equal to one, and wherein the seventh flag being equal to zero specifies that the palette mode is disabled for the current video unit,

wherein the format rule further specifies whether an eighth flag is included in the bitstream, wherein the eighth flag comprises fourth general constraints information for the one or more pictures in the set of output layers of the video, wherein the eighth flag indicates whether a ninth flag in the SPS specifies that a color transform mode for the current video unit is disabled, and wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the eighth flag is denoted as no_act_constraint_flag, and wherein the ninth flag is denoted as sps_act_enabled_flag,

wherein the ninth flag is equal to zero in response to the eighth flag being equal to one, wherein the ninth flag being equal to zero specifies that the color transform mode is disabled for the current video unit,

wherein the rule further specifies that additional quantization parameter offsets are further applied on intermedia quantization parameters, wherein the additional quantization parameter offsets are determined based on whether to apply a color transform mode for the current video block,

wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the intermedia quantization parameters are derived from one or more chroma offsets specified in the PPS and a slice header (SH),

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cb component is defined as:

$$Qp'Cb=Clip3(-QpBdOffset,63,qPCb+pps_cb_qp_offset+sh_cb_qp_offset+CuQpOffsetCb)+QpBdOffset,$$

wherein the qPCb denotes an original quantization parameter for the Cb component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cb_qp_offset is the chroma offset specified in the PPS for the Cb component, sh_cb_qp_offset is the chroma offset specified in the SH for the Cb component, and CuQpOffsetCb denotes a variable derived in the PPS for the Cb component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cr component is defined as:

$$Qp'Cr=Clip3(-QpBdOffset,63,qPCr+pps_cr_qp_offset+sh_cr_qp\ offset+CuQpOffsetCr)+QpBdOffset,$$

wherein the qPCr denotes an original quantization parameter for the Cr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cr_qp_offset is the chroma offset specified in the PPS for the Cr component, sh_cr_qp_offset is the chroma offset specified in the SH for the Cr component, and CuQpOffsetCr denotes a variable derived in the PPS for the Cr component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a joint CbCr component is defined as:

$$Qp'CbCr=Clip3(-QpBdOffset,63,qPCbCr+pps_joint_cbcr_qp_offset+sh_joint_cbcr_qp_offset+CuQpOffsetCbCr)+QpBdOffset,$$

wherein the qPCbCr denotes an original quantization parameter for the joint CbCr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_joint_cbcr_qp_offset is the chroma offset specified in the PPS for the joint CbCr component, sh_joint_cbcr_qp_offset is the chroma offset specified in the SH for the joint CbCr component, and CuQpOffsetCbCr denotes a variable derived in the PPS for the joint CbCr component, and Clip3 is a clipping function,

wherein the rule further specifies that, due to the current video block being coded using a joint CbCr coding mode, the additional quantization parameter offset is not equal to −5,

wherein a mode index of the joint CbCr coding mode is equal to 2, and

wherein the color transform mode is applied on the current video block.

15. A non-transitory computer-readable storage medium storing instructions that cause a processor to:

perform a conversion between a current video unit of a video and a bitstream of the video,

wherein the bitstream conforms to a format rule,

wherein the format rule specifies whether a first flag is included in the bitstream, wherein the first flag comprises second general constraints information for one or more pictures in a set of output layers of the video, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) of the bitstream specifies that a pulse code modulation mode for the current video unit is disabled,

wherein the format rule further specifies whether a third flag is included in the bitstream, the third flag comprising general constraints information for the one or more pictures in the set of output layers of the video,

wherein the second flag is equal to zero in response to the third flag being equal to one, wherein the second flag being equal to zero specifies that the pulse code modulation mode is disabled for the current video unit,

wherein the third flag indicates whether a fourth flag in the SPS of the bitstream specifies that a resampling mode for the current video unit is disabled, and

wherein in response to the resampling mode being disabled, a current picture and a reference picture share

same position offsets included in a picture parameter set (PPS), and the position offsets are used to calculate a scaling ratio.

16. The non-transitory computer-readable storage medium of claim 15,

wherein the third flag is denoted as no_ref_pic_resampling_constraint_flag, and wherein the fourth flag is denoted as ref_pic_resampling_enabled_flag,

wherein the fourth flag is equal to zero in response to the third flag being equal to one, wherein the fourth flag being equal to zero specifies that the resampling mode is disabled for the current video unit,

wherein the format rule further specifies whether the third flag is conditionally signaled in the bitstream,

wherein the first flag is denoted as no_bdpcm_constraint_flag, and wherein the second flag is denoted as sps_bdpcm_enabled_flag,

wherein the first flag indicates whether a fifth flag in the SPS specifies that the pulse code modulation mode for a chroma component of the current video unit is disabled,

wherein the format rule further specifies whether a sixth flag is included in the bitstream, wherein the sixth flag comprises third general constraints information for the one or more pictures in the set of output layers of the video, and wherein the sixth flag indicates whether a seventh flag in the SPS specifies that a palette mode for the current video unit is disabled,

wherein the sixth flag is denoted as no_palette_constraint_flag, and wherein the seventh flag is denoted as sps_palette_enabled_flag,

wherein the seventh flag is equal to zero in response to the sixth flag being equal to one, and wherein the seventh flag being equal to zero specifies that the palette mode is disabled for the current video unit,

wherein the format rule further specifies whether an eighth flag is included in the bitstream, wherein the eighth flag comprises fourth general constraints information for the one or more pictures in the set of output layers of the video, wherein the eighth flag indicates whether a ninth flag in the SPS specifies that a color transform mode for the current video unit is disabled, and wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the eighth flag is denoted as no_act_constraint_flag, and wherein the ninth flag is denoted as sps_act_enabled_flag,

wherein the ninth flag is equal to zero in response to the eighth flag being equal to one, wherein the ninth flag being equal to zero specifies that the color transform mode is disabled for the current video unit,

wherein the rule further specifies that additional quantization parameter offsets are further applied on intermedia quantization parameters, wherein the additional quantization parameter offsets are determined based on whether to apply a color transform mode for the current video block,

wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the intermedia quantization parameters are derived from one or more chroma offsets specified in the PPS and a slice header (SH),

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cb component is defined as:

$$Qp'Cb=Clip3(-QpBdOffset,63,qPCb+pps_cb_qp_offset+sh_cb_qp_offset+CuQpOffsetCb)+QpBdOffset,$$

wherein the qPCb denotes an original quantization parameter for the Cb component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cb_qp_offset is the chroma offset specified in the PPS for the Cb component, sh_cb_qp_offset is the chroma offset specified in the SH for the Cb component, and CuQpOffsetCb denotes a variable derived in the PPS for the Cb component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cr component is defined as:

$$Qp'Cr=Clip3(-QpBdOffset,63,qPCr+pps_cr_qp_offset+sh_cr_qp_offset+CuQpOffsetCr)+QpBdOffset,$$

wherein the qPCr denotes an original quantization parameter for the Cr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cr_qp_offset is the chroma offset specified in the PPS for the Cr component, sh_cr_qp_offset is the chroma offset specified in the SH for the Cr component, and CuQpOffsetCr denotes a variable derived in the PPS for the Cr component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a joint CbCr component is defined as:

$$Qp'CbCr=Clip3(QpBdOffset,63,qPCbCr+pps_joint_cbcr_qp_offset+sh_joint_cbcr_qp_offset+CuQpOffsetCbCr)+QpBdOffset,$$

wherein the qPCbCr denotes an original quantization parameter for the joint CbCr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_joint_cbcr_qp_offset is the chroma offset specified in the PPS for the joint CbCr component, sh_joint_cbcr_qp_offset is the chroma offset specified in the SH for the joint CbCr component, and CuQpOffsetCbCr denotes a variable derived in the PPS for the joint CbCr component, and Clip3 is a clipping function,

wherein the rule further specifies that, due to the current video block being coded using a joint CbCr coding mode, the additional quantization parameter offset is not equal to −5,

wherein a mode index of the joint CbCr coding mode is equal to 2, and

wherein the color transform mode is applied on the current video block.

17. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises:

generating a bitstream for a current video unit of a video, wherein the bitstream conforms to a format rule,

wherein the format rule specifies whether a first flag is included in the bitstream, wherein the first flag comprises second general constraints information for one or

more pictures in a set of output layers of the video, and wherein the first flag indicates whether a second flag in a sequence parameter set (SPS) of the bitstream specifies that a pulse code modulation mode for the current video unit is disabled,

wherein the format rule further specifies whether a third flag is included in the bitstream, the third flag comprising general constraints information for the one or more pictures in the set of output layers of the video,

wherein the second flag is equal to zero in response to the third flag being equal to one, wherein the second flag being equal to zero specifies that the pulse code modulation mode is disabled for the current video unit,

wherein the third flag indicates whether a fourth flag in the SPS of the bitstream specifies that a resampling mode for the current video unit is disabled, and

wherein in response to the resampling mode being disabled, a current picture and a reference picture share same position offsets included in a picture parameter set (PPS), and the position offsets are used to calculate a scaling ratio.

18. The non-transitory computer-readable recording medium of claim **17**,

wherein the third flag is denoted as no_ref_pic_resampling_constraint_flag, and wherein the fourth flag is denoted as ref_pic_resampling_enabled_flag,

wherein the fourth flag is equal to zero in response to the third flag being equal to one, wherein the fourth flag being equal to zero specifies that the resampling mode is disabled for the current video unit,

wherein the format rule further specifies whether the third flag is conditionally signaled in the bitstream,

wherein the first flag is denoted as no_bdpcm_constraint_flag, and wherein the second flag is denoted as sps_bdpcm_enabled_flag,

wherein the first flag indicates whether a fifth flag in the SPS specifies that the pulse code modulation mode for a chroma component of the current video unit is disabled,

wherein the format rule further specifies whether a sixth flag is included in the bitstream, wherein the sixth flag comprises third general constraints information for the one or more pictures in the set of output layers of the video, and wherein the sixth flag indicates whether a seventh flag in the SPS specifies that a palette mode for the current video unit is disabled,

wherein the sixth flag is denoted as no_palette_constraint_flag, and wherein the seventh flag is denoted as sps_palette_enabled_flag,

wherein the seventh flag is equal to zero in response to the sixth flag being equal to one, and wherein the seventh flag being equal to zero specifies that the palette mode is disabled for the current video unit,

wherein the format rule further specifies whether an eighth flag is included in the bitstream, wherein the eighth flag comprises fourth general constraints information for the one or more pictures in the set of output layers of the video, wherein the eighth flag indicates whether a ninth flag in the SPS specifies that a color transform mode for the current video unit is disabled, and wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the eighth flag is denoted as no_act_constraint_flag, and wherein the ninth flag is denoted as sps_act_enabled_flag,

wherein the ninth flag is equal to zero in response to the eighth flag being equal to one, wherein the ninth flag being equal to zero specifies that the color transform mode is disabled for the current video unit,

wherein the rule further specifies that additional quantization parameter offsets are further applied on intermedia quantization parameters, wherein the additional quantization parameter offsets are determined based on whether to apply a color transform mode for the current video block,

wherein in the color transform mode, for an encoding operation, visual signals are converted from a first color domain to a second color domain, or for a decoding operation, the visual signals are converted from the second color domain to the first color domain,

wherein the intermedia quantization parameters are derived from one or more chroma offsets specified in the PPS and a slice header (SH),

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cb component is defined as:

$$Qp'Cb=Clip3(-QpBdOffset,63,qPCb+pps_cb_qp_offset+sh_cb_qp_offset+CuQpOffsetCb)+QpBdOffset,$$

wherein the qPCb denotes an original quantization parameter for the Cb component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cb_qp_offset is the chroma offset specified in the PPS for the Cb component, sh_cb_qp_offset is the chroma offset specified in the SH for the Cb component, and CuQpOffsetCb denotes a variable derived in the PPS for the Cb component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a Cr component is defined as:

$$Qp'Cr=Clip3(-QpBdOffset,63,qPCr+pps_cr_qp_offset+sh_cr_qp_offset+CuQpOffsetCr)+QpBdOffset,$$

wherein the qPCr denotes an original quantization parameter for the Cr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_cr_qp_offset is the chroma offset specified in the PPS for the Cr component, sh_cr_qp_offset is the chroma offset specified in the SH for the Cr component, and CuQpOffsetCr denotes a variable derived in the PPS for the Cr component, and Clip3 is a clipping function,

wherein an intermedia quantization parameter of the intermedia quantization parameters for a joint CbCr component is defined as:

$$Qp'CbCr=Clip3(QpBdOffset,63,qPCbCr+pps_joint_cbcr_qp_offset+sh_joint_cbcr_qp_offset+CuQpOffsetCbCr)+QpBdOffset,$$

wherein the qPCbCr denotes an original quantization parameter for the joint CbCr component, QpBdOffset denotes a bitdepth-based quantization parameter offset, pps_joint_cbcr_qp_offset is the chroma offset specified in the PPS for the joint CbCr component, sh_joint_cbcr_qp_offset is the chroma offset specified in the SH for the joint CbCr component, and CuQpOff-

setCbCr denotes a variable derived in the PPS for the joint CbCr component, and Clip3 is a clipping function,

wherein the rule further specifies that, due to the current video block being coded using a joint CbCr coding mode, the additional quantization parameter offset is not equal to −5,

wherein a mode index of the joint CbCr coding mode is equal to 2, and

wherein the color transform mode is applied on the current video block.

* * * * *