(51) International Patent Classification:
H04N 7/36 (2006.01)  H04N 7/26 (2006.01)

(21) International Application Number:
PCT/US2013/025988

(22) International Filing Date:
13 February 2013 (13.02.2013)

(25) Filing Language:  English
(26) Publication Language:  English

(30) Priority Data:
61/640,538 30 April 2012 (30.04.2012)  US
13/676,691 13 February 2013 (13.02.2013)  US

(71) Applicant: MOTOROLA MOBILITY LLC [US/US];
600 North US Highway 45, Libertyville, IL 60048 (US).

(72) Inventors: YU, Yue; 11175 Caminito Inocenta, San Diego, California 92130 (US); LOU, Jian; 5240 Fiore Ter, Apt # 413, San Diego, California 92122 (US); WANG, Limin; 13641 Shoal Summit Drive, San Diego, California 92128 (US).


ZM, ZW.


Published:
— with international search report (Art. 21(3))

(54) Title: SIGNALING OF TEMPORAL MOTION VECTOR PREDICTOR (MVP) FLAG FOR TEMPORAL PREDICTION

![FIG.6A](image-url)

(57) Abstract: In one embodiment, a method determines whether a flag that controls whether a temporal motion vector is used for slice is present in an encoded bitstream. If the flag is present, the method determines a value for the flag, and if the flag is not present, the method infers that the value for the flag is a default value. Then, the method evaluates whether to determine a collocated reference picture in decoding the slice from the encoded bitstream based on the value of the flag.
SIGNALING OF TEMPORAL MOTION VECTOR PREDICTOR (MVP) FLAG FOR TEMPORAL PREDICTION

CROSS REFERENCE TO RELATED APPLICATIONS

The present disclosure claims priority to U.S. Provisional App. No. 61/640,538 for "Modification for Signaling Collocated Picture for HEVC" filed April 30, 2012, and U.S. Provisional App. No. 61/623,036 for "Modification for Signaling Collocated Picture for HEVC" filed April 11, 2012, the contents of both of which are incorporated herein by reference in its entirety.

BACKGROUND

[0001] Video compression systems employ block processing for most of the compression operations. A block is a group of neighboring pixels and may be treated as one coding unit in terms of the compression operations. Theoretically, a larger coding unit is preferred to take advantage of correlation among immediate neighboring pixels. Various video compression standards, e.g., Motion Picture Expert Group (MPEG)-1, MPEG-2, and MPEG-4, use block sizes of 4x4, 8x8, and 16x16 (referred to as a macroblock (MB)).

[0002] High efficiency video coding (HEVC) is also a block-based hybrid spatial and temporal predictive coding scheme. HEVC partitions an input picture into square blocks referred to as coding tree units (CTUs) as shown in FIG. 1. Unlike prior coding standards, the CTU can be as large as 128x128 pixels. Each CTU can be partitioned into smaller square blocks called coding units (CUs). FIG. 2 shows an example of a CTU partition of CUs. A CTU 100 is first partitioned into four CUs 102. Each CU 102 may also be further split into four smaller CUs 102 that are a quarter of the size of the CU 102. This partitioning process can be repeated based on certain criteria, such as limits to the number of times a CU can be partitioned may be imposed. As shown, CUs 102-1, 102-3, and 102-4 are a quarter of the size of CTU 100. Further, a CU 102-2 has been split into four CUs 102-5, 102-6, 102-7, and 102-8.
Each CU 102 may include one or more blocks, which may be referred to as prediction units (PUs). FIG. 3 shows an example of a CU partition of PUs. The PUs may be used to perform spatial prediction or temporal prediction. A CU can be either spatially or temporally predictive coded. If a CU is coded in intra mode, each PU of the CU can have its own spatial prediction direction. If a CU is coded in inter mode, each PU of the CU can have its own motion vector(s) and associated reference picture(s).

In HEVC, motion vectors (MVs) are predictively coded in a spatial/temporal prediction process. For a current PU having one current motion vector and an associated reference index, a motion vector predictor (MVP) is derived from motion vectors of spatially neighboring or temporally collocated PUs of the current PU. The difference between the current motion vector and the MVP is then determined and coded. This reduces overhead as only the difference is sent instead of information for the current motion vector. Also, when in merge mode, a single motion vector may be applied to a group of spatially neighboring or temporally collocated PUs.

Given a current PU in a current picture, an associated collocated PU resides in an associated collocated reference picture. The collocated PU is used as one of the candidates for the MVP or in a merge/skip mode for the current PU. The encoder typically encodes information for a flag to indicate whether temporal motion vectors should be used in the decoding process. Encoding information for the flag incurs overhead in the encoded bitstream.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows an input picture partitioned into square blocks referred to as coding tree units (CTUs).

FIG. 2 shows an example of a CTU partition of CUs.

FIG. 3 shows an example of a CU partition of PUs.
[0009] FIG. 4 depicts an example of a system for encoding and decoding video content according to one embodiment.

[0010] FIG. 5A depicts an example of the coding process when the flag enable_temporal_mvp_flag is not present according to one embodiment.

[0011] FIG. 5B depicts a second example for a coding process using the flag enable_temporal_mvp_flag according to one embodiment.

[0012] FIG. 6A depicts a simplified flowchart of a method for encoding video according to one embodiment.

[0013] FIG. 6B depicts a simplified flowchart of a method for decoding video according to one embodiment.

[0014] FIG. 7A shows an example of a syntax for evaluating whether to use the flag collocated_from_to_flag and the syntax collocated_ref_idx according to one embodiment.

[0015] FIG. 7B depicts an example of the encoder and the decoder according to one embodiment.

[0016] FIG. 7C shows a second example of the encoder and the decoder according to one embodiment.

[0017] FIG. 8 depicts a simplified flowchart of a method for encoding or decoding a current block according to one embodiment.

[0018] FIG. 9 depicts a more detailed flowchart of an encoding process according to one embodiment.

[0019] FIG. 10 depicts a simplified flowchart of a method for decoding a current block according to one embodiment.

[0020] FIG. 11A depicts an example of encoder according to one embodiment.

[0021] FIG. 11B depicts an example of decoder according to one embodiment.
DETAILED DESCRIPTION

[0022] Described herein are techniques for a video compression system. In the following description, for purposes of explanation, numerous examples and specific details are set forth in order to provide a thorough understanding of particular embodiments. Particular embodiments as defined by the claims may include some or all of the features in these examples alone or in combination with other features described below, and may further include modifications and equivalents of the features and concepts described herein.

[0023] In one embodiment, a method determines whether a flag that controls whether a temporal motion vector is used for slice is present in an encoded bitstream. If the flag is present, the method determines a value for the flag, and if the flag is not present, the method infers that the value for the flag is a default value. Then, the method evaluates whether to determine a collocated reference picture in decoding the slice from the encoded bitstream based on the value of the flag.

[0024] In one embodiment, a decoder includes: one or more computer processors; and a non-transitory computer-readable storage medium comprising instructions that, when executed, control the one or more computer processors to be configured for: determining whether a flag that controls whether a temporal motion vector is used for slice is present in an encoded bitstream; if the flag is present, determining a value for the flag; if the flag is not present, inferring that the value for the flag is a default value; and evaluating whether to determine a collocated reference picture in decoding the slice from the encoded bitstream based on the value of the flag.

[0025] In one embodiment, a method includes: determining, by a computing device, whether a temporal motion vector should be used for encoding slice; determining, by the computing device, whether a flag that controls whether a temporal motion vector should be used for encoding the slice; if the flag should be used, encoding, by the computing device, a value for the flag in an encoded bitstream for the slice; if the flag should not be used, not encoding, by the computing device, the value for the flag in
the encoded bitstream; and sending, by the computing device, the encoded bitstream to a decoder, wherein the decoder infers that the value of the flag is a default value when the value for the flag is not present in the encoded bitstream.

[0026] In one embodiment, an encoder includes: one or more computer processors; and a non-transitory computer-readable storage medium comprising instructions that, when executed, control the one or more computer processors to be configured for: determining whether a temporal motion vector should be used for encoding slice; determining whether a flag that controls whether a temporal motion vector should be used for encoding the slice; if the flag should be used, encoding a value for the flag in an encoded bitstream for the slice; if the flag should not be used, not encoding the value for the flag in the encoded bitstream; and sending the encoded bitstream to a decoder, wherein the decoder infers that the value of the flag is a default value when the value for the flag is not present in the encoded bitstream.

Overview

[0027] FIG. 4 depicts an example of a system 400 for encoding and decoding video content according to one embodiment. Encoder 402 and decoder 403 may encode and decode a bitstream using HEVC; however, other video compression standards may also be appreciated.

[0028] Encoder 402 includes a collocated reference picture manager 404-1 and decoder 403 includes a collocated reference picture manager 404-2. Collocated reference picture manager 404 determines a reference picture to use as a collocated reference picture for a current picture. The collocated reference picture includes motion vector (MV) information that can be used in a temporal prediction process. For example, a temporal motion vector for a partition in the collocated reference picture may be used as a temporal motion vector predictor (MVP) in advanced motion vector prediction (AMVP) or as a temporal merge/skip candidate motion vector in a merge process. When the motion vector of the collocated block is used as a temporal MVP, a difference of the motion vector of the collocated block and the current motion vector of the current block is coded and sent from encoder 402 to decoder 403. When
the motion vector of the collocated block is used in a merge/skip process, the current block and the collocated block are merged and use the same motion vector of the collocated block.

[0029] The temporal MVP or temporal merge/skip candidate MV is defined as the MV of the selected collocated block, which resides in a reference picture and has the same geometrical position as a current block in the current picture. In one example, given a PU in a current picture, an associated collocated PU can reside in a preceding or subsequent PU in time. The collocated PU can be used as a candidate to enable MVP and merge/skip mode for the current PU. Like with a PU, a collocated block can be provided for a transform unit (TU), CU or a picture itself. Subsequent discussion will reference a picture, although the same could apply to any of the listed subcomponents of a picture or a slice. Additionally, a block may be referenced, which may be a PU, but may also be a TU, CU, a picture, or a slice.

[0030] A flag collocated_from_iO_flag is used to specify which list (e.g., listO and listl) includes the collocated reference picture. For example, when the flag collocated_from_iO_flag is equal to 1, then the collocated reference picture shall be derived from listO. Otherwise, the collocated reference picture shall be derived from list 1, such as when the flag collocated_from_iO_flag is set to 0. When the flag collocated_from_iO_flag is not present in the encoded bitstream, decoder 403 may infer that the flag collocated_from_iO_flag is equal to 1 or set the flag collocated_from_iO_flag to 1.

[0031] A syntax collocated_ref_idx specifies the index for the collocated reference picture in either listO or listl (or a combined list). For example, the syntax collocated_ref_idx may indicate a position in listO or listl. Depending on the value of the flag collocated_from_iO_flag, the syntax collocated_ref_idx may refer to listO if the flag collocated_from_iO_flag is equal to 1; otherwise, the syntax collocated_ref_idx refers to listl. In one embodiment, encoder 402 explicitly signals information for the flag collocated_from_iO_flag and syntax collocated_ref_idx to decoder 403 in the encoded bitstream if certain conditions are met. However, if other conditions are met, encoder 402 may not explicitly signal information for the flag
collocated_from_to_flag and syntax_collocated_ref_idx to decoder 403 in the encoded bitstream, which saves bits and reduces overhead. The conditions will be described in more detail below.

[0032] A flag, referred to as enable_temporal_mvp_flag, is used to control whether the temporal motion vector is used or not in encoding (or decoding) a current block (or current slice or picture). For example, if the flag enable_temporal_mvp_flag is enabled (e.g., equal to 1), then temporal motion vector predictors may be used to encode (or decode) a current block (or blocks in a slice or picture). When a temporal MVP is used, a motion vector from a collocated block may be included in a group of candidate motion vectors, which may also include motion vectors from spatially located blocks. A motion vector is selected from the candidate group to use in encoding or decoding the current block. In another case, in a merge/skip mode, the temporal motion vector may be used as a motion vector for a group of blocks. Otherwise, if the flag enable_temporal_mvp_flag is not enabled (e.g., equal to 0), temporal motion vector are not used in encoding (or decoding) a current block (or slice of blocks/pictures).

[0033] The flag enable_temporal_mvp_flag may be set for a slice, picture, series of pictures, or a single block. For example, when the flag enable_temporal_mvp_flag is set for a slice, the flag applies to all blocks being encoded within the slice.

[0034] The following will describe a process where a default value is inferred for the flag when a value for the flag is not present in the encoded bitstream. Then, a process for evaluating whether to use the syntax elements of the flag collocated_from_to_flag and the syntax collocated_ref_idx.
**Signaling of Temporal Motion Vector Predictor (MVP) Flag for Temporal Prediction**

[0035] Particular embodiments may not encode a value for the flag enable_temporal_mvp_flag in the encoded bitstream. When decoder 403 determines that the flag enable_temporal_mvp_flag is not present in the encoded bitstream, decoder 403 infers that the value for the flag enable_temporal_mvp_flag is equal to a default value. For example, decoder 403 may determine that the flag enable_temporal_mvp_flag is enabled or equal to 1. Other values may also be inferred, such as the flag enable_temporal_mvp_flag may be disabled or equal to 0.

[0036] FIG. 5A depicts an example of the coding process when the flag enable_temporal_mvp_flag is not present according to one embodiment. Encoder 402 encodes video into an encoded bitstream that is sent to decoder 403. In this example, the encoded bitstream does not include the flag enable_temporal_mvp_flag.

[0037] When decoder 403 receives the encoded bitstream, collocated reference picture manager 404-2 in decoder 403 determines that the flag enable_temporal_mvp_flag is not present in the encoded bitstream. For example, if the flag enable_temporal_mvp_flag is associated with one or more blocks being decoded (e.g., a current block, a slice of blocks, a picture, or a series of pictures), collocated reference picture manager 404-2 may look for the flag enable_temporal_mvp_flag in a header (e.g., a slice header, picture parameter set (PPS) header, or other headers). If the flag enable_temporal_mvp_flag is not present in the applicable header, collocated reference picture manager 404-2 infers that the value for the flag enable_temporal_mvp_flag is a default value. For example, collocated reference picture manager 404-2 infers that the value for the flag enable_temporal_mvp_flag is equal to 1. In this case, decoder 403 may set the value for the flag enable_temporal_mvp_flag to 1. Then, temporal motion vectors are used in determining the MVP or MV for merge/skip mode for a current block (or any blocks associated with the flag enable_temporal_mvp_flag, such as a slice of blocks).
[0038] FIG. 5B depicts a second example for a coding process using the flag enable_temporal_mvp_flag according to one embodiment. As shown, encoder 402 sends an encoded bitstream that includes the flag enable_temporal_mvp_flag. For example, encoder 402 sends the flag enable_temporal_mvp_flag in a slice header for a slice of blocks in a picture. However, the flag enable_temporal_mvp_flag may be sent at other levels, such as for a series of pictures at the picture parameter set (PPS) level or at a block level.

[0039] When collocated reference picture manager 404-2 at decoder 403 receives the encoded bitstream and determines that the flag enable_temporal_mvp_flag is present (e.g., present in the slice header), collocated reference picture manager 404-2 then determines the value of the flag enable_temporal_mvp_flag. For example, the value of the flag may indicate that use of temporal motion vectors should be enabled or disabled (e.g., 0 or 1). Collocated reference picture manager 404-2 then sets the value for the flag enable_temporal_mvp_flag and uses the value to decode the encoded bitstream. For example, depending on the value, temporal motion vectors may or may not be used in decoding a current block. In this case, the flag enable_temporal_mvp_flag is set to 0 in the encoded bitstream and collocated reference picture manager 404-2 sets the value to 0.

[0040] The following describes the encoding and decoding processes in more detail.

FIG. 6A depicts a simplified flowchart 600 of a method for encoding video according to one embodiment. At 602, encoder 402 encodes a current block. For example, a current block may be encoded in a slice of blocks within a picture. At 604, encoder 402 determines a value for the flag enable_temporal_mvp_flag. For example, depending on characteristics of the video, collocated reference picture manager 404-1 may determine that temporal motion vectors should or should not be used. Sometimes, temporal motion vectors are not used because the temporal similarities between pictures are not great. In order to increase the ability of error resilience, some encoders also do not want to use temporal motion vectors for some pictures.

[0041] At 606, encoder 402 determines whether to encode a value for the flag enable_temporal_mvp_flag. For example, when encoder 402 determines that the
value for the flag enable_temporal_mvp_flag corresponds to the default value, encoder 402 does not encode the value for the flag.

[0042] At 608, if encoder 402 determines that the value for the flag enable_temporal_mvp_flag should not be encoded, encoder 402 does not encode a value for the flag enable_temporal_mvp_flag in the encoded bitstream. For example, a value for the flag enable_temporal_mvp_flag is not included in a slice header (or at any other level). At 610, if encoder 402 determines that the value should be encoded, encoder 402 encodes a value for the flag enable_temporal_mvp_flag in the encoded bitstream. As discussed above, a value for the flag enable_temporal_mvp_flag may be encoded in a slice header (or at any other level). At 612, encoder 402 sends the encoded bitstream to decoder 403.

[0043] FIG. 6B depicts a simplified flowchart 650 of a method for decoding video according to one embodiment. At 652, decoder 403 receives an encoded bitstream from encoder 402. At 654, decoder 403 determines whether the flag enable_temporal_mvp_flag is present in the encoded bitstream. For example, decoder 403 may look for a value of the flag enable_temporal_mvp_flag in a slice header. At 656, if the flag enable_temporal_mvp_flag is not present, decoder 403 infers the value for the flag enable_temporal_mvp_flag to be equal to a default value. For example, decoder 403 may be programmed to set the value for the flag enable_temporal_mvp_flag to be equal to 1 when the flag enable_temporal_mvp_flag is not present. At 658, if the flag enable_temporal_mvp_flag is present, decoder 403 determines the value of the flag from the encoded bitstream and sets the value for the flag enable_temporal_mvp_flag.

[0044] At 660, decoder 403 decodes the encoded bitstream based on the value of the flag enable_temporal_mvp_flag. For example, if the value of the flag is 0, then temporal motion vectors are not used to decode the current block. However, if the value of the flag enable_temporal_mvp_flag is 1, then temporal motion vectors are used to decode the current block.
Accordingly, by inferring that the value of the flag enable_temporal_mvp_flag is equal to a default value, bits may be saved in the encoded bitstream. For example, when encoder 402 determines that the value for the flag enable_temporal_mvp_flag is equal to the default value, then encoder 402 does not need to send a bit for the flag enable_temporal_mvp_flag in the encoded bitstream. This is because decoder 403 is configured to infer that the value for the flag enable_temporal_mvp_flag is a default value when the flag enable_temporal_mvp_flag is not present in the encoded bitstream. In this case, decoder 403 implicitly determines the value for the flag enable_temporal_mvp_flag without communication with encoder 402.

Particular embodiments may set the default value based on a desired coding characteristic. For example, setting the default value to be equal to 1 means that temporal motion vectors will be used when the flag enable_temporal_mvp_flag is not present in the encoded bitstream. This may increase the computational complexity for decoding a current block. However, the decoding of the current block may be more efficient. However, by setting the default value to be 0, then temporal MVPs are not used by default when the flag enable_temporal_mvp_flag is not present in the encoded bitstream. This may decrease the complexity when decoding a current block. Additionally, the efficiency of the video compression may be decreased.

In one embodiment, the following syntax may be evaluated using the above process of inferring that the flag enable_temporal_mvp_flag is a default value. For example, if decoder 403 determines that the flag enable_temporal_mvp_flag in the encoded bitstream and then evaluates the syntax below based on setting the value of the flag enable_temporal_mvp_flag.

**Evaluation of Signaling of Collocated reference picture for Temporal Prediction**

In one embodiment, if the flag enable_temporal_mvp_flag is equal to 0, syntax elements related to the collocated reference picture can be ignored. This is because temporal motion vector predictors are not going to be used for encoding or decoding any blocks that are associated with the flag enable_temporal_mvp_flag,
such as blocks in a slice or blocks in pictures that are associated with the enable_temporal_mvp_flag. Thus, in one embodiment, the syntax elements of the flag collocated_from_10_flag and the syntax collocated_ref_idx are not needed. In this case, encoder 402 may not encode any information for the flag collocated_from_10_flag and the syntax collocated_ref_idx in the encoded bitstream. Further, decoder 403 does not look to decode any information for the flag collocated_from_10_flag and the syntax collocated_ref_idx in the encoded bitstream when collocated reference picture manager 404-2 detects that the flag enable_temporal_mvp_flag is 0. Conventionally, even if the flag enable_temporal_mvp_flag is 0, decoder 403 would decode information for the flag collocated_from_10_flag and the syntax collocated_ref_idx.

[0049] Particular embodiments may use a syntax to evaluate conditions to determine whether to use the flag collocated_from_10_flag and the syntax collocated_ref_idx. FIG. 7A shows an example of a syntax 700 for evaluating whether to use the flag collocated_from_10_flag and the syntax collocated_ref_idx according to one embodiment. At 702, a syntax tests whether to use a flag collocated_from_10_flag that is shown at 704. The syntax tests whether a prediction type (e.g., the syntax slicejtype) is equal to a B type and the flag enable_temporal_mvp_flag is enabled, such as equal to "1". In this case, the prediction type is a slice type. The value of "1" may indicate that temporal motion vectors should be used. Although the value of "1" is described, other values may be used. The syntax slice_type may be the prediction type assigned to a series of blocks (e.g., a slice) in a picture. Although a slice type is described, the syntax slicejtype may refer to a prediction type for a block, a picture, or a series of pictures.

[0050] The syntax slice_type for a slice of blocks in a picture may be a prediction type for a slice (e.g., slice type) of an I type, a P type, or a B type. An I slice only uses intra-prediction and does not use temporal MVPs. The I slice only includes blocks that use intra prediction. Also, a P slice may include both I type blocks and P type blocks. Blocks in a P slice only use one motion vector. A B slice may include B, P, and I type blocks. A B type block may use two motion vectors. Thus, the
syntax at 702 tests whether the syntax slice_type is a B type, which means some blocks in the slice may use one of listO and listl to determine the collocated reference picture. In this case, encoder 402 may need to signal which list to use the flag collocated_from_10_flag. Additionally, the syntax at 702 tests whether the flag enable_temporal_mvp_flag has been enabled (e.g., is set to 1). When both of these conditions are true, then encoder 402 may encode a value for the flag collocated_from_10_flag in the encoded bitstream. For example, encoder 402 may set the flag collocated_from_10_flag to a value to indicate which list of listO or listl is used. Further, decoder 403 may decode the flag collocated_from_10_flag in the encoded bitstream to determine which list of listO and listl to use when decoding a current block.

[0051] In the syntax at 702, the flag collocated_from_10_flag is only set when the flag collocated_from_10_flag is needed. That is, if the slice_type is an I type or a P type, then blocks in the slice will either not use a collocated reference picture (e.g., I type) or will use a collocated reference picture from only listO (e.g., P type). When the predication type is P type, and the flag collocated_from_10_flag is not included in the encoded bitstream, decoder 403 assumes the value of flag collocated_from_10_flag is 1 and may set the value of the flag collocated_from_10_flag to 1 by default. Additionally, when the flag enable_temporal_mvp_flag is not enabled, which means a temporal MVP is not used, then a flag indicating which list of listO and listl to use is not needed because decoder 403 will not need to determine a reference picture from listO or listl due to a temporal MVP not being used.

[0052] At 706, a syntax tests whether to use a syntax collocated_ref_idx that is shown at 708. In the syntax at 706, collocated reference picture manager 404 determines if the syntax slice_type associated with a current block is not equal to an I type (i.e., the syntax slice_type is equal to a B type or a P type). Also, collocated reference picture manager 404 determines if the flag enable_temporal_mvp_flag is enabled (e.g., equal to 1). In the above syntax at 706, collocated reference picture manager 404 is determining whether one of listO or listl will be used in the encoding
or decoding process. That is, if the \texttt{slice\_type} is a P type or a B type and a temporal MVP is going to be used, then a collocated reference picture in listO or listl will be used. Then, collocated reference picture manager 404 determines whether the syntax \texttt{collocated\_ref\_idx} should be used to identify a reference picture in listO or listl. For example, in either listO or listl (i.e., whichever one is being used based on the flag \texttt{collocated\_from\_listO\_flag}), if only 1 reference picture is found in either listO or listl, then the syntax \texttt{collocated\_ref\_idx} does not need to be used. Rather, in either listO or listl, encoder 402 or decoder 403 knows to use the only available reference picture in either listO or listl. However, if more than 1 reference picture is found in either listO or listl, then particular embodiments may use the syntax \texttt{collocated\_ref\_idx} to identify the reference picture to use in either listO or listl.

\begin{verbatim}
[0053] Accordingly, in one embodiment, collocated reference picture manager 404 checks whether the flag \texttt{collocated\_from\_listO\_flag} is equal to 1 (e.g., listO should be used) and whether the syntax \texttt{num\_ref\_idx\_listO\_active\_minusl > 0} is true. This means that listO is used and also the number of reference pictures in listO that apply to the current block is greater than 1 (or the number of reference pictures in listO minus 1 is > 0). If this evaluates to true, then collocated reference picture manager 404 uses the syntax \texttt{collocated\_ref\_idx}. For example, encoder 402 sets the syntax \texttt{collocated\_ref\_idx} to a position in listO and decoder 403 decodes the syntax \texttt{collocated\_ref\_idx} to determine the position in listO. As discussed above, the position identifies the reference picture to use as the collocated reference picture.

[0054] Also, collocated reference picture manager 404 determines if the flag \texttt{collocated\_from\_listl\_flag} is not equal to 1 (e.g., equal to 0 and indicating that listl should be used) and the syntax \texttt{num\_ref\_idx\_listl\_active\_minusl > 0} is greater than 0. In this case, collocated reference picture manager 404 is evaluating whether listl should be used and also if the number of reference pictures in listl that apply to the current block is greater than 1 (or the number of reference pictures in listl minus 1 is > 0). If the above statement evaluates to true, then the syntax \texttt{collocated\_ref\_idx} is used where the syntax indicates a position in listl. For example, encoder 402 sets the syntax \texttt{collocated\_ref\_idx} to a position in listl and decoder 403 decodes the syntax
\end{verbatim}
collocated_ref_idx to determine the position in listl. As discussed above, the position identifies the reference picture to use as the collocated reference picture.

[0055] Accordingly, as shown in the syntax at 706, the syntax collocated_ref_idx is used when the slice type is not equal to an I type, and whenever listO is used and the number of references pictures in listO is greater than 1 or when listl is used and the number of reference pictures in listl is greater than 1.

[0056] FIG. 6 depicts an example of encoder 402 and decoder 403 according to one embodiment. When the flag enable_temporal_mvp_flag is enabled (equal to 1) and other conditions may apply as described above, the encoded bitstream includes the flag enable_temporal_mvp_flag, the flag collocated_from_I0_flag, and a syntax collocated_ref_idx. In this case, encoder 402 sends the encoded bitstream including flag enable_temporal_mvp_flag, the flag collocated_from_I0_flag, and syntax collocated_ref_idx to decoder 403. In one embodiment, encoder 402 sends the flag collocated_from_I0_flag and syntax collocated_ref_idx to decoder 403 in a slice header in the encoded bitstream. In other embodiments, encoder 402 sends the flag collocated_from_I0_flag and syntax collocated_ref_idx in a picture parameter set (e.g., for a series of pictures). It should be noted that this illustration is a simplification and other conditions may apply to determine whether to include the flag collocated_from_I0_flag and syntax collocated_ref_idx as was described above and also will be described below.

[0057] FIG. 7 shows a second example of encoder 402 and decoder 403 according to one embodiment. As shown, the flag enable_temporal_mvp_flag is not enabled (equal to 0). This means that a temporal MVP will not be used in the encoding process for a current block. In this case, encoder 402 sends an encoded bitstream that includes the flag enable_temporal_mvp_flag, but not the flag collocated_from_I0_flag and the syntax collocated_ref_idx. Decoder 403 receives the encoded bitstream, determines that the flag enable_temporal_mvp_flag is not enabled and thus knows that the flag collocated_from_I0_flag and the syntax collocated syntax collocated_ref_idx are not included in the encoded bitstream and are not used to decode the current block.
A general method for encoding or decoding a current block will be described and then separate methods for encoding and then decoding a current block are described. FIG. 8 depicts a simplified flowchart 800 of a method for encoding or decoding a current block according to one embodiment. Although a current block is discussed, the process may be performed for multiple blocks. For example, the flag enable_temporal_mvp_flag may be associated with multiple blocks in a slice. At 802, collocated reference picture manager 404 determines a current block being encoded or decoded. For example, blocks in a current picture may be encoded or decoded.

At 804, collocated reference picture manager 404 determines a value for the flag enable_temporal_mvp_flag that controls whether a temporal motion vector is used for encoding or decoding the current block. For example, collocated reference picture manager 404 determines the value for the flag enable_temporal_mvp_flag that is associated with the current block. In one embodiment, the current block may be associated with a slice and the slice is associated with a value of the flag enable_temporal_mvp_flag.

At 806, collocated reference picture manager 404 determines if the enable_temporal_mvp_flag is a first value or a second value. The first value may be a value of 1 where the temporal motion vector is used for the current block and the second value may be a value of 0, which means that the temporal motion vector is not used for the current block.

At 808, collocated reference picture manager 404 determines whether to use the flag collocated_from_10_flag. For example, if the flag enable_temporal_mvp_flag is equal to 0, then collocated reference picture manager 404 does not use the flag collocated_from_10_flag. Also, if the syntax slice_type associated with the current block being encoded or decoded is a P type or I type, then the flag collocated_from_10_flag may not be used. However, if the syntax slice_type associated with the current block being encoded or decoded is a B type and the flag enable_temporal_mvp_flag is 1, then, collocated reference picture manager 404 uses the flag collocated_from_10_flag.
At 810, collocated reference picture manager 404 then determines if the syntax collocated_ref_idx should be used. In one embodiment, if the syntax slice_type is not equal to an I type and whichever of listO or listl that is being used has more than 1 reference picture in the list, then collocated reference picture manager 404 uses the syntax collocated_ref_idx.

FIG. 9 depicts a more detailed flowchart 900 of an encoding process according to one embodiment. At 902, encoder 402 determines a slice type and the value of the flag enable_temporal_mvp_flag. At 904, encoder 402 evaluates the condition "if(slice_type == B && enable_temporal_mvp_flag)". If this condition is true, then at 906 encoder 402 sets the flag collocated_from_10_flag to a value depending on whether listO or list1 should be used to encode the current block. If the condition evaluated at 904 is not true, then at 908, encoder 402 sets the flag collocated_from_10_flag to a default value, such as 1. However, encoder 402 still may not encode the value of the flag in the encoded bitstream.

At 910, encoder 402 determines the value of the flag collocated_from_10_flag and the number of pictures in listO or listl. At 912, encoder 402 then evaluates the condition "if(slice_type != I && enable_temporal_mvp_flag && ((collocated_from_10_flag && num_ref_idx_10_active_minusl > 0) || (!collocated_from_10_flag && num_ref_idx_11_active_minusl > 0))". If the above condition evaluates to true, then at 914 encoder 402 sets the syntax collocated_ref_idx to a value. For example, the syntax collocated_ref_idx is set to a position in one of listO or listl that is being used. If the above condition does not evaluate to true, then at 916 encoder 402 sets the syntax collocated_ref_idx to a default value, such as 0. However, encoder 402 still may not encode the value of the syntax collocated_ref_idx in the encoded bitstream.

At 918, encoder 402 encodes the current block based on the above evaluations. For example, depending on the above evaluations, the flag collocated_from_10_flag and syntax collocated_ref_idx may or may not be encoded in the bitstream with the encoded block.
FIG. 10 depicts a simplified flowchart 1000 of a method for decoding a current block according to one embodiment. At 1002, decoder 403 determines a slice type and value for the flag enable_temporal_mvp_flag for a current block being decoded in an encoded bitstream received from encoder 402.

At 1004, decoder 403 evaluates a first condition "if(slice_type == B && enable_temporal_mvp_flag)". If the first condition evaluates to true, then at 1006 decoder 403 decodes the flag collocated_from_10_flag in the encoded bitstream. If the above condition does not evaluate to true, then at 1008 decoder 403 determines that the flag collocated_from_10_flag is not present in the encoded bitstream. In this case, decoder 403 may set the flag collocated_from_10_flag to a default value, such as 1, or may not use the flag collocated_from_10_flag.

At 1010, decoder 403 determines the value of the flag collocated_from_10_flag and the number of pictures in listO or listl. At 1012, decoder 403 evaluates a second condition "if( slice_type != I && enable_temporal_mvp_flag && (collocated_from_10_flag && num_ref_idx_10_active_minusl > 0) || (!collocated_from_10_flag && num_ref_idx_11_active_minusl > 0)". If the second condition is true, then at 1014 decoder 403 decodes the syntax collocated_ref_idx in the encoded bitstream. If the second condition is not true, then at 1016 decoder 403 determines that the syntax collocated_ref_idx was not encoded in the encoded bitstream. In this case, decoder 403 may assume the index is the first position (e.g., the position is 0).

At 1018, decoder 403 decodes the current block using the flag collocated_from_10_flag and the syntax collocated_ref_idx, if applicable.

Accordingly, particular embodiments save overhead by not sending information for the flag collocated_from_10_flag and the syntax collocated_ref_idx if certain conditions apply. For example, if a temporal MVP is not going to be used, then the flag collocated_from_10_flag and the syntax collocated_ref_idx do not need to be used.
Encoder and Decoder Examples

[0071] In various embodiments, encoder 402 described can be incorporated or otherwise associated with a transcoder or an encoding apparatus at a headend and decoder 403 can be incorporated or otherwise associated with a downstream device, such as a mobile device, a set top box or a transcoder. FIG. 11A depicts an example of encoder 402 according to one embodiment. A general operation of encoder 402 will now be described; however, it will be understood that variations on the encoding process described will be appreciated by a person skilled in the art based on the disclosure and teachings herein.

[0072] For a current PU, x, a prediction PU, x', is obtained through either spatial prediction or temporal prediction. The prediction PU is then subtracted from the current PU, resulting in a residual PU, e. Spatial prediction relates to intra mode pictures. Intra mode coding can use data from the current input image, without referring to other images, to code an I picture. A spatial prediction block 1104 may include different spatial prediction directions per PU, such as horizontal, vertical, 45-degree diagonal, 135-degree diagonal, DC (flat averaging), and planar, or any other direction. The spatial prediction direction for the PU can be coded as a syntax element. In some embodiments, brightness information (Luma) and color information (Chroma) for the PU can be predicted separately. In one embodiment, the number of Luma intra prediction modes for all block sizes is 35. An additional mode can be used for the Chroma intra prediction mode. In some embodiments, the Chroma prediction mode can be called "IntraFromLuma."

[0073] Temporal prediction block 1106 performs temporal prediction. Inter mode coding can use data from the current input image and one or more reference images to code "P" pictures and/or "B" pictures. In some situations and/or embodiments, inter mode coding can result in higher compression than intra mode coding. In inter mode PUs 204 can be temporally predictive coded, such that each PU 204 of the CU 202 can have one or more motion vectors and one or more associated reference images. Temporal prediction can be performed through a motion estimation operation that searches for a best match prediction for the PU over the associated reference images.
The best match prediction can be described by the motion vectors and associated reference images. P pictures use data from the current input image and one or more reference images, and can have up to one motion vector. B pictures may use data from the current input image and one or more reference images, and can have up to two motion vectors. The motion vectors and reference pictures can be coded in the encoded bitstream. In some embodiments, the motion vectors can be syntax elements "MV," and the reference pictures can be syntax elements "refldx." In some embodiments, inter mode can allow both spatial and temporal predictive coding. The best match prediction is described by the motion vector (MV) and associated reference picture index (refldx). The motion vector and associated reference picture index are included in the coded bitstream.

[0074] Transform block 1107 performs a transform operation with the residual PU, e. A set of block transforms of different sizes can be performed on a CU, such that some PUs can be divided into smaller TUs and other PUs can have TUs the same size as the PU. Division of CUs and PUs into TUs 20 can be shown by a quadtree representation. Transform block 1107 outputs the residual PU in a transform domain, E.

[0075] A quantizer 1108 then quantizes the transform coefficients of the residual PU, E. Quantizer 1108 converts the transform coefficients into a finite number of possible values. In some embodiments, this is a lossy operation in which data lost by quantization may not be recoverable. After the transform coefficients have been quantized, entropy coding block 1110 entropy encodes the quantized coefficients, which results in final compression bits to be transmitted. Different entropy coding methods may be used, such as context-adaptive variable length coding (CAVLC) or context-adaptive binary arithmetic coding (CABAC).

[0076] Also, in a decoding process within encoder 402, a de-quantizer 1112 de-quantizes the quantized transform coefficients of the residual PU. De-quantizer 1112 then outputs the de-quantized transform coefficients of the residual PU, E'. An inverse transform block 1114 receives the de-quantized transform coefficients, which are then inverse transformed resulting in a reconstructed residual PU, e'.
reconstructed PU, e', is then added to the corresponding prediction, x', either spatial or temporal, to form the new reconstructed PU, x". Particular embodiments may be used in determining the prediction, such as collocated reference picture manager 404 is used in the prediction process to determine the collocated reference picture to use.

A loop filter 1116 performs de-blocking on the reconstructed PU, x". to reduce blocking artifacts. Additionally, loop filter 1116 may perform a sample adaptive offset process after the completion of the de-blocking filter process for the decoded picture, which compensates for a pixel value offset between reconstructed pixels and original pixels. Also, loop filter 1116 may perform adaptive loop filtering over the reconstructed PU, which minimizes coding distortion between the input and output pictures. Additionally, if the reconstructed pictures are reference pictures, the reference pictures are stored in a reference buffer 1118 for future temporal prediction. Intra mode coded images can be a possible point where decoding can begin without needing additional reconstructed images.

[0077] FIG. 11B depicts an example of decoder 403 according to one embodiment. A general operation of decoder 403 will now be described; however, it will be understood that variations on the decoding process described will be appreciated by a person skilled in the art based on the disclosure and teachings herein. Decoder 403 receives input bits from encoder 402 for encoded video content.

[0078] An entropy decoding block 1130 performs entropy decoding on the input bitstream to generate quantized transform coefficients of a residual PU. A de-quantizer 1132 de-quantizes the quantized transform coefficients of the residual PU. De-quantizer 1132 then outputs the de-quantized transform coefficients of the residual PU, E'. An inverse transform block 1134 receives the de-quantized transform coefficients, which are then inverse transformed resulting in a reconstructed residual PU, e'.

[0079] The reconstructed PU, e', is then added to the corresponding prediction, x', either spatial or temporal, to form the new reconstructed PU, x". A loop filter 1136 performs de-blocking on the reconstructed PU, x", to reduce blocking artifacts. Additionally, loop filter 1136 may perform a sample adaptive offset process after the
completion of the de-blocking filter process for the decoded picture, which compensates for a pixel value offset between reconstructed pixels and original pixels. Also, loop filter 1136 may perform adaptive loop filtering over the reconstructed PU, which minimizes coding distortion between the input and output pictures.

Additionally, if the reconstructed pictures are reference pictures, the reference pictures are stored in a reference buffer 1138 for future temporal prediction.

[0080] The prediction PU, x', is obtained through either spatial prediction or temporal prediction. A spatial prediction block 1140 may receive decoded spatial prediction directions per PU, such as horizontal, vertical, 45-degree diagonal, 135-degree diagonal, DC (flat averaging), and planar. The spatial prediction directions are used to determine the prediction PU, x'.

[0081] A temporal prediction block 1106 performs temporal prediction through a motion estimation operation. Particular embodiments may be used in determining the prediction, such as collocated reference picture manager 404 is used in the prediction process to determine the collocated reference picture to use. A decoded motion vector is used to determine the prediction PU, x'. Interpolation may be used in the motion estimation operation.

[0082] Particular embodiments may be implemented in a non-transitory computer-readable storage medium for use by or in connection with the instruction execution system, apparatus, system, or machine. The computer-readable storage medium contains instructions for controlling a computer system to perform a method described by particular embodiments. The instructions, when executed by one or more computer processors, may be operable to perform that which is described in particular embodiments.

[0083] As used in the description herein and throughout the claims that follow, "a", "an", and "the" includes plural references unless the context clearly dictates otherwise. Also, as used in the description herein and throughout the claims that follow, the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise.
The above description illustrates various embodiments along with examples of how aspects of particular embodiments may be implemented. The above examples and embodiments should not be deemed to be the only embodiments, and are presented to illustrate the flexibility and advantages of particular embodiments as defined by the following claims. Based on the above disclosure and the following claims, other arrangements, embodiments, implementations and equivalents may be employed without departing from the scope hereof as defined by the claims.
CLAIMS

What is claimed is:

1. A method comprising:
   determining, by a computing device, whether a flag that controls whether a temporal motion vector is used for a slice is present in an encoded bitstream;
   if the flag is present, determining, by the computing device, a value for the flag;
   if the flag is not present, inferring, by the computing device, that the value for the flag is a default value; and
   evaluating, by the computing device, whether to determine a collocated reference picture in decoding the slice from the encoded bitstream based on the value of the flag.

2. The method of claim 1, wherein the default value enables the using the temporal motion vector for the slice.

3. The method of claim 2, wherein the default value is "1".

4. The method of claim 1, further comprising receiving the encoded bitstream from an encoder, wherein the encoder determined whether to encode a value for the flag in the encoded bitstream.

5. The method of claim 1, wherein when the value of the flag enables the use of the temporal motion vector, the method further comprises:
   determining the collocated reference picture from a list of reference pictures.
6. The method of claim 1, wherein the flag comprises a first flag, the method further comprising:
   determining, by the computing device, the value for the first flag;
   if the value of the first flag is a first value, performing:
      evaluating, by the computing device, a first condition to determine whether to decode a second flag that indicates whether to use a first list containing a first list of reference pictures or a second list containing a second list of reference pictures for determining a collocated reference picture;
      if the first condition indicates the second flag should be decoded, decoding, by the computing device, the second flag in decoding the one or more blocks;
      evaluating, by the computing device, a second condition to determine whether to decode position information that indicates a position of the collocated reference picture in one of the first list or the second list; and
      if the second condition indicates the position information should be decoded, decoding, by the computing device, the position information in decoding the slice; and
      if the first flag is a second value, not decoding, by the computing device, the second flag and the position information in decoding the slice.

7. The method of claim 6, wherein evaluating the first condition comprises:
   determining a slice type associated with the slice; and
   if the slice type is equal to a B type, then decoding the second flag in decoding the slice.

8. The method of claim 7, further comprising if the slice type is not equal to the B type, then not decoding the second flag in decoding the slice.
9. The method of claim 6, wherein evaluating the second condition comprises:
   determining a slice type associated with the slice;
   if the slice type is not equal to an I type, performing:
      evaluating a third condition to determine whether to decode the position information; and
      if the third condition indicates the position information should be decoded, then decoding the position information in decoding the slice.

10. The method of claim 1, wherein:
    a first value indicates that the temporal motion vector should be used in a temporal prediction process in encoding or decoding the slice, and
    a second value indicates that the temporal motion vector should not be used in the temporal prediction process in decoding the slice.

11. The method of claim 1, wherein the slice includes at least one block.

12. A decoder comprising:
    one or more computer processors; and
    a non-transitory computer-readable storage medium comprising instructions that, when executed, control the one or more computer processors to be configured for:
      determining whether a flag that controls whether a temporal motion vector is used for a slice is present in an encoded bitstream;
      if the flag is present, determining a value for the flag;
      if the flag is not present, inferring that the value for the flag is a default value; and
      evaluating whether to determine a collocated reference picture in decoding the slice from the encoded bitstream based on the value of the flag.
13. The decoder of claim 12, wherein the default value enables the using the temporal motion vector for the slice.

14. A method comprising:
   determining, by a computing device, whether a temporal motion vector should be used for encoding slice;
   determining, by the computing device, whether a flag that controls whether a temporal motion vector should be used for encoding the slice;
   if the flag should be used, encoding, by the computing device, a value for the flag in an encoded bitstream for the slice;
   if the flag should not be used, not encoding, by the computing device, the value for the flag in the encoded bitstream; and
   sending, by the computing device, the encoded bitstream to a decoder, wherein the decoder infers that the value of the flag is a default value when the value for the flag is not present in the encoded bitstream.

15. The method of claim 14, wherein the default value enables the using the temporal motion vector for the slice.

16. The method of claim 15, wherein the default value is "1".

17. The method of claim 14, wherein when the value of the flag is first value that enables the use of the temporal motion vector, the method further comprising:
   determining the collocated reference picture from a list of reference pictures.

18. The method of claim 14, wherein the flag comprises a first flag, the method further comprising:
   determining, by the computing device, the value for the first flag;
if the value of the first flag is a first value, performing:

   evaluating, by the computing device, a first condition to determine
whether to decode a second flag that indicates whether to use a first list
containing a first list of reference pictures or a second list containing a second
list of reference pictures for determining a collocated reference picture;

   if the first condition indicates the second flag should be encoded,
encoding, by the computing device, the second flag in encoding the one or
more blocks;

   evaluating, by the computing device, a second condition to determine
whether to encode position information that indicates a position of the
collocated reference picture in one of the first list or the second list; and

   if the second condition indicates the position information should be
encoded, encoding, by the computing device, the position information in
encoding the slice; and

   if the first flag is a second value, not encoding, by the computing device, the
second flag and the position information in encoding the slice.

19. An encoder comprising:

one or more computer processors; and

a non-transitory computer-readable storage medium comprising instructions
that, when executed, control the one or more computer processors to be configured
for:

determining whether a temporal motion vector should be used for encoding
slice;

determining whether a flag that controls whether a temporal motion vector
should be used for encoding the slice;

   if the flag should be used, encoding a value for the flag in an encoded
bitstream for the slice;

   if the flag should not be used, not encoding the value for the flag in the
encoded bitstream; and
sending the encoded bitstream to a decoder, wherein the decoder infers that the value of the flag is a default value when the value for the flag is not present in the encoded bitstream.

20. The encoder of claim 19, wherein the default value enables the using the temporal motion vector for the slice.
FIG. 1 (PRIOR ART)

FIG. 2 (PRIOR ART)
FIG. 3 (PRIOR ART)

FIG. 4
FIG. 5A

FIG. 5B
**FIG. 6B**

1. **RECEIVE AN ENCODED BITSTREAM FROM ENCODER**
   - 652

2. **DETERMINE THE VALUE OF THE FLAG FROM THE ENCODED BITSTREAM**
   - 658
   - YES

3. **FLAG PRESENT IN BITSTREAM?**
   - 654
   - NO

4. **INFER THE VALUE FOR THE FLAG enable_temporal_mvp_flag TO BE EQUAL TO A DEFAULT VALUE**
   - 656

5. **DECODE THE ENCODED BITSTREAM BASED ON THE VALUE OF THE FLAG enable_temporal_mvp_flag**
   - 660
if(slice_type == B && enable_temporal_mvp_flag)
  
  collocated_from_I0_flag

if(slice_type != 1 && enable_temporal_mvp_flag &&
((collocated_from_I0_flag && num_ref_idx_I0_active_minus_1 > 0) ||
  (collocated_from_I0_flag && num_ref_idx_I1_active_minus_1 > 0))
  
  collocated_ref_idx

FIG. 7A

ENCODER

enable_temporal_mvp_flag=1

ENCODED BITSTREAM
(enable_temporal_mvp_flag, collocated_from_I0_flag, and collocated_ref_idx)

DECODER

FIG. 7B

ENCODER

enable_temporal_mvp_flag=Ø

ENCODED BITSTREAM
(enable_temporal_mvp_flag)

DECODER

FIG. 7C
800

802

DETERMINE A CURRENT BLOCK BEING ENCODED OR DECODED

804

DETERMINE A VALUE FOR THE FLAG enable_temporal_mvp_flag THAT CONTROLS WHETHER A TEMPORAL MOTION VECTOR IS USED FOR ENCODING OR DECODING THE CURRENT BLOCK

806

DETERMINE IF THE enable_temporal_mvp_flag IS A FIRST VALUE OR A SECOND VALUE

808

DETERMINE WHETHER TO USE THE FLAG collocated_from_l0_flag

810

DETERMINE IF THE SYNTAX collocated_ref_idx SHOULD BE USED

FIG. 8
FIG. 10

1000

DETERMINE THAT THE SYNTAX COLLOCATED ref_idx, COLLOCATED ref_idx, IF APPLICABLE
WIN NOT ENCODED IN THE BITSTREAM

1008

DETERMINE THAT THE FLAG COLLOCATED FROM 10, flag IS NOT PRESENT IN THE BITSTREAM

1002

DETERMINE A SLICE TYPE AND VALUE FOR THE FLAG enable_temporal mvp_flag

1004

Determine the value of the flag = 0, enable_temporal mvp

1006

DECODE THE FLAG COLLOCATED FROM 10, flag IN THE ENCODED BITSTREAM

1010

Determine the value of the flag AND THE NUMBER OF PICTURES IN LIST0 OR LIST1

1012

slice_type = 1, enable_temporal mvp, flag

1014

TRUE

DECODE THE SYNTAX COLLOCATED ref_idx, COLLOCATED ref_idx, IN THE ENCODED BITSTREAM

1016

FALSE

DETERMINE THAT THE SYNTAX COLLOCATED ref_idx, COLLOCATED ref_idx, IF APPLICABLE
WIN NOT ENCODED IN THE BITSTREAM

1018

flag AND (collocated_from_10_flag && num_ref_idx_10_active_minus_1 > 0)
### A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N7/36 H04N7/26

ADD.

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)

EPO-Internal

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

"A" 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) on which the invention may not be considered to be obvious under the principle of priority
- "O" document referring to an oral disclosure, use, exhibition or other means

"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 prior art because of the novelty which it offers or because of its inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered prior art because of the inventive step when the document is taken in combination with one or more other such documents, such combination, being obvious to a person skilled in the art

"S" document member of the same patent family

Date of the actual completion of the international search: 2 May 2013

Date of mailing of the international search report: 14/05/2013

Name and mailing address of the ISA:

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016

Authorized officer: Oelbaum, Tobias
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>