



- (51) International Patent Classification:
H04N 19/52 (2014.01) *H04N 19/467* (2014.01)
- (21) International Application Number:
PCT/EP2017/061531
- (22) International Filing Date:
12 May 2017 (12.05.2017)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
62/335998 13 May 2016 (13.05.2016) US
- (71) Applicant: TELEFONAKTIEBOLAGET LM
ERICSSON (PUBL) [SE/SE]; SE-164 83 Stockholm (SE).
- (72) Inventors: WENNERSTEN, Per; Skedviksvägen 48,
SE-120 54 Årsta (SE). YU, Ruoyang; Carl Thunbergs Väg
7, SE-169 69 Solna (SE).
- (74) Agent: ERICSSON; Torshamnsgatan 21-23, 164 80
STOCKHOLM (SE).
- (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, DJ, 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, KH, KN, KP, KR,
KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
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.

(54) Title: MOTION VECTOR DIFFERENCE CODING AND DECODING

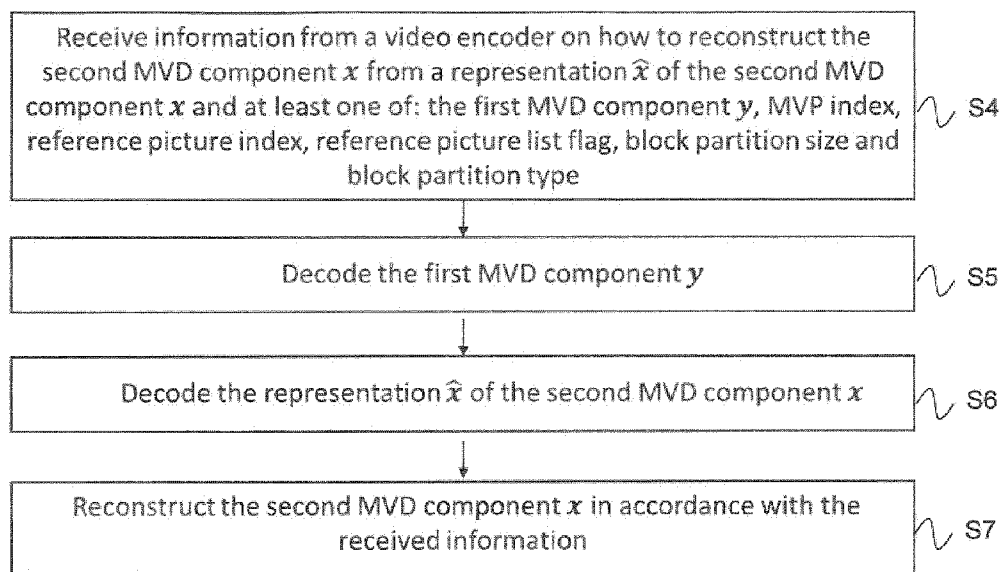


Figure 5

(57) Abstract: There are provided mechanisms for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The method comprises encoding the first MVD component y . The method comprises encoding a representation $\hat{x}$ of the second MVD component x . The method comprises sending information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index MVPindex, reference picture index Refldx, reference picture list flag, block partition size PartSize and block partition type. There are provided mechanisms for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD). The MVD comprises a first MVD component y and a second MVD component x . The method

(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, 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).

Published:

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

comprises receiving information from a video encoder on how to reconstruct the second MVD component x from a representation x of the second MVD component x and at least one of: the first MVD component y, MVP index MVPindex, reference picture index Refldx, reference picture list flag, block partition size PartSize and block partition type. The method comprises decoding the first MVD component y. The method comprises decoding the representation x of the second MVD component x. The method comprises reconstructing the second MVD component x in accordance with the received information.

MOTION VECTOR DIFFERENCE CODING AND DECODING

TECHNICAL FIELD

Embodiments herein relate to the field of video coding, such as High Efficiency Video Coding (HEVC) or the like. In particular, embodiments herein relate to a method and a video encoder for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP. Embodiments herein relate to a method and a video decoder for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD). Corresponding computer programs therefor are also disclosed.

BACKGROUND

High Efficiency Video Coding (HEVC), also known as H.265, is a block based hybrid video codec, standardized by the Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) and the Moving Pictures Experts Group (MPEG), that utilizes both temporal and spatial prediction.

Pictures in a video sequences are divided into small square units as bases for encoding. In HEVC, these units are referred to as coding unit (CU) which can range from size of 8x8 to 64x64. Each unit can be further partitioned into smaller blocks for spatial or temporal prediction. Up to 4 symmetric and 4 asymmetric partition structures are supported for better capturing characteristics for different video content.

Spatial prediction, which is also called intra prediction, is achieved by predicting current block using previous decoded blocks within the same picture. A picture consisting of only intra-predicted blocks is referred as an I-picture. As I-pictures do not have any dependency on any other pictures, they provide starting points for encoding or decoding. Therefore, the first picture inside a video sequence is typically encoded as an I-picture.

Temporal prediction, also known as inter prediction, is achieved by predicting blocks in a current picture using blocks from previous decoded pictures, the so-called reference pictures, along with motion vectors indicating the movement between the pictures. Temporal prediction is achieved using inter (P) prediction, i.e., prediction from one block in one reference picture, or bi-directional inter (B) prediction, i.e. prediction from blending of two blocks in one or two reference pictures. A picture containing only P-predicted blocks is referred as a P-picture. A picture that contains at least one bi-predicted block is referred as a B-picture.

When encoding a picture, it is possible to have several reference pictures. These reference pictures are grouped into two lists. The reference pictures that are displayed before the current picture are usually grouped into list 0 while the reference pictures that are displayed after the current picture are usually grouped into list 1. For signaling the usage of a specific reference picture for current block, a reference picture list flag and a reference picture index are needed to be encoded. The reference picture list flag specifies which reference picture list to use, while the reference picture index specifies the targeted reference picture's index inside the list.

The position of a referenced block is indicated using motion vector (MV). Each MV consists of x and y components that signal the displacements between current block and the referenced block. The MV may point to full-pel (full-pixel), half-pel (half-pixel) and quarter-pel (quarter-pixel) samples. Half-pel and quarter-pel samples are generated by interpolating their corresponding neighboring full-pel samples. For an encoded video bitstream, MVs can constitute up to about 50% of the total bitrate.

Since a moving object in a picture may span across several blocks, the MV of the current block is usually highly correlated with neighboring blocks' MVs. Thus, one way of improving the compression efficiency is to predict current MV from neighboring MVs. A common motion vector prediction process begins with constructing a list of neighboring MVs which is referred to as MVP list, selects one of them as motion vector predictor (MVP) and then calculates the motion vector difference (MVD) between the predictor and current MV. Therefore, instead of encoding a "full" MV, only the MVP index which tells the index of the selected MVP inside the MVP list and the corresponding MVD are encoded. Figure 1 shows an example of motion vector prediction. The current MV (-2,4) is to be predicted from a list of two MVP candidates: a MV with index 0 and having a value (24,9) and a MV with index 1 and having a value (-4,7). As the current MV is closer to the MV with index 1, the MV with index 1 is chosen as the MVP candidate. What is being encoded is the MVP index 1 and the MVD (2, -3), i.e., the difference between the current MV (-2, 4) and the MVP candidate (-4,7). As shown in the figure, the MVD has x and y components. Each of them consists of two parts: magnitude part and sign part. These two parts are usually encoded separately.

SUMMARY

The problem with motion vector prediction is that there still exists redundancy in motion vector difference encoding which leaves room for further improving compression efficiency. In order to reduce a bitrate or, equivalently, increase coding efficiency, a method and arrangements for encoding and decoding of motion vectors are provided.

A first aspect of the embodiments defines a method performed by a video encoder, for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP)

and a motion vector difference (MVD) between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The method comprises encoding the first MVD component y . The method comprises encoding a representation $\hat{x}$ of the second MVD component x . The method comprises sending information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

A second aspect of the embodiments defines a video encoder, for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The video encoder comprises processing means operative to encode the first MVD component y . The video encoder comprises processing means operative to encode a representation $\hat{x}$ of the second MVD component x . The video encoder comprises processing means operative to send information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

A third aspect of the embodiments defines a computer program, for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The computer program comprises code means which, when run on a computer, causes the computer to encode the first MVD component y . The computer program comprises code means which, when run on a computer, causes the computer to encode a representation $\hat{x}$ of the second MVD component x . The computer program comprises code means which, when run on a computer, causes the computer to send information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

A fourth aspect of the embodiments defines a computer program product comprising computer readable means and a computer program according to the third aspect, stored on the computer readable means.

A fifth aspect of the embodiments defines a method, performed by a video decoder, for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD). The MVD

comprises a first MVD component y and a second MVD component x . The method comprises receiving information from a video encoder on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. The method comprises decoding the first MVD component y . The method comprises decoding the representation $\hat{x}$ of the second MVD component x . The method comprises reconstructing the second MVD component x in accordance with the received information.

A sixth aspect of the embodiments defines a video decoder, for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD). The MVD comprises a first MVD component y and a second MVD component x . The video decoder comprises processing means operative to receive information from a video encoder on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. The video decoder comprises processing means operative to decode the first MVD component y . The video decoder comprises processing means operative to decode the representation $\hat{x}$ of the second MVD component x . The video decoder comprises processing means operative to reconstruct the second MVD component x in accordance with the received information.

A seventh aspect of the embodiments defines a computer program, for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD). The MVD comprises a first MVD component y and a second MVD component x . The computer program comprises code means which, when run on a computer, causes the computer to receive information from a video encoder on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. The computer program comprises code means which, when run on a computer, causes the computer to decode the first MVD component y . The computer program comprises code means which, when run on a computer, causes the computer to decode the representation $\hat{x}$ of the second MVD component x . The computer program comprises code means which, when run on a computer, causes the computer to reconstruct the second MVD component x in accordance with the received information.

An eighth aspect of the embodiments defines a computer program product comprising computer readable means and a computer program according to the seventh aspect, stored on the computer readable means.

Advantageously, at least some of the embodiments provide a reduced bitrate, without introducing noticeable decoding complexity.

It is to be noted that any feature of the first, second, third, fourth, fifth, sixth, seventh and eighth aspects may be applied to any other aspect, whenever appropriate. Likewise, any advantage of the first aspect may equally apply to the second, third, fourth, fifth, sixth, seventh and eighth aspect respectively, and vice versa. Other objectives, features and advantages of the enclosed embodiments will be apparent from the following detailed disclosure, from the attached dependent claims and from the drawings.

Generally, all terms used in the claims are to be interpreted according to their ordinary meaning in the technical field, unless explicitly defined otherwise herein. All references to "a/an/the element, apparatus, component, means, step, etc." are to be interpreted openly as referring to at least one instance of the element, apparatus, component, means, step, etc., unless explicitly stated otherwise. The steps of any method disclosed herein do not have to be performed in the exact order disclosed, unless explicitly stated.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, together with further objects and advantages thereof, may best be understood by referring to the following description taken together with the accompanying drawings, in which:

Figure 1 illustrates an example of motion vector prediction according to the prior art.

Figure 2 illustrates the steps performed in an encoding method according to the embodiments of the present invention.

Figures 3 (A) and (B) illustrate examples of motion vector search patterns according to an embodiment of the present invention.

Figure 4 illustrates a motion vector search pattern according to an embodiment of the present invention.

Figure 5 illustrates the steps performed in a decoding method according to the embodiments of the present invention.

Figure 6 depicts a schematic block diagram illustrating functional units of a video encoder for encoding a motion vector according to embodiments of the present invention.

Figure 7 illustrates a schematic block diagram illustrating a computer comprising a computer program product with a computer program for encoding a motion vector, according to embodiments of the present invention.

Figure 8 depicts a schematic block diagram illustrating functional units of a video decoder for reconstructing a motion vector, according to an embodiment of the present invention.

Figure 9 illustrates a schematic block diagram illustrating a computer comprising a computer program product with a computer program for reconstructing a motion vector, according to an embodiment of the present invention.

DETAILED DESCRIPTION

The accompanying drawings, which are incorporated herein and form part of the specification, illustrate various embodiments of the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the art to make and use the invention.

Even though the description of the invention is based on the HEVC codec, it is to be understood by a person skilled in the art that the invention could be applied to any other state-of-the-art and a future block-based video coding standard.

The present embodiments generally relate to a method and an encoder for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction and a motion vector difference between the motion vector and the MVP, as well as a method and a decoder for reconstructing a motion vector from a motion vector prediction and a motion vector difference.

According to one aspect, a method, performed by a video encoder, for encoding a motion vector is provided, as described in Figure 2. The motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The method comprises a step S1 of encoding the first MVD component y . The first MVD component y may, for example, be encoded as is currently done in HEVC.

The method comprises a step S2 of encoding a representation $\hat{x}$ of the second MVD component x . The representation $\hat{x}$ may be a reduced size version of the second MVD component x , i.e., may carry a part of the second MVD component x . For example, it may be an incomplete version of x , obtained by removing at least one bit of the second MVD component x . Typically, the representation may contain all but the least significant bit of the second MVD component, in which case the parity of the second MVD component is removed, or all but the most significant bit of the second MVD component, in which case the sign of the MVD component is removed and the representation corresponds to the magnitude of the second MVD component. More details about the representation $\hat{x}$ will be given below.

The method comprises a step S3 of sending information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. This in principle means that the representation $\hat{x}$ is created in such a way that the second MVD component x can be reconstructed from $\hat{x}$ and at least one of: the first MVD component y and syntax elements such as MVP index, reference picture index, reference picture list flag, block partition size, block partition type etc. Sending this information (i.e., signaling information) ensures identical reconstruction at the video decoder. The video decoder is this way informed about how the reconstruction of the second MVD component and the MVD accordingly needs to be performed, i.e., which parts of the MVD components and syntax elements are used and what operation needs to be performed on them.

The semantics of each aforementioned syntax elements are as follows: MVP index specifies the MVP candidate index in a MVP list; reference picture list flag specifies which reference picture list of usage, i.e. list 0 or list 1; reference picture index specifies the index of the targeted reference picture inside the reference picture list; block partition size specifies the partitioning size and block partition type specifies the partitioning structure. As can be seen, these elements contain various prediction information for an inter-coded block. Therefore, it gives the video encoder additional freedom of deciding a best way to represent $\hat{x}$ by selectively using these syntax elements. As will be further seen below, the identical reconstruction needs to be performed at both the video encoder and the video decoder in order to correctly reconstruct the second MVD component x .

Sending (signaling) information may be done, but is not limited to, by using a sequence parameter set (SPS), picture parameter set (PPS) or video parameter set (VPS). The information may also be sent "offline", i.e., the video encoder and the video decoder may exchange in advance the details of the reconstruction process.

According to the first embodiment of the present invention, the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x . The sign of the second MVD component x is reconstructed as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, and AND is a logical AND operation. In this expression, $\text{Magnitude}(x)$ and $\text{Magnitude}(y)$ respectively refer to the magnitudes of the second and first MVD components, and PartSize and RefIdx refer to a block partition size and a reference picture index, as described above. 'AND 0x01' in the expression above means that only the last bit of the

operation (Magnitude(x) XOR Magnitude(y) XOR PartSize XOR RefIdx) is taken as the Sign(x). Sign(x) may also be referred to as the most significant bit of x , or MSB(x).

According to this embodiment, the sign of the second MVD component x is not encoded, i.e., it is not sent to a video decoder. Instead, Sign(x) is reconstructed at the decoder from the sent information that tells to use the expression above. Not encoding a bit of the second MVD component, while knowing how to reconstruct it, results in bitrate savings. This also implies that the reconstruction process of the second MVD component, according to this embodiment, needs to be identical at the video encoder and the video decoder side, as mentioned above.

According to the second embodiment of the present invention, the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , LSB(x). The least significant bit of the second MVD component x is, according to this embodiment, embedded (contained) in the parity information (i.e. the least significant bit) of the first MVD component y , LSB(y), and the MVP index:

$$\text{LSB}(x) = (\text{LSB}(y) \text{ XOR MVPIndex}) \text{ AND } 0x01.$$

In this case, the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR MVPIndex}) \text{ AND } 0x01).$$

Similar as above, 'AND 0x01' in the expression above means that only the last bit of the operation LSB(y) XOR MVPIndex is taken as LSB(x). The same operation is performed at the video encoder and the video decoder, so that the least significant bit of the second MVD component x does not need to be sent.

This embodiment may also be generalized such that a bit other than the LSB is embedded in a corresponding bit from the first MVD component.

According to the third embodiment of the present invention, the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , LSB(x).

However, in this case, the least significant bit of the second MVD component is obtained by combining multiple syntax elements:

$$\text{LSB}(x) = (\text{LSB}(y) \text{ XOR MVPIndex XOR RefIdx XOR PartSize}) \text{ AND } 0x01.$$

In this case the second MVD component x is reconstructed as:

$$x = 2\hat{x} + (\text{LSB}(y) \text{ XOR MVPIndex XOR RefIdx XOR PartSize}) \text{ AND } 0x01.$$

Similar as in the second embodiment above, this embodiment may also be generalized such that a bit other than the LSB is embedded in a corresponding bit from the first MVD component.

According to the fourth embodiment of the present invention, the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, which is instead embedded in the parity information of the first MVD component y :

$$\text{LSB}(x) = \text{LSB}(y).$$

For example, if the MVD is (4, -8), i.e. $x=4$ and $y=-8$, then the second MVD component's least significant bit 0 can be derived using the parity of the first MVD component. Thus, only (2, -8) will be encoded.

According to this embodiment, the second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

Setting a condition $\text{LSB}(x) = \text{LSB}(y)$ may imply that, on the video encoder side, a motion vector search pattern as shown in Figure 3(A) may be applied when searching for the best motion vector around a certain MVP. The circles correspond to allowed MV positions, i.e. the positions for which $\text{LSB}(x) = \text{LSB}(y)$, while crosses indicate disallowed positions, i.e. $\text{LSB}(x) \neq \text{LSB}(y)$. By using this search pattern it is assured that the condition $\text{LSB}(x) = \text{LSB}(y)$, in which case the second MVD component x can always be reconstructed as above.

Another alternative is to embed the second MVD component's LSB using the inverse parity of the first MVD component y . In this case, a motion search pattern as shown in Figure 3(B) may be applied on the encoder side.

Similar as above, this embodiment may also be generalized to use other bit than the LSB to be embedded in a corresponding bit from another motion vector component.

According to another version of this embodiment, when reconstructing, instead of always multiplying the second MVD component's magnitude by two and adding the LSB of the first MVD component's magnitude, the procedure depends on the value of the second MVD component. For example, when this value is 0, no modification is made. When the value is smaller than 0, it is multiplied by 2 and the LSB of the first component's magnitude is added. When it is larger than 0, it is multiplied by 2, and the inverse of the LSB of the y component's magnitude is subtracted. In this case, a motion vector search pattern as shown in Figure 4 may be applied by the video encoder. The benefit of this is that the commonly used motion vectors (0,0), (0,1), (1,0), (0,-1) and (-1,0) can be all used and reconstructed, which leads to slightly better overall compression.

Similar as above, this embodiment may also be generalized to use a bit other than the LSB to be embedded in a corresponding bit from another MVD component.

According to another aspect, a method, performed by a video decoder, for reconstructing a motion vector from a motion vector prediction and a motion vector difference is provided, as illustrated in

5 Figure 5. The MVD comprises a first MVD component y and a second MVD component x . The method comprises a step S4 of receiving information from a video encoder on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. This ensures a correct reconstruction of the
10 second MVD component x . The received information may be signaled by using a sequence parameter set, picture parameter set or video parameter set, as mentioned above. The video encoder and the video decoder may as well exchange in advance the details of the reconstruction process.

The method comprises a step S5 of decoding the first MVD component y . The method further comprises a step S6 of decoding the representation $\hat{x}$ of the second MVD component x . Similar as described in
15 step S2 above, the representation $\hat{x}$ may be a reduced size version of the second MVD component x and may, for example, be obtained by removing at least one bit of the second MVD component x . Typically, the representation may contain all but the least significant bit of the second MVD component, in which case the parity of the second MVD component is removed, or all but the most significant bit of the second MVD component, in which case the sign of the MVD component is removed and the
20 representation corresponds to the magnitude of the second MVD component.

The method further comprises a step S7 of reconstructing the second MVD component x in accordance with the received information.

According to an embodiment, which is combined with the first encoder embodiment, the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and
25 a sign of the second MVD component x is calculated as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01.$$

The reconstruction process in this embodiment is identical to the one described in the first encoder embodiment. For more details, please refer to the related section above.

According to another embodiment, which is related to the second encoder embodiment above, the
30 representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$. The second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

According to yet another embodiment, which is related to the third encoder embodiment above, the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$. The second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

According to yet another embodiment, which is related to the fourth encoder embodiment above, the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$. The second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

Figure 6 is a schematic block diagram of a video encoder 100, for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction and a motion vector difference between the motion vector and the MVP. The MVD comprises a first MVD component y and a second MVD component x . The video encoder 100 comprises, according to this aspect, an encoding unit 160, configured to encode the first MVD component y and to encode a representation $\hat{x}$ of the second MVD component x . The video encoder 100 comprises, according to this aspect, a sending unit 170, configured to send information to a video decoder on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index MVPindex , reference picture index RefIdx , reference picture list flag, block partition size PartSize and block partition type.

The encoding 160 and sending 170 units may be hardware based, software based (in this case they are called encoding and sending modules respectively) or may be a combination of hardware and software.

The encoder 100 may be an HEVC encoder or any other state of the art or future video encoder.

The sending unit 170 may send information to the video decoder that the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and that the sign of the second MVD component x is reconstructed as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, and AND is a logical AND operation.

The sending unit 170 may send information to the video decoder that the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, and that the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

The sending unit 170 may send information to the video decoder that the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, and that the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

The sending unit 170 may send information to the video decoder that the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, and that the second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

The video encoder 100 can be implemented in hardware, in software or a combination of hardware and software. The video encoder 100 can be implemented in user equipment, such as a mobile telephone, tablet, desktop, netbook, multimedia player, video streaming server, set-top box or computer. The video encoder 100 may also be implemented in a network device in the form of or connected to a network node, such as radio base station, in a communication network or system.

Although the respective units disclosed in conjunction with Figure 6 have been disclosed as physically separate units in the device, where all may be special purpose circuits, such as ASICs (Application Specific Integrated Circuits), alternative embodiments of the device are possible where some or all of the units are implemented as computer program modules running on a general purpose processor. Such an embodiment is disclosed in Figure 7.

Figure 7 schematically illustrates an embodiment of a computer 150 having a processing unit 110 such as a DSP (Digital Signal Processor) or CPU (Central Processing Unit). The processing unit 110 can be a single unit or a plurality of units for performing different steps of the method described herein. The computer also comprises an input/output (I/O) unit 120 for receiving a video sequence. The I/O unit 120 has been illustrated as a single unit in Figure 6 but can likewise be in the form of a separate input unit and a separate output unit.

Furthermore, the computer 150 comprises at least one computer program product 130 in the form of a non-volatile memory, for instance an EEPROM (Electrically Erasable Programmable Read-Only Memory), a flash memory or a disk drive. The computer program product 130 comprises a computer program 140, which comprises code means which, when run on the computer 150, such as by the processing unit 110, causes the computer 150 to perform the steps of the method described in the foregoing in connection with Figure 2.

Figure 8 is a schematic block diagram of a video decoder 200 for reconstructing a motion vector from a motion vector prediction and a motion vector difference, wherein the motion vector difference comprises a first MVD component y and a second MVD component x . The video decoder 200 comprises, according to this aspect, a receiving unit 260, configured to receive information from a video encoder 100 on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type. The video decoder 200 comprises, according to this aspect, a decoding unit 270, configured to decode the first MVD component y and to decode the representation $\hat{x}$ of the second MVD component x . The video decoder comprises, according to this aspect, a reconstructing unit 280, configured to reconstruct the second MVD component x in accordance with the received information.

The receiving 260, decoding 270 and reconstructing 280 units may be hardware based, software based (in this case they are called receiving, decoding and reconstructing modules respectively) or may be a combination of hardware and software.

The video decoder 200 may be an HEVC decoder or any other state of the art or future video decoder.

The reconstructing unit 280 may, when the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , reconstruct the sign of the second MVD component x as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, and AND is a logical AND operation.

The reconstructing unit 280 may, when the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, reconstruct the second MVD component x as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

The reconstructing unit 270 may, when the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, reconstruct the second MVD component x as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

- 5 The reconstructing unit 270 may, when the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, reconstruct the second MVD component x as:

$$x = 2\hat{x} + \text{LSB}(y).$$

- 10 The video decoder 200 can be implemented in hardware, in software or a combination of hardware and software. The video decoder 200 can be implemented in user equipment, such as a mobile telephone, tablet, desktop, netbook, multimedia player, video streaming server, set-top box or computer. The video decoder 200 may also be implemented in a network device in the form of or connected to a network node, such as radio base station, in a communication network or system.
- 15 Although the respective units disclosed in conjunction with Figure 8 have been disclosed as physically separate units in the device, where all may be special purpose circuits, such as ASICs (Application Specific Integrated Circuits), alternative embodiments of the device are possible where some or all of the units are implemented as computer program modules running on a general purpose processor. Such an embodiment is disclosed in Figure 9.
- 20 Figure 9 schematically illustrates an embodiment of a computer 250 having a processing unit 210 such as a DSP (Digital Signal Processor) or CPU (Central Processing Unit). The processing unit 210 can be a single unit or a plurality of units for performing different steps of the method described herein. The computer also comprises an input/output (I/O) unit 220 for receiving a video bitstream. The I/O unit 220 has been illustrated as a single unit in Figure 9 but can likewise be in the form of a separate input unit and
- 25 a separate output unit.

Furthermore, the computer 250 comprises at least one computer program product 230 in the form of a non-volatile memory, for instance an EEPROM (Electrically Erasable Programmable Read-Only Memory), a flash memory or a disk drive. The computer program product 230 comprises a computer program 240, which comprises code means which, when run on the computer 250, such as by the processing unit 210,

30 causes the computer 250 to perform the steps of the method described in the foregoing in connection with Figure 5.

The embodiments described above are to be understood as a few illustrative examples of the present invention. It will be understood by those skilled in the art that various modifications, combinations and

changes may be made to the embodiments without departing from the scope of the present invention. In particular, different part solutions in the different embodiments can be combined in other configurations, where technically possible.

5

10

15

20

25

CLAIMS

1. A method, performed by a video encoder (100), for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP, wherein the MVD comprises a first MVD component y and a second MVD component x , the method comprising:

5 encoding (S1) the first MVD component y ;

encoding (S2) a representation $\hat{x}$ of the second MVD component x ;

10 sending (S3) information to a video decoder (200) on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

2. The method according to claim 1, wherein the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and wherein the sign of the second MVD component x is reconstructed as:

15 $Sign(x) = (Magnitude(x) \text{ XOR } Magnitude(y) \text{ XOR } PartSize \text{ XOR } RefIdx) \text{ AND } 0x01$,
wherein XOR is an exclusive OR logical operation, and AND is a logical AND operation.

3. The method according to claim 1, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $LSB(x)$, and wherein the second MVD component x is reconstructed as:

20 $x = 2\hat{x} + ((LSB(y) \text{ XOR } MVPindex) \text{ AND } 0x01)$.

4. The method according to claim 1, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $LSB(x)$, wherein the second MVD component x is reconstructed as:

$x = 2\hat{x} + ((LSB(y) \text{ XOR } MVPindex \text{ XOR } RefIdx \text{ XOR } PartSize) \text{ AND } 0x01)$.

25 5. The method according to claim 1, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $LSB(x)$, wherein the second MVD component x is reconstructed as:

$x = 2\hat{x} + LSB(y)$.

6. The method according to any of the preceding claims, wherein the video encoder (100) is an HEVC video encoder.

7. A method, performed by a video decoder (200), for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD), wherein the MVD comprises a first MVD component y and a second MVD component x , the method comprising:

receiving (S4) information from a video encoder (100) on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type;

decoding (S5) the first MVD component y ;

decoding (S6) the representation $\hat{x}$ of the second MVD component x ;

reconstructing (S7) the second MVD component x in accordance with the received information.

8. The method according to claim 7, wherein the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and wherein a sign of the second MVD component x is calculated as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, AND is a logical AND operation.

9. The method according to claim 7, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

10. The method according to claim 7, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

11. The method according to claim 7, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

12. The method according to any of claims 7-11, wherein the video decoder (200) is an HEVC video decoder.

13. A video encoder (100), for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP, wherein the MVD comprises a first MVD component y and a second MVD component x , the video encoder (100) comprising processing means (110) operative to:

encode the first MVD component y ;

encode a representation $\hat{x}$ of the second MVD component x ;

send information to a video decoder (200) on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

14. The video encoder (100) according to claim 13, wherein the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and wherein the sign of the second MVD component x is reconstructed as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, AND is a logical AND operation.

15. The video encoder (100) according to claim 13, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, and wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

16. The video encoder (100) according to claim 13, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

17. The video encoder (100) according to claim 13, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

18. The video encoder (100) according to any of claims 13-17, wherein the video encoder (100) is an HEVC video encoder.

19. A video decoder (200), for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD), wherein the MVD comprises a first MVD component y and a second MVD component x , the video decoder (200) comprising processing means (210) operative to:

receive information from a video encoder (100) on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type;

decode the first MVD component y ;

decode the representation $\hat{x}$ of the second MVD component x ;

reconstruct the second MVD component x in accordance with the received information.

20. The video decoder (200) according to claim 19, wherein the representation $\hat{x}$ of the second MVD component x corresponds to a magnitude of the second MVD component x , and wherein a sign of the second MVD component x is calculated as:

$$\text{Sign}(x) = (\text{Magnitude}(x) \text{ XOR } \text{Magnitude}(y) \text{ XOR } \text{PartSize} \text{ XOR } \text{RefIdx}) \text{ AND } 0x01,$$

wherein XOR is an exclusive OR logical operation, AND is a logical AND operation.

21. The video decoder (200) according to claim 19, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex}) \text{ AND } 0x01).$$

22. The video decoder (200) according to claim 19, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + ((\text{LSB}(y) \text{ XOR } \text{MVPindex} \text{ XOR } \text{RefIdx} \text{ XOR } \text{PartSize}) \text{ AND } 0x01).$$

23. The video decoder (200) according to claim 19, wherein the representation $\hat{x}$ of the second MVD component x comprises all but the least significant bit of the second MVD component x , $\text{LSB}(x)$, wherein the second MVD component x is reconstructed as:

$$x = 2\hat{x} + \text{LSB}(y).$$

24. The video decoder (200) according to any of claims 19-23, wherein the video decoder (200) is an HEVC video decoder.

25. A computer program (140), for encoding a motion vector, wherein the motion vector is represented by a sum of a motion vector prediction (MVP) and a motion vector difference (MVD) between the motion vector and the MVP, wherein the MVD comprises a first MVD component y and a second MVD component x , the computer program (140) comprising code means which, when run on a computer (150), causes the computer (150) to:

encode the first MVD component y ;

encode a representation $\hat{x}$ of the second MVD component x ;

send information to a video decoder (200) on how reconstruct the second MVD component x from the representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type.

26. A computer program (240), for reconstructing a motion vector from a motion vector prediction (MVP) and a motion vector difference (MVD), wherein the MVD comprises a first MVD component y and a second MVD component x , the computer program (240) comprising code means which, when run on a computer (250), causes the computer (250) to:

receive information from a video encoder (100) on how to reconstruct the second MVD component x from a representation $\hat{x}$ of the second MVD component x and at least one of: the first MVD component y , MVP index $MVPindex$, reference picture index $RefIdx$, reference picture list flag, block partition size $PartSize$ and block partition type;

decode the first MVD component y ;

decode the representation $\hat{x}$ of the second MVD component x ;

reconstruct the second MVD component x in accordance with the received information.

27. A computer program product (300) comprising computer readable means (310) and a computer program (140) according to claim 25 stored on the computer readable means (310).

28. A computer program product (400) comprising computer readable means (410) and a computer program (240) according to claim 26 stored on the computer readable means (410).

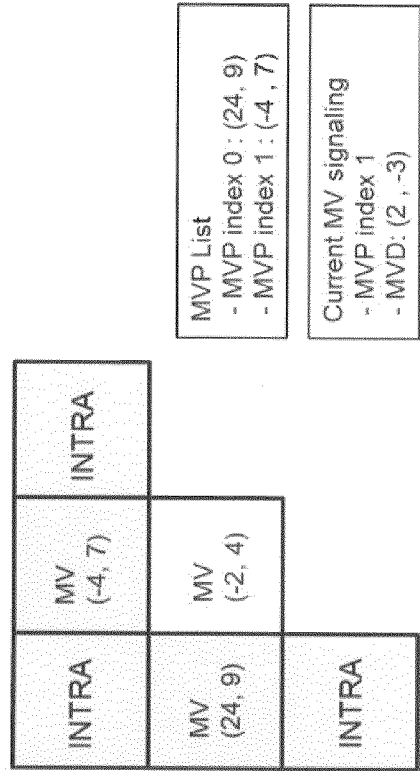


Figure 1

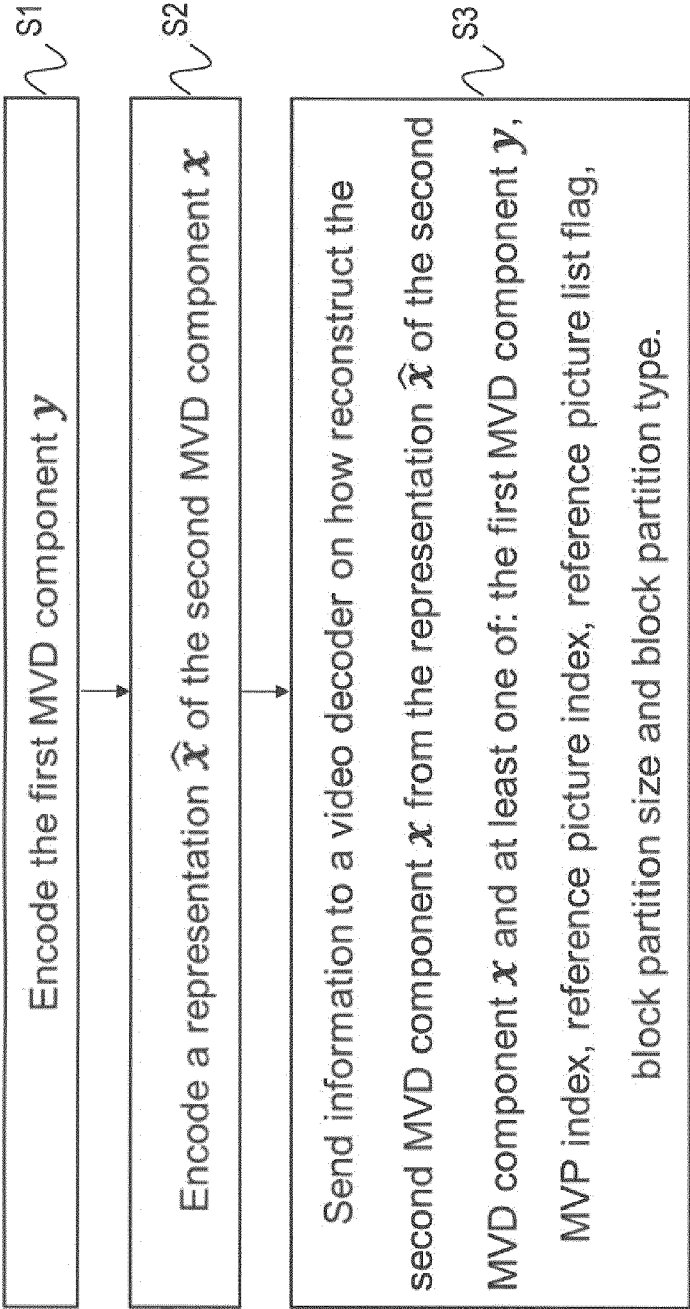


Figure 2

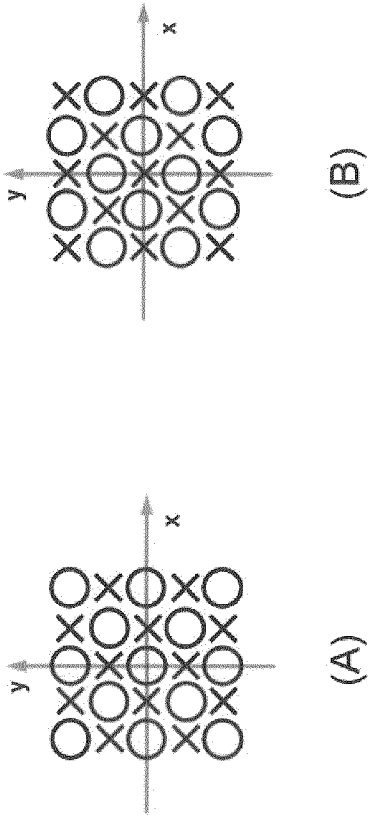


Figure 3

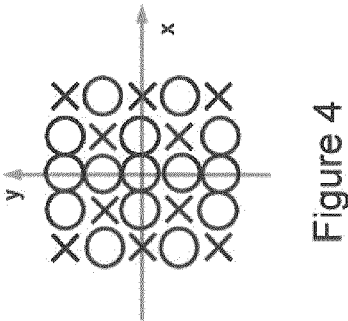


Figure 4

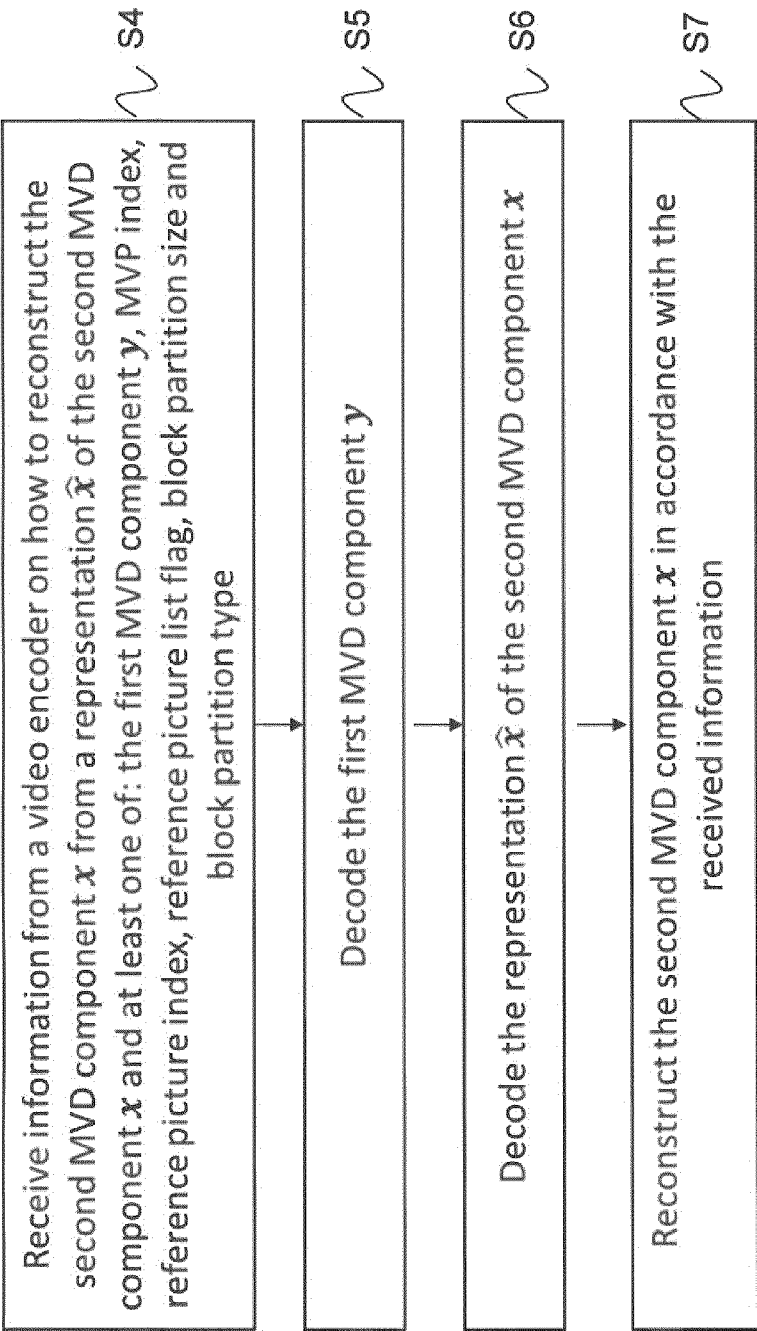


Figure 5

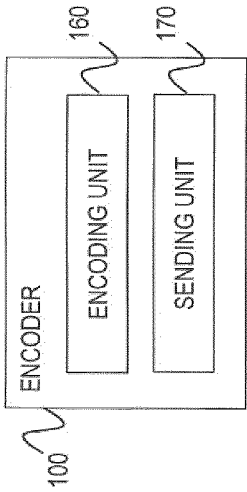


Figure 6

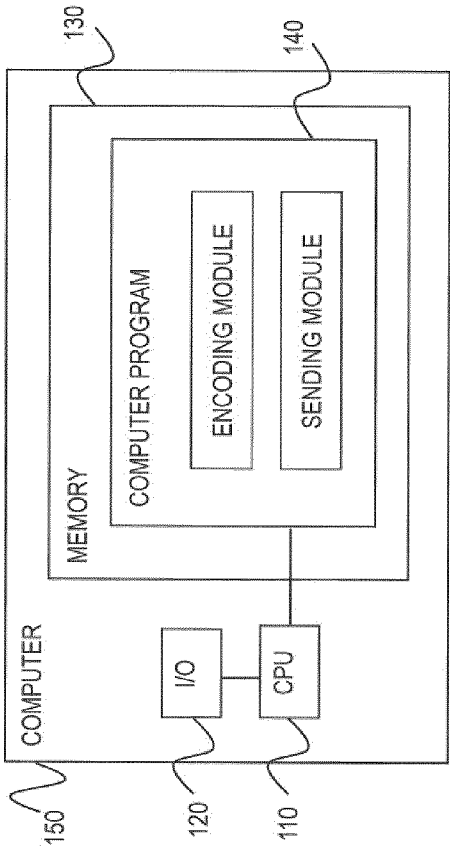


Figure 7

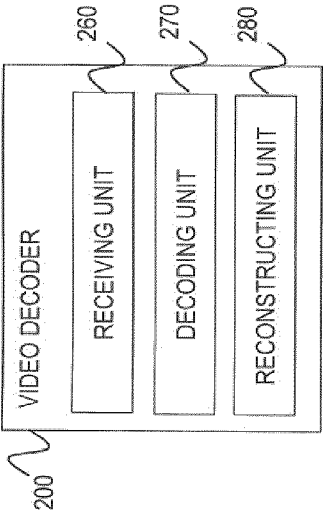


Figure 8

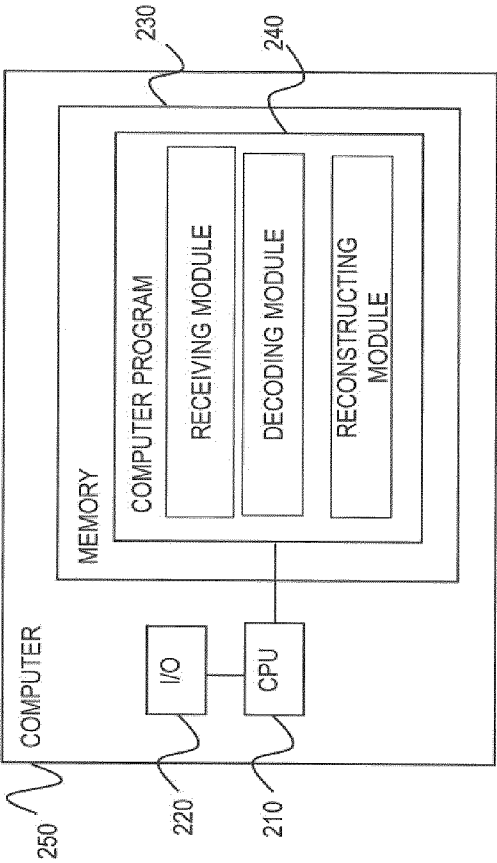


Figure 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/061531

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/52 H04N19/467
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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2 637 409 A1 (BLACKBERRY LTD [CA]) 11 September 2013 (2013-09-11)	1,6,7, 12,13, 18,19, 24-28
A	paragraphs [0023] - [0074] ----- -/--	2-5, 8-11, 14-17, 20-23



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"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

22 June 2017

Date of mailing of the international search report

10/07/2017

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

Sanahuja, Francesc

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2017/061531

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	SAMUELSSON J ET AL: "Motion vector coding optimizations", 3. JVET MEETING; 26-5-2016 - 1-6-2016; GENEVA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-C0068, 17 May 2016 (2016-05-17), XP030150171,	1,2,6-8, 12-14, 18-20, 24-28
A,P	the whole document	3-5, 9-11, 15-17, 21-23
A	----- CLARE (ORANGE LABS) G ET AL: "Sign Data Hiding", 98. MPEG MEETING; 28-11-2011 - 2-12-2011; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m21833, 24 November 2011 (2011-11-24), XP030050396, Sections 1-4.	1-28
A	----- EP 2 618 572 A1 (RESEARCH IN MOTION LTD [CA]) 24 July 2013 (2013-07-24) paragraphs [0064] - [0066]	1-28
A	----- US 2012/189055 A1 (CHIEN WEI-JUNG [US] ET AL) 26 July 2012 (2012-07-26) paragraphs [0043] - [0047] -----	1-28

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2017/061531

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2637409	A1	11-09-2013	CA 2807786 A1 08-09-2013
		EP 2637409 A1	11-09-2013
		US 2013235936 A1	12-09-2013

EP 2618572	A1	24-07-2013	AU 2013200325 B2 04-12-2014
		CA 2801767 A1	20-07-2013
		CN 103220515 A	24-07-2013
		CN 105959697 A	21-09-2016
		EP 2618572 A1	24-07-2013
		EP 3094094 A1	16-11-2016
		JP 5537681 B2	02-07-2014
		JP 2013150323 A	01-08-2013
		KR 20130085977 A	30-07-2013
		SG 192360 A1	30-08-2013
		TW 201340717 A	01-10-2013

US 2012189055	A1	26-07-2012	CN 103339935 A 02-10-2013
		EP 2666293 A1	27-11-2013
		JP 5778299 B2	16-09-2015
		JP 2014509116 A	10-04-2014
		KR 20130126688 A	20-11-2013
		US 2012189055 A1	26-07-2012
		WO 2012099692 A1	26-07-2012
