



- (51) International Patent Classification:
H04N 19/70 (2014.01)
- (21) International Application Number:
PCT/CN2016/078859
- (22) International Filing Date:
8 April 2016 (08.04.2016)
- (25) Filing Language:
English
- (26) Publication Language:
English
- (30) Priority Data:
62/144,529 8 April 2015 (08.04.2015) US
62/145,578 10 April 2015 (10.04.2015) US
PCT/CN2015/078269 5 May 2015 (05.05.2015) CN
62/168,151 29 May 2015 (29.05.2015) US
62/173,030 9 June 2015 (09.06.2015) US
- (71) Applicant: **MEDIATEK INC.** [CN/CN]; No. 1, Dusing Rd. 1st, Science-Based Industrial Park, Hsin-Chu 300, Taiwan (CN).

- (72) Inventors: **YE, Jing**; 414 Galleria Dr. #1, San Jose, CA California 95134 (US). **KIM, Jungsun**; 3700 Casa Verde Street Apt. 1231, San Jose, CA California 95134 (US). **LIU, Shan**; 1155 Nevada Ave., San Jose, CA California 95125 (US). **ZHANG, Kai**; A1101, Dongfang Yujinxiang Building, No. 216 Tangli Road, Chaoyang District, Beijing (CN). **CHUANG, Tzu-Der**; 5F., No. 50, Zhuangjing 5th St., Zhubei City, Hsinchu County 302, Taiwan (CN). **ZHU, Weijia**; No. 1007, Unit 2, Building 2, zuoanpuyuan community, dongcheng District, Beijing (CN).
- (74) Agent: **WAN HUI DA INTELLECTUAL PROPERTY AGENCY**; Yiyuan Office Building, Friendship Hotel, No.1 Zhongguancun Street South, Haidian District, Beijing 100873 (CN).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,

[Continued on next page]

(54) Title: METHODS FOR PALETTE MODE CONTEXT CODING AND BINARIZATION IN VIDEO CODING

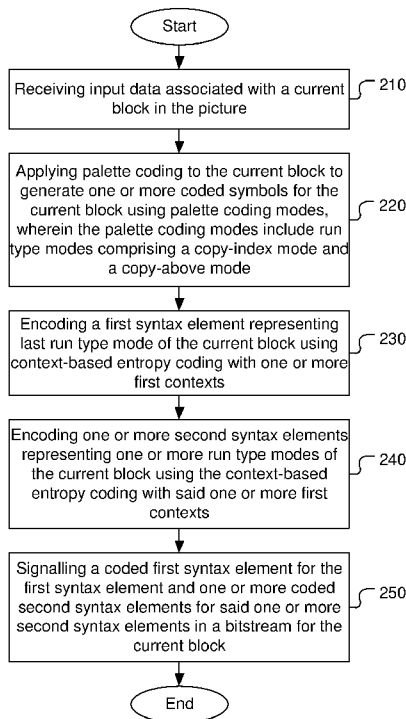


Fig. 2

(57) Abstract: A method of video coding using the palette coding mode with reduced complexity by sharing context(s) between context-based entropy coding of syntax element for the last run type mode and context-based entropy coding of syntax element for the run type mode is disclosed. A method with reduced complexity for binarization of the syntax element associated with the number of palette indices of the current block is disclosed, where Rice parameter is derived for the number of palette indices without the need for a division or table lookup operation. In another method, the syntax element to specify whether transpose process is applied to associated palette indices of the current block is signalled in the bitstream for the current block in a location after the syntax element to specify a last run type mode for the current block.

WO 2016/161974 A1



MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,

Published:

— *with international search report (Art. 21(3))*

METHODS FOR PALETTE MODE CONTEXT CODING AND BINARIZATION IN VIDEO CODING

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present invention claims priority to U.S. Provisional Patent Application, Serial No. 62/144,529, filed on April 8, 2015, U.S. Provisional Patent Application, Serial No. 62/145,578, filed on April 10, 2015, U.S. Provisional Patent Application, Serial No. 62/168,151, filed on May 29, 2015, U.S. Provisional Patent Application, Serial No. 62/173,030, filed on June 9, 2015, and PCT Patent Application, Serial No. PCT/CN2015/078269, filed on May 5, 2015. The U.S. Provisional Patent Applications and the PCT Patent Application are hereby incorporated by reference in their entireties.

FIELD OF THE INVENTION

[0002] The present invention relates to palette coding for video data. In particular, the present invention relates to various techniques for reducing complexity and/or increasing throughput related to syntax elements associated with palette coding.

BACKGROUND AND RELATED ART

[0003] High Efficiency Video Coding (HEVC) is a new coding standard that has been developed in recent years. In the High Efficiency Video Coding (HEVC) system, the fixed-size macroblock of H.264/AVC is replaced by a flexible block, named coding unit (CU). In recent HEVC development, palette-based coding is being used for screen content coding (SCC). The syntaxes for palette coding mainly consist of two parts. One is signalling palette table and the other is signalling the index map of the block (i.e., a CU) which represents the information of the block coded. For the index map, number of indices, last run_type flag and grouped indices are signalled. After signalling index information, a pair of run type and number of run is

Syntax element	binIdx					
	0	1	2	3	4	>= 5
palette_transpose_flag	0	na	na	na	na	na
num_palette_indices_idc	bypass	bypass	bypass	bypass	bypass	bypass
last_palette_run_type_flag	0	na	na	na	na	na
palette_run_type_flag	0	na	na	na	na	na
palette_index_idc	bypass	bypass	bypass	bypass	bypass	bypass

BRIEF SUMMARY OF THE INVENTION

[0007] A method of video coding using the palette coding mode with reduced complexity by sharing context(s) between context-based entropy coding of syntax element for the last run type mode and context-based entropy coding of syntax element for the run type mode is disclosed. In particular, the context-based entropy coding corresponds to context-adaptive binary coding (CABAC) and the CABAC coding for the syntax element corresponding to the last run type mode and the syntax element corresponding to the run type mode share one single context.

[0008] A method with reduced complexity for binarization of the syntax element associated with the number of palette indices of the current block is also disclosed. When the palette coding is applied to the current block, it generates one or more coded symbols corresponding to palette coding modes comprising a copy-index mode and a copy-above mode. The copy-index mode represents a number of consecutive indices all having the same index value by signalling the index value and the number of runs. The number of palette indices is associated with the index values signalled for the copy-index modes of the current block. A Rice parameter is derived for the number of palette indices and a Truncated Rice code is applied to the number of palette indices according to the derived Rice parameter to generate a prefix part. A conventional approach to the Rice parameter derivation requires a division operation. Embodiment of the present invention reduces the complexity by avoiding the division operation for Rice parameter derivation. For example, the Rice parameter can be

equal to $(2 + \text{MaxPaletteIndex}/M)$, where M is a power-of-2 integer to avoid the need for the division operation related to $\text{MaxPaletteIndex}/M$. According to JCTVC-T1005, the variable MaxPaletteIndex specifies the maximum possible value for a palette index for the current coding unit. The value of MaxPaletteIndex is set equal to $\text{CurrentPaletteSize} - 1 + \text{palette_escape_val_present_flag}$. In another example, the Rice parameter can be equal to $(2 + (\text{MaxPaletteIndex} \gg N))$, and where N is a positive integer and “ \gg ” corresponds to a right-shift operation. In yet another example, the Rice parameter can be equal to $((L + (\text{MaxPaletteIndex} \gg N))$, and where L and N are positive integers.

[0009] In yet another method, the syntax element to specify whether transpose process is applied to associated palette indices of the current block is signalled in the bitstream for the current block in a location after the syntax element to specify a last run type mode for the current block.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Fig. 1 illustrates an exemplary palette index map syntax signalling according to HEVC (High Efficiency Video Coding) based Screen Content Coding Test Module Version 4 (SCM-4.0).

[0011] Fig. 2 illustrates an exemplary flow chart of palette coding according to an embodiment of the present invention, where context-based entropy coding of syntax element for the last run type mode and context-based entropy coding of syntax element for the run type mode share the same context.

[0012] Fig. 3 illustrates an exemplary flow chart of palette coding according to an embodiment of the present invention, where the Rice parameter for binarization of the number of indices is derived with reduced complexity.

[0013] Fig. 4 illustrates an exemplary flow chart of palette coding according to an embodiment of the present invention, where the syntax element to specify whether transpose

process is applied to associated palette indices of the current block is signalled in the bitstream for the current block in a location after the syntax element to specify a last run type mode for the current block.

DETAILED DESCRIPTION OF THE INVENTION

[0014] The following description is of the best-contemplated mode of carrying out the invention. This description is made for the purpose of illustrating the general principles of the invention and should not be taken in a limiting sense. The scope of the invention is best determined by reference to the appended claims.

[0015] Last Palette Run Type Signalling

[0016] As shown in Fig. 1, a flag indicating whether the last run_type is copy_index or copy_above can be signalled before or after signalling indices. Also, the last run_type flag can be signalled before or after signalling the number of indices. The location of the last run_type can be anywhere in the bitstream for the block as long as it is signalled before the runs are signalled.

[0017] In order to improve the performance, in one embodiment of the present invention, the **last_palette_run_type_flag** can share the context of **palette_run_type_flag**. In the existing practice, both **last_palette_run_type_flag** and **palette_run_type_flag** are coded using context-based entropy coding. Furthermore, the context-based entropy coding for **last_palette_run_type_flag** and **palette_run_type_flag** uses different contexts. In this case, the parsing process for derivation of ctxInc for the syntax element **palette_run_type_flag** and **last_palette_run_type_flag** should be as below:

Inputs to this process are the current sample location (xC, yC) specifying the sample relative to the top-left luma sample of the picture.

Output of this process is the variable ctxInc.

[0018] The variable ctxInc is derived as follows: ctxInc = 0. The assignment of ctxInc to

syntax elements with context coded bins should be modified as below, where the text enclosed in a box indicates deletion. Since the **last_palette_run_type_flag** shares the context of **palette_run_type_flag**, there is no need of separate context for **last_palette_run_type_flag**.

Table 3

syntax element	binIdx					
	0	1	2	3	4	>= 5
...
palette_transpose_flag	0	na	na	na	na	na
num_palette_indices_idc	bypass	bypass	bypass	bypass	bypass	bypass
last_palette_run_type_flag	0	na	na	na	na	na
Same context for last_palette_run_type_flag and palette_run_type_flag	0	na	na	na	na	na
palette_index_idc	bypass	bypass	bypass	bypass	bypass	bypass

[0019] **Palette Run Type Signalling**

[0020] The signalling of the run type according to JCTVC-T1005 is shown in Fig. 1. In this application, various techniques to improve the coding efficiency related to run type signalling are disclosed.

[0021] **Run length based run_type coding**

[0022] In the **first embodiment**, the number of consecutive copy_index can be signalled instead of a series of 0/1 bits. After each number indicating consecutive copy_index is signalled, one syntax element for copy_above follows. A symbol, END is used to indicate the end of run_type coding, where a predefined value is used for END. When the signalled value is not END, the run length based run_type coding is repeated. The table below shows two simple examples incorporating the above embodiment.

Table 4

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Signalled values

(1)	000001000100101	1	5, 3, 2, 1, END
(2)	00000100010010	0	5, 3, 2, 1, END

[0023] Both example (1) and example (2) above show identical signalled values. These two cases are differentiated by the last run_type flag. If the last run_type flag is 1 (i.e., copy_above), the series of run types corresponds to example (1). If the last run_type flag is 0 (i.e., copy_index), the series of run types corresponds to example (2).

[0024] In the **second embodiment**, since the first pixel is always coded using the copy_index mode, the number of consecutive copy_index modes at the beginning should never be zero. Furthermore, there will be always a non-zero number of consecutive copy_index modes between two copy_above modes. Therefore, signalled values can be reduced by 1 to improve the coding efficiency. Examples incorporating the above embodiment are shown in the following table.

Table 5

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	# of consecutive copy index	Signalled values
(1)	000001000100101	1	5, 3, 2, 1	4,2,1,0, END
(2)	00000100010010	0	5, 3, 2, 1	4,2,1,0, END

[0025] In the **third embodiment**, when the number of indices is already known and the series of run types ends with consecutive copy_index (i.e., last copy_above can follow), the last part of consecutive copy_index can be derived as (number of indices – signalled number of copy_index) without signalling the number of last consecutive copy_index. If the last flag is 1, copy_index will be added as the last run type at final stage. Examples according to this embodiment are shown in the following table.

Table 6

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of indices	# of consecutive copy index	Signalled values
(1)	000001000100101	1	11	5, 3, 2, 1	4,2,1, END

(2)	00000100010010	0	11	5, 3, 2, 1	4,2,1, END
(3)	00000	0	5	5	END
(4)	000001	1	5	5	END
(5)	010000	0	5	1, 4	0, END

[0026] In the **fourth embodiment**, when the last value to be skipped (e.g. number of last consecutive copy_index) is smaller than the predefined number for END signal, the last value is signalled and END signal is skipped. For example, the predefined value for END can be 3 and examples according to this embodiment are shown in the following table.

Table 7

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of indices	# of consecutive copy index	Signalled values
(1)	000001000100101	1	11	5, 3, 2, 1	4,2,1,0
(2)	00000100010010	0	11	5, 3, 2, 1	4,2,1, 0
(3)	000001000000	0	11	5, 6	4, END

[0027] In the **fifth embodiment**, syntax element, END is signalled adaptively. When the number of indices is smaller than the predefined number for END, the predefined number is derived as (number of indices -1).

[0028] In the **sixth embodiment**, syntax element END is not signalled at all. Every number of consecutive copy_index is signalled. The accumulated number of copy_index and the number of indices are compared. If both numbers are identical, the parsing process stops. Therefore, signalling of END is not needed. Examples according to this embodiment are shown in the following table.

Table 8

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of indices	# of consecutive copy index	Signalled values
(1)	000001000100101	1	11	5, 3, 2, 1	4,2,1, 0
(2)	00000100010010	0	11	5, 3, 2, 1	4,2,1,0

(3)	000001000000	0	11	5, 6	4, 5
-----	--------------	---	----	------	------

[0029] In the **seventh embodiment**, syntax element for END is always signalled without signalling the number of indices. Number of indices can be derived from the first embodiment or the second embodiment without the need for signalling the number of indices. Examples according to this embodiment are shown in the following table.

Table 9

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	# of consecutive copy index	Signalled values
(1)	000001000100101	1	5, 3, 2, 1	4,2,1,0, END
(2)	00000100010010	0	5, 3, 2, 1	4,2,1,0, END
(3)	00000	0	5	4, END
(4)	000001	1	5	4, END

[0030] For examples (1) and (2) in the above table, the total number of copy_index modes can be derived as 11 after parsing the END signal. For examples (3) and (4), the total number of copy_index can be derived as 5.

[0031] In the **eighth embodiment**, when the number of signalled copy_above mode is already known, and the series of run types ends when the current number of copy_above mode is equal to the number of signalled copy_above mode. If the last flag is 0, the last copy_above mode is removed as shown in examples (2), (3) and (5) in the following table, where the removed modes are shown within double parentheses.

Table 10

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of copy_above mode	# of consecutive copy index	Signalled values
(1)	000001000100101	1	4	5, 3, 2, 1	4,2,1, 0
(2)	00000100010010((1))	0	4	5, 3, 2, 1	4,2,1, 0
(3)	00000((1))	0	1	5	4
(4)	000001	1	1	5	4

(5)	010000((1))	0	2	1, 4	0, 3
-----	-------------	---	---	------	------

[0032] In the **ninth embodiment**, when the number of index groups minus 1 is already known, the series of run types ends when the number of parsed index groups is equal to the signalled number of index groups. If the last flag is 1, a copy_index will be added as the last run type at final stage. Examples according to this embodiment are shown in the following table.

Table 11

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of index group - 1	# of consecutive copy index	Signalled values
(1)	000001000100101	1	3	5, 3, 2, 1	4,2,1, 0
(2)	00000100010010	0	3	5, 3, 2, 1	4,2,1, 0
(3)	00000	0	0	5	4
(4)	000001	1	0	5	4
(5)	010000	0	1	1, 4	0, 3

[0033] In the **tenth embodiment**, the value of END is predefined (e.g. 2). When the number of END minus 1 (the number of “2” minus 1) is already known, the series of run types ends when the number of parsed “2” is equal to the signalled number of END. If the last flag is 1, a copy_index will be added as the last run type at final stage. Examples according to this embodiment are shown below, where “2” in brackets corresponds to predefined value for END.

Table 12

	Run type (0 : copy_index, 1 : copy_above)	Last run type flag	Number of END - 1 (i.e., 2 - 1)	# of consecutive copy index	Signalled values
(1)	000001000100101	1	1	5, 3, 2, 1	4,2,1, 0, [2]
(2)	000001000010010	0	0	5, 4, 2, 1	4,3,1, 0, [2]
(3)	00000	0	0	5	4, [2]
(4)	000001	1	0	5	4, [2]
(5)	010000	0	0	1, 4	0, 3, [2]

[0034] **Binarization of number of consecutive copy_index**

[0035] In the above embodiments, various techniques to convert the run types for a block into coding elements (e.g. signalled run values). The value indicating the number of consecutive copy_index needs to be binarized. The binarized values may be further compressed by entropy coding before the values are included in the bitstream. Various binarization techniques are disclosed as follows.

[0036] In one embodiment, fixed-length code. When the number of indices is known, the maximum bit number required to code consecutive copy_index can be fixed. For example, if the maximum number of consecutive indices is 9, the number of bits required to represent each number of consecutive indices is 4. Therefore first signalled value (i.e., 4) in example (1) is a fixed 4-bit code, 0100. The second signalled value (i.e., 2) is coded as 0010. In another embodiment, the number of consecutive indices can be binarized using truncated binary codes.

[0037] In yet another embodiment, the number of bits can be decided by (the maximum number of consecutive indices -1) since the number of consecutive indices is always larger than 0. In example (2) of the following table, the number of bits is 3 and the first signalled value, 4 is coded as 100 and the second signalled value, 2 is coded as 010.

Table 13

	Run type (0 : copy_index, 1 : copy_above)	Number of indices	Signalled values
(1)	000001000101	9	4,2,0, END
(2)	000001000	8	4,2, END

[0038] In yet another embodiment, adaptive fixed-length code and truncated binary code are used. When the number of indices is known, the maximum number of bits required to code consecutive copy_index can be fixed. Also, after parsing each consecutive copy_index, the maximum number of bits can be reduced by the number of consecutive copy_index. In example (1) of Table 13 with the maximum number of consecutive indices as 9, the first signalled value 4 is represented by a fixed 4-bit code, 0100. The second signalled value 2 can be represented by a fixed 3-bit code, 010 since the available number of indices is 4 (i.e., 9 -

$(4+1) = 4$), which requires only 3 bits. Last signalled value 1 is coded by a fixed 1-bit code, 1. In another embodiment, the maximum number of bits can be determined by (number of indices – 1) since the number of copy_index between two copy_above is always larger than 0. In example (2) of Table 13, the number of bits is 3 and the first signalled value 4 is coded as 100 and the second signalled value 2 is coded as 10. For both embodiments, truncated binary coding can be used.

[0039] In yet another embodiment, N-th order Exponential Golomb code can be used to code the number of consecutive indices. In this case, a number of contexts can be used. The order of exponential Golomb code and the remaining number of indices after parsing consecutive copy_index can be taken into consideration for binarization.

[0040] Similar to the coding for the runs of copy_index mode or copy_above mode, the number of consecutive indices can also be coded with MSB part and remaining part. Remaining part can be coded using fixed-length codes or truncated binary codes depending on the remaining number of indices after parsing the consecutive copy_index. Also a number of contexts can be used.

[0041] **Context formation for palette run type**

[0042] In this application, the context formation of palette_run_type independent of the neighboring information or the previous decoded information is disclosed. In this case, the context formation of palette_run_type only depends on the scan order.

[0043] In one embodiment, the first palette run type uses an independent context set, and the rest palette run modes use other context sets.

[0044] In another embodiment, the first palette run mode uses a first context (e.g. context-0), and the rest palette run modes use a second context (e.g. context – 1).

[0045] **Binarization of number of indices (num_palette_indices_idc)**

[0046] In SCM-4.0 palette index map coding, the number of indices as indicated by

num_palette_indices_idc or similar syntax element is first signalled. To signal the number of indices, a variable (number of indices – palette size) is first derived. A mapping process is then performed to map the (number of indices – palette size) to a “mapped value”. The mapped value is binarized with the same binarization method as “coeff_abs_level_remaining” and signalled. The prefix part is coded using Truncated Rice (TR) code. The suffix part is coded using Exp-Golomb code. The Truncated Rice binarization process uses parameter, cRiceParam, to derive the prefix part. For this binarization method, the input cParam is set to be $2 + \text{MaxPaletteIndex}/6$, where the variable MaxPaletteIndex specifies the maximum possible value for a palette index for the current block. However, the $\text{MaxPaletteIndex}/6$ needs a division operation (i.e., division by 6) or a look up table operation. Accordingly, an embodiment of the present invention sets cParam to be equal to $2 + \text{MaxPaletteIndex}/M$ and M is a power-of-2 integer to avoid the need for the division operation. For example, M can be 4 or 8 and cParam can be derived as $(2 + (\text{MaxPaletteIndex} \gg 2))$ or $(2 + (\text{MaxPaletteIndex} \gg 3))$, where “ \gg ” corresponds to the right-shift operation.

[0047] In another embodiment, the division by 6 operation for the Rice parameter derivation is avoided by assigning the Rice parameter to $(2 + (\text{MaxPaletteIndex} \gg N))$, where N can be integer number smaller than $\log_2(\text{MaxPaletteIndex})$, such as 3.

[0048] In another embodiment, the Rice parameter cRiceParam can be determined according to $\text{cRiceParam} = (L + \text{MaxPaletteIndex}) \gg N$, N can be 0, 1, 2, ... , MaxPaletteIndex, L is an integer. For example, L and N can be set to 24 and 3 respectively, and cParam becomes $((24 + \text{MaxPaletteIndex}) \gg 3) = (3 + (\text{MaxPaletteIndex}) \gg 3)$.

[0049] It is also possible to binarize the syntax **num_palette_indices_idc** or similar syntax element using fixed-length binarization. It also can be binarized in a same way as palette_run, which is coded with or without contexts. It also can also be coded using N^{th} exponential Golomb code.

[0050] **Higher Throughput Coding for Palette Coding Mode**

[0051] For palette coding, various syntax elements such as palette index, run types, number of runs and escape values have to be signalled. Some of these syntax elements are coded using context-adaptive binary arithmetic codes (CABAC), where the CABAC includes a context mode (i.e., regular mode) and a bypass mode. The context mode is more compression efficient. However, the processing speed is much slower compared to the bypass mode. Various techniques to improve the throughput rate of palette coding are disclosed.

[0052] In one embodiment, the palette coding syntax elements coded in the bypass mode are grouped together. Also, the palette coding syntax elements coded with the context mode are grouped together. Therefore, when a group of context-coded syntax elements are decoded, the throughput rate should be improved since the arithmetic decoder does not have to switch between different modes.

[0053] In one embodiment, a palette coding syntax element coded in the bypass mode should not be placed in the bitstream between two palette coding syntax elements coded with coding contexts.

[0054] In palette coding, a syntax element, **palette_transpose_flag** is used to specify whether the transpose process is applied to the associated palette indices of the current coding unit. When **palette_transpose_flag** is equal to 1, the transpose process is applied to the associated palette indices of the current coding unit. When **palette_transpose_flag** is equal to 0, the transpose process is not applied to the associated palette indices of the current coding unit. One aspect of the present invention addresses the location of **palette_transpose_flag** in the coded data for the current block.

[0055] In one embodiment, the **palette_transpose_flag** should not be placed in the bitstream between two palette coding syntax elements coded with bypass mode.

[0056] In one embodiment, the **palette_transpose_flag** can be coded after the **last_palette_run_type_flag**. For example, the location of **palette_transpose_flag** in JCTVC-T1005 can be moved to the location after **last_palette_run_type_flag** as shown in the

Table 14 below, where the original **palette_transpose_flag** location is indicated by Note (14-1) and the new **palette_transpose_flag** location according to this embodiment is indicated by Note (14-2). In Table 14, where the text enclosed in a box indicates deletion.

Table 14

if(MaxPaletteIndex > 0) {	Note
palette_transpose_flag	(14-1)
num_palette_indices_idc	
for(i=0; i < NumPaletteIndices; i++) {	
palette_index_idc	
PaletteIndexIdc[i] = palette_index_idc	
}	
last_palette_run_type_flag	
palette_transpose_flag	(14-2)
}	

[0057] According to other embodiments, the **palette_transpose_flag** can be coded in other locations of the bitstream for the block. For example, the **palette_transpose_flag** can be coded after **palette_run_type_flag**, **palette_run_msb_id_plus**, or **palette_run_refinement_bits**.

[0058] In one embodiment, the **palette_transpose_flag** can be coded by the bypass mode or the **palette_transpose_flag** can be derived by the decoder implicitly.

[0059] In one embodiment, the **last_palette_run_type_flag** can be coded by in the bypass mode.

[0060] Fig. 2 illustrates an exemplary flow chart of palette coding according to an embodiment of the present invention, where context-based entropy coding of syntax element for the last run type mode and context-based entropy coding of syntax element for the run type mode share the same context. The encoder receives input data associated with a current block in the picture in step 210. Palette coding is applied to the current block to generate one or more coded symbols for the current block using palette coding modes in step 220, where the palette coding modes include run type modes comprising a copy-index mode and a copy-above mode.

A first syntax element representing last run type mode of the current block is encoded using context-based entropy coding with one or more first contexts in step 230. One or more second syntax elements representing one or more run type modes of the current block are encoded using the context-based entropy coding with said one or more first contexts in step 240. A coded first syntax element for the first syntax element and one or more coded second syntax elements for said one or more second syntax elements are signalled in a bitstream for the current block in step 250.

[0061] The decoder flowchart corresponding to the encoder in Fig. 2 can be derived using an inverse process of the flowchart in Fig. 2.

[0062] Fig. 3 illustrates an exemplary flow chart of palette coding according to an embodiment of the present invention, where the Rice parameter for binarization of the number of indices is derived with reduced complexity. The encoder receives input data associated with a current block in the picture in step 310. Palette coding is applied to the current block to generate one or more coded symbols for the current block using palette coding modes comprising a copy-index mode and a copy-above mode in step 320. The copy-index mode represents $N1$ consecutive indices all having one first index value by signalling said one first index value and $N1$ runs, and the copy-above mode represents $N2$ consecutive second index values all having same second index values as corresponding $N2$ above pixels by signalling $N2$ runs, and $N1$ and $N2$ are positive integers. A first number related to a total number of first index values to be signalled for one or more copy-index modes generated for the current block is identified in step 330. A Rice parameter is derived from a first variable specifying a maximum possible value for a palette index for the current block without using division operation or table lookup operation in step 340. The first number is binarized using Truncated Rice code with the Rice parameter to generate a prefix part in step 350.

[0063] The decoder flowchart corresponding to the encoder in Fig. 3 can be derived using an inverse process of the flowchart in Fig. 3.

[0064] Fig. 4 illustrates an exemplary flow chart of palette coding according to an

embodiment of the present invention, where the syntax element to specify whether transpose process is applied to associated palette indices of the current block is signalled in the bitstream for the current block in a location after the syntax element to specify a last run type mode for the current block. The encoder receives input data associated with a current block in the picture in step 410. Palette coding is applied to the current block to generate one or more coded symbols for the current block using palette coding modes in step 420. The palette coding modes include run type modes comprising a copy-index mode and a copy-above mode. A first syntax element to specify whether transpose process is applied to associated palette indices of the current block is signalled in step 430. A second syntax element to specify a last run type mode for the current block is signalled at a location of a bitstream for the current block after the first syntax element in step 440. The last run type mode specifies whether a last run type of the current block is the copy-index mode or the copy-above mode.

[0065] The decoder flowchart corresponding to the encoder in Fig. 4 can be derived using an inverse process of the flowchart in Fig. 4.

[0066] The above description is presented to enable a person of ordinary skill in the art to practice the present invention as provided in the context of a particular application and its requirement. Various modifications to the described embodiments will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed. In the above detailed description, various specific details are illustrated in order to provide a thorough understanding of the present invention. Nevertheless, it will be understood by those skilled in the art that the present invention may be practiced.

[0067] Embodiment of the present invention as described above may be implemented in various hardware, software codes, or a combination of both. For example, an embodiment of the present invention can be one or more electronic circuits integrated into a video compression

chip or program code integrated into video compression software to perform the processing described herein. An embodiment of the present invention may also be program code to be executed on a Digital Signal Processor (DSP) to perform the processing described herein. The invention may also involve a number of functions to be performed by a computer processor, a digital signal processor, a microprocessor, or field programmable gate array (FPGA). These processors can be configured to perform particular tasks according to the invention, by executing machine-readable software code or firmware code that defines the particular methods embodied by the invention. The software code or firmware code may be developed in different programming languages and different formats or styles. The software code may also be compiled for different target platforms. However, different code formats, styles and languages of software codes and other means of configuring code to perform the tasks in accordance with the invention will not depart from the spirit and scope of the invention.

[0068] The invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described examples are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

1. A method of palette index map decoding for blocks in a picture, comprising:
 - receiving input bitstream associated with a current block in the picture;
 - parsing a coded first syntax element and one or more coded second syntax elements from a bitstream for the current block;
 - decoding the coded first syntax element into a first syntax element using context-based entropy decoding with one or more first contexts, wherein the first syntax element represents last run type mode of the current block;
 - decoding said one or more coded second syntax elements into one or more second syntax elements using the context-based entropy decoding with said one or more first contexts, wherein said one or more second syntax elements represent one or more run type modes of the current block; and
 - applying palette decoding to the current block according to decoded information including the last run type mode of the current block and said one or more run type modes of the current block.
2. The method of Claim 1, wherein the context-based entropy decoding corresponds to context-adaptive binary decoding and said one or more first contexts correspond to one single context.
3. A method of palette index map coding for blocks in a picture, comprising:
 - receiving input data associated with a current block in the picture;
 - applying palette coding to the current block to generate one or more coded symbols for the current block using palette coding modes, wherein the palette coding modes include run type modes comprising a copy-index mode and a copy-above mode;
 - encoding a first syntax element representing last run type mode of the current block using context-based entropy coding with one or more first contexts;
 - encoding one or more second syntax elements representing one or more run type modes of the current block using the context-based entropy coding with said one or more

first contexts; and

signalling a coded first syntax element for the first syntax element and one or more coded second syntax elements for said one or more second syntax elements in a bitstream for the current block.

4. The method of Claim 3, wherein the context-based entropy coding corresponds to context-adaptive binary encoding and said one or more first contexts correspond to one single context.

5. A method of palette index map decoding for blocks in a picture, comprising:

receiving input bitstream associated with a current block in the picture;

deriving a Rice parameter from a first variable, MaxPaletteIndex, specifying a maximum possible value for a palette index for the current block without using division operation or table lookup operation;

parsing, from the input bitstream, a coded first number corresponding to a first number related to a total number of first index values signalled for one or more copy-index modes generated for the current block by applying palette coding to the current block, wherein the copy-index mode represents N consecutive indices all having one first index value by signalling said one first index value and N runs and N is a positive integer;

recovering the first number from the coded first number by applying decoding process including applying truncated Rice decoding with the Rice parameter to a prefix part of the coded first number; and

applying palette decoding to recover the current block using information including the first number.

6. The method of Claim 5, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $(2 + \text{MaxPaletteIndex}/M)$ and M is a power-of-2 integer to avoid a need for the division operation related to $\text{MaxPaletteIndex}/M$.

7. The method of Claim 5, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $(2 + (\text{MaxPaletteIndex} \gg N))$, and wherein N is a positive integer and “ \gg ” corresponds to a right-shift operation.
8. The method of Claim 5, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $((L + \text{MaxPaletteIndex}) \gg N)$, and wherein L and N are positive integers and “ \gg ” corresponds to a right-shift operation.
9. A method of palette index map coding for blocks in a picture, comprising:
- receiving input data associated with a current block in the picture;
 - applying palette coding to the current block to generate one or more coded symbols for the current block using palette coding modes comprising a copy-index mode and a copy-above mode, wherein the copy-index mode represents N1 consecutive indices all having one first index value by signalling said one first index value and N1 runs, and the copy-above mode represents N2 consecutive second indices all having same second index values as corresponding N2 above pixels by signalling N2 runs, and N1 and N2 are positive integers;
 - identifying a first number related to a total number of first index values to be signalled for one or more copy-index modes generated for the current block;
 - deriving a Rice parameter from a first variable specifying a maximum possible value for a palette index for the current block without using division operation or table lookup operation; and
 - binarizing the first number using Truncated Rice code with the Rice parameter to generate a prefix part.
10. The method of Claim 9, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $(2 + \text{MaxPaletteIndex}/M)$ and M is a power-of-2 integer to avoid a need for the division operation related to $\text{MaxPaletteIndex}/M$.

11. The method of Claim 9, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $(2 + (\text{MaxPaletteIndex} \gg N))$, and wherein N is a positive integer and “ \gg ” corresponds to a right-shift operation.
12. The method of Claim 9, wherein said deriving the Rice parameter from the first variable assigns the Rice parameter equal to $((L + \text{MaxPaletteIndex}) \gg N)$, and wherein L and N are positive integers and “ \gg ” corresponds to a right-shift operation.
13. A method of palette index map coding for blocks in a picture, comprising:
 - receiving input bitstream associated with a current block in the picture;
 - parsing a first syntax element to specify whether transpose process is applied to associated palette indices of the current block;
 - parsing a second syntax element to specify a last run type mode for the current block at a location of a bitstream for the current block after the first syntax element, wherein the last run type mode specifies whether a last run type of the current block is a copy-index mode or a copy-above mode, and wherein the current block is coded using palette coding modes including run type modes comprising the copy-index mode and the copy-above mode; and
 - applying palette decoding to the current block according to decoded information including whether the transpose process is applied to the associated palette indices of the current block and the last run type mode for the current block.
14. A method of palette index map coding for blocks in a picture, comprising:
 - receiving input data associated with a current block in the picture;
 - applying palette coding to the current block to generate one or more coded symbols for the current block using palette coding modes, wherein the palette coding modes include run type modes comprising a copy-index mode and a copy-above mode;
 - signalling a first syntax element to specify whether transpose process is applied to associated palette indices of the current block; and

signalling a second syntax element to specify a last run type mode for the current block at a location of a bitstream for the current block after the first syntax element, wherein the last run type mode specifies whether a last run type of the current block is the copy-index mode or the copy-above mode.

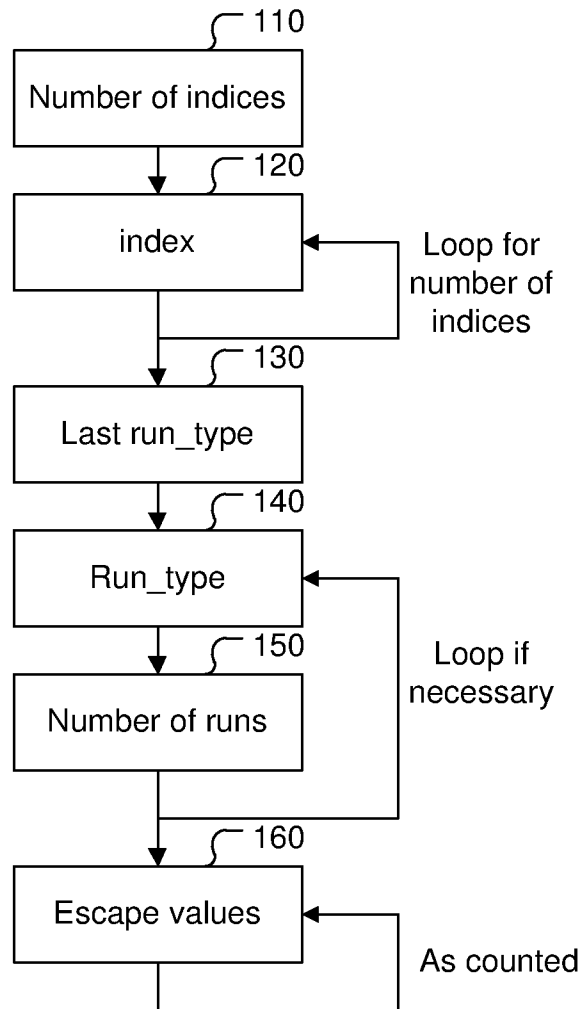
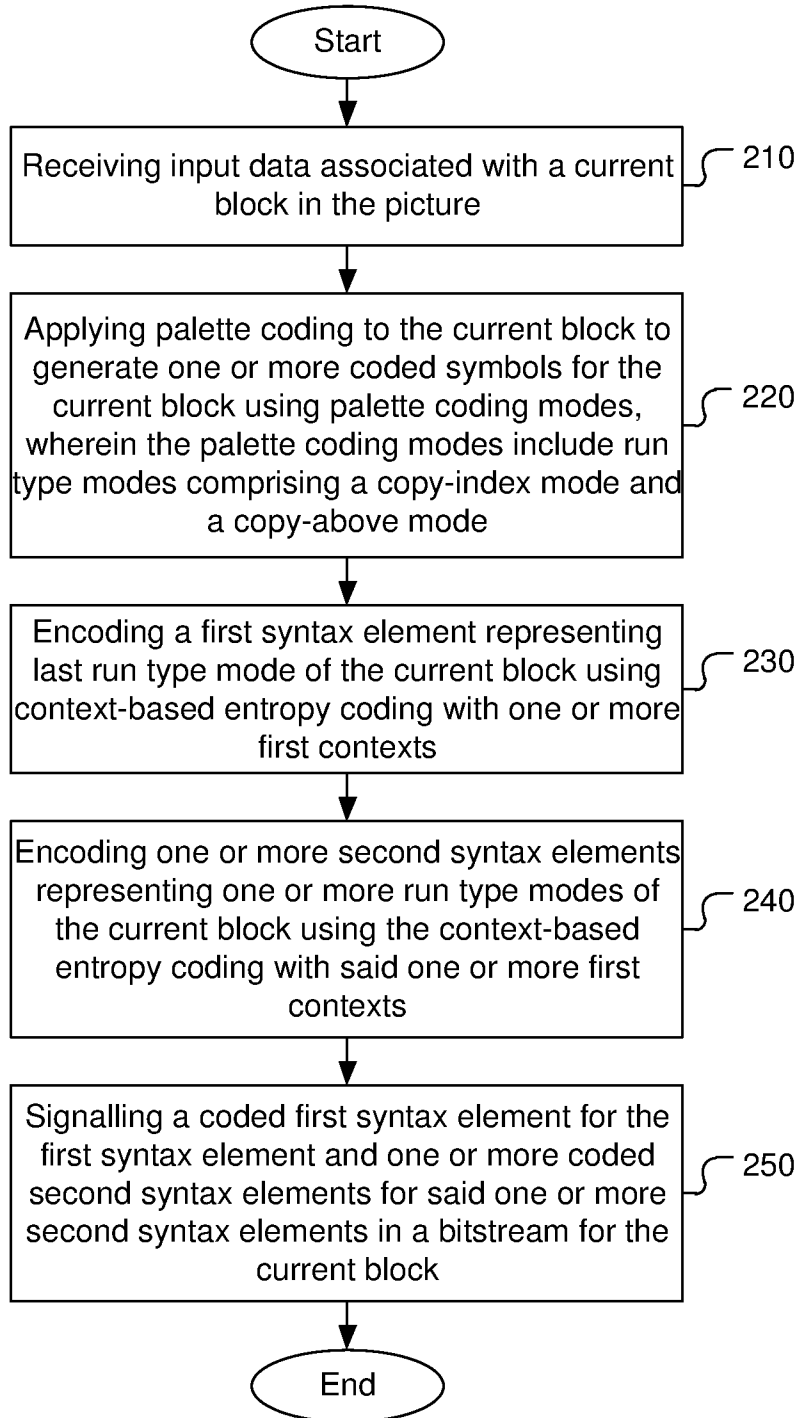
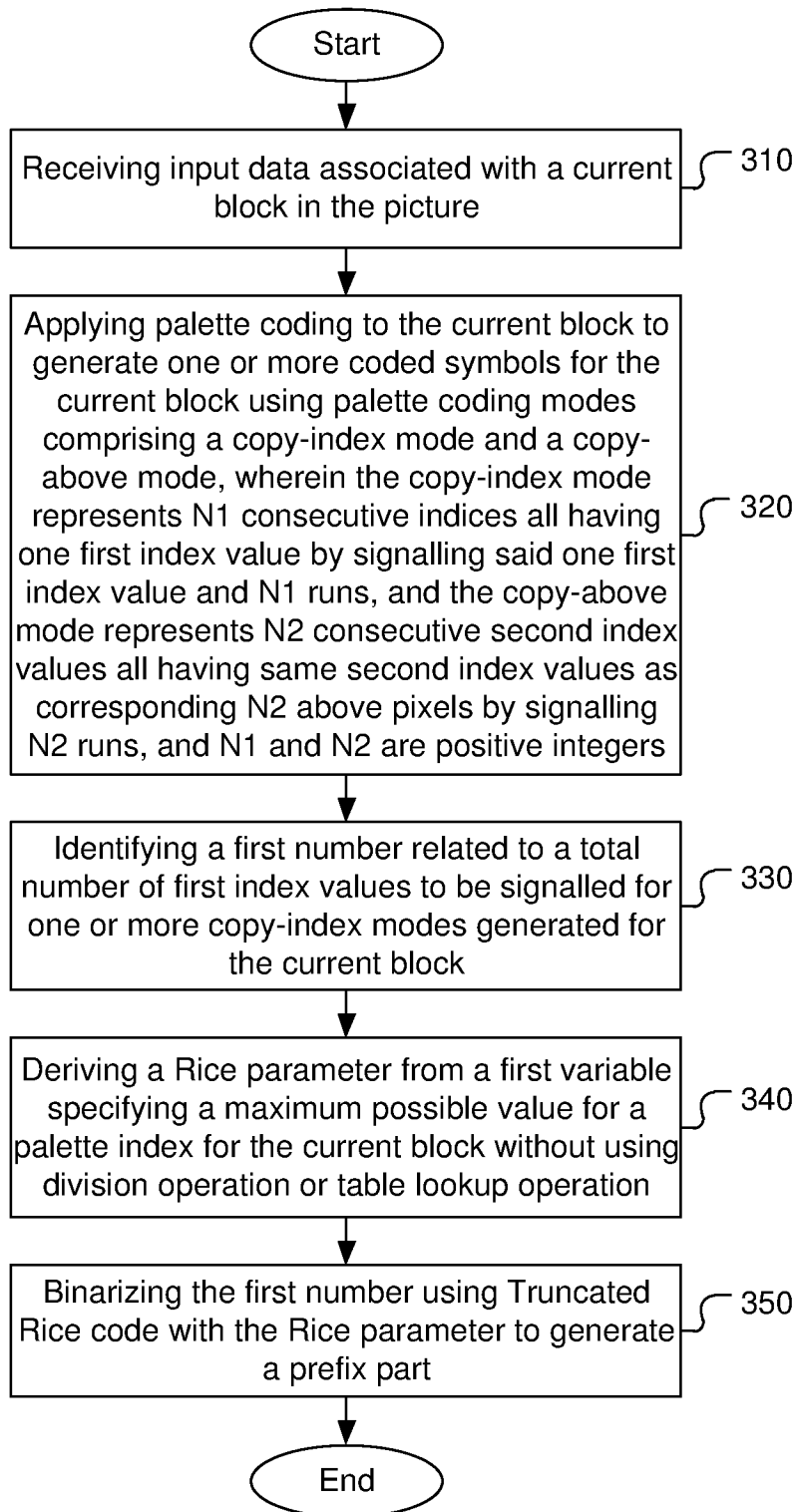
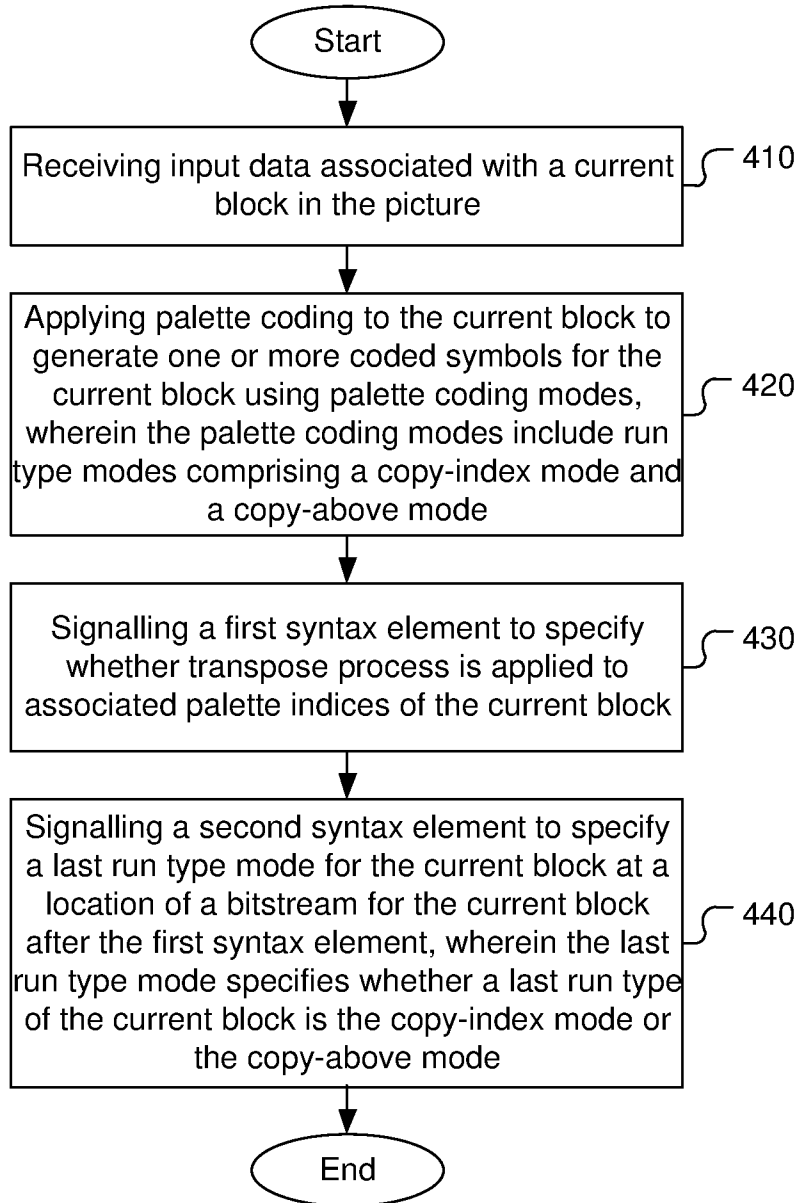


Fig. 1.

**Fig. 2**

**Fig. 3**

**Fig. 4**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/078859

A. CLASSIFICATION OF SUBJECT MATTER		
H04N 19/70(2014.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
H04N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
CNABS, CNTXT, CNKI, VEN, JCTVC: palette, cod+, encod+, run, context, syntax, run?type, transpose, flag, same, shar+, division, lookup, location		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Rajan Joshi, et al. "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 3" <i>JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 20th Meeting: Geneva, CH, Document: JCTVC-T1005</i> , 17 February 2015 (2015-02-17), the section 5.1, 7.3.3.8, 7.4.9.6, 9.3.3.15	5-14
A	Rajan Joshi, et al. "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 3" <i>JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 20th Meeting: Geneva, CH, Document: JCTVC-T1005</i> , 17 February 2015 (2015-02-17), the section 5.1, 7.3.3.8, 7.4.9.6, 9.3.3.15, 9.3.4.2.1	1-4
A	CN 104301737 A (HUAWEI TECHNOLOGIES CO., LTD.) 21 January 2015 (2015-01-21) the whole document	1-14
A	US 2014301474 A1 (QUALCOMM INC) 09 October 2014 (2014-10-09) the whole document	1-14
A	US 2015010053 A1 (SONY CORP) 08 January 2015 (2015-01-08) the whole document	1-14
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
21 June 2016		06 July 2016
Name and mailing address of the ISA/CN		Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China		YAO,Nan
Facsimile No. (86-10)62019451		Telephone No. (86-10)62089578

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/078859

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

[1] Claims 1,3; Claims 5, 9; Claims 13, 14.

[2] Claim 3 is corresponding to claim 1 about coding. The special technical features in the claims 1 and 3 are " decoding said one or more coded second syntax elements into one or more second syntax elements using the context-based entropy decoding with said one or more first contexts, wherein said one or more second syntax elements represent one or more run type modes of the current block" and " encoding one or more second syntax elements representing one or more run type modes of the current block using the context-based entropy coding with said one or more first contexts". However, claims 5, 9 and claims 13, 14 don't include the common or corresponding special technical features. In conclusion, the groups of claims in the invention are not linked by common or corresponding special technical features. The application, hence, does not meet the requirements of unity of invention as defined in Rules 13.1 and 13.2 PCT.

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

- Remark on Protest**
- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
 - The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
 - No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2016/078859

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	104301737	A	21 January 2015	WO	2015007207	A1	22 January 2015
US	2014301474	A1	09 October 2014	EP	2982125	A1	10 February 2016
				EP	2982126	A1	10 February 2016
				KR	20150140722	A	16 December 2015
				WO	2014165784	A1	09 October 2014
				CN	105075272	A	18 November 2015
				CN	105191320	A	23 December 2015
				KR	20150139902	A	14 December 2015
				US	2014301475	A1	09 October 2014
				WO	2014165789	A1	09 October 2014
US	2015010053	A1	08 January 2015	WO	2015006169	A1	15 January 2015