METHOD AND APPARATUS FOR INTRA PREDICTION

~~CROSS-REFERENCE TO RELATED APPLICATIONS~~

~~[0001] This application claims priority to U.S. Provisional Application No. 62/685,297, filed on June 15, 2018, and U.S. Provisional Application No. 62/792,363, filed on January 14, 2019. Each of the aforementioned applications is hereby incorporated by reference in its entirety.~~

~~[0002]~~[0001] The present disclosure relates to the technical field of image and/or video coding and decoding, and in particular to method and apparatus for intra prediction.

Background

~~[0003]~~[0002] Digital video has been widely used since the introduction of DVD-discs. Before transmission the video is encoded and transmitted using a transmission medium. The viewer receives the video and uses a viewing device to decode and display the video. Over the years the quality of video has improved, for example, because of higher resolutions, color depths and frame rates. This has lead into larger data streams that are nowadays commonly transported over internet and mobile communication networks.

~~[0004]~~[0003] Higher resolution videos, however, typically require more bandwidth as they have more information. In order to reduce bandwidth requirements video coding standards involving compression of the video have been introduced. When the video is encoded the bandwidth requirements (or corresponding memory requirements in case of storage) are reduced. Often this reduction comes at the cost of quality. Thus, the video coding standards try to find a balance between bandwidth requirements and quality.

~~[0005]~~[0004] The High Efficiency Video Coding (HEVC) is an example of a video coding standard that is commonly known to persons skilled in the art. In HEVC, to split a coding unit (CU) into prediction units (PU) or transform units (TUs). The Versatile Video Coding (VVC) next generation standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Video Exploration Team (JVET). VVC is also referred to as ITU-T H.266/Next Generation Video Coding (NGVC) standard. In VVC, it removes the concepts of multiple partition types, i.e. it removes the separation of the CU, PU and TU concepts except as needed for CUs that have a size too large for the maximum transform length, and supports more flexibility for CU partition shapes.

~~[0006]~~[0005] Processing of these coding units (CUs) (also referred to as blocks) depend on their size, spatial position and a coding mode specified by an encoder. Coding modes can be

classified into two groups according to the type of prediction: intra- and inter-prediction modes. Intra prediction modes use samples of the same picture (also referred to as frame or image) to generate reference samples to calculate the prediction values for the samples of the block being reconstructed. Intra prediction is also referred to as spatial prediction. Inter-prediction modes are designed for temporal prediction and uses reference samples of previous or next pictures to predict samples of the block of the current picture.

~~[0007]~~[0006] For a rectangular block, an average value of adjacent pixels at a given intra-prediction mode is less accuracy or much complicated, which leads lower coding efficiency.

Summary

~~[0008]~~[0007] Apparatus and method for intra prediction are disclosed to obtain intra-prediction value of a current block with high accuracy or less complicated, so as to improve coding efficiency.

~~[0009]~~[0008] A first aspect of the embodiments of the present application discloses a method for intra prediction of a current block implemented in an encoder or a decoder. The method includes obtaining a width and a height of the current block, and comparing the width with the height. The method further includes determining a DC value of the current block based on the longer one of the width and the height. For example, when the width is greater than the height, a DC value of the current block is determined based on the width. When the height is greater than the width, a DC value of the current block is determined based on the height. This allows to keep the design of DC mode determination computationally simple.

~~[0010]~~[0009] Optionally, an implementation of the first aspect provides, when the width is greater than the height, the DC value is calculated by:

$$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + (W >> 1) \right) >> w,$$

$$w = \log_2(W), \sim\!\!\sout{w = \log_2(W)}$$

wherein DC denotes the DC value, W denotes the width of the current block, and p_i denotes reference sample of the current block.

~~[0011]~~[0010] Optionally, another implementation of the first aspect provides, when the height is greater than the width, the DC value is calculated by:

$$DC = \left(\left(\sum_{i=0}^{H-1} p_i \right) + (H >> 1) \right) >> h,$$

$$h = \log_2(H),$$

wherein DC denotes the DC value, H denotes the height of the current block, and p_i denotes

a reference sample of the current block.

~~[0012]~~[0011] A second aspect of the embodiments of the present application discloses a method for intra prediction of a current block implemented in an encoder or a decoder. The method includes obtaining a width and a height of the current block. The method further includes determining a DC value of the current block based on the width when the width is greater than a product of the height and a first threshold value. The method further includes determining a DC value of the current block based on the height when the height is greater than a product of the width and a second threshold value. This allows to keep the design of DC mode determination computationally simple.

~~[0013]~~[0012] Optionally, an implementation of the second aspect provides, that the first threshold value may be the same as the second threshold value. For example, the first threshold value may equal to 1.

~~[0014]~~[0013] Optionally, in the second aspect, another implementation of the aspect provides, when $W > f_{\mathrm{THR}} \cdot H$, the DC value is determined as:

$$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + (W >> 1) \right) >> w,$$

$$w = \log_2(W), \sim\!\!\!

aspect of the invention correspond to the features and implementation forms of the method according to the first aspect of the invention.

~~[0017]~~[0016] A fourth aspect of the embodiments of the present application discloses an apparatus configured for performing the method according to the second aspect of the invention. Further features and implementation forms of the apparatus according to the fourth aspect of the invention correspond to the features and implementation forms of the method according to the second aspect of the invention.

~~[0018]~~[0017] According to a fifth aspect, an embodiment of the invention relates to an apparatus for decoding intra prediction of a current block, wherein the apparatus includes a processor and a memory. The memory is storing instructions that cause the processor to perform the method according to the first aspect or the second aspect or any possible embodiment of the first or second aspect.

~~[0019]~~[0018] According to a sixth aspect, an embodiment of the invention relates to an apparatus for encoding intra prediction of a current block, wherein the apparatus includes a processor and a memory. The memory is storing instructions that cause the processor to perform the method according to the first aspect or the second aspect or any possible embodiment of the first or second aspect.

~~[0020]~~[0019] According to a seventh aspect, a computer-readable storage medium having stored thereon instructions that when executed cause one or more processors configured to encode or decode intra prediction of a current block is proposed. The instructions cause the one or more processors to perform a method according to the first or second aspect or any possible embodiment of the first or second aspect.

~~[0021]~~[0020] According to an eighth aspect, an embodiment of the invention relates to a computer program comprising program code for performing the method according to the first or second aspect or any possible embodiment of the first or second aspect when executed on a computer.

~~[0022]~~[0021] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.

~~[0023]~~[0022] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

Brief description of drawings

~~[0024]~~[0023] In the following exemplary embodiments are described in more detail with reference to the attached figures and drawings, in which:

~~[0025]~~[0024] FIG. 1 shows a schematic diagram illustrating an example of a video encoding and decoding system 100.

~~[0026]~~[0025] FIG. 2 shows a schematic diagram illustrating an example of a video encoder 200.

~~[0027]~~[0026] FIG. 3 shows a schematic diagram illustrating an example of a video decoder 300.

~~[0028]~~[0027] FIG. 4 shows a schematic diagram illustrating proposed 67 intra prediction modes.

~~[0029]~~[0028] FIG. 5(a) shows an exemplary DC intra- prediction mode for rectangular blocks.

~~[0030]~~[0029] FIG. 5(b) shows other exemplary DC intra- prediction mode for rectangular blocks.

~~[0031]~~[0030] FIG. 5(c) shows another exemplary DC intra- prediction mode for rectangular blocks.

~~[0032]~~[0031] FIG. 6 shows exemplary embodiment of DC value calculation.

~~[0033]~~[0032] FIG. 7 is a schematic diagram illustrating an exemplary structure of an apparatus.

Detailed description of embodiments

~~[0034]~~[0033] In the following description, reference is made to the accompanying drawings, which form part of the disclosure, and in which are shown, by way of illustration, specific aspects in which the invention may be placed.

~~[0035]~~[0034] For instance, it is understood that a disclosure in connection with a described method may also hold true for a corresponding device or system configured to perform the method and vice versa. For example, if a specific method step is described, a corresponding device may include a unit to perform the described method step, even if such unit is not explicitly described or illustrated in the figures. Further, it is understood that the features of the various exemplary aspects described herein may be combined with each other, unless specifically noted otherwise.

~~[0036]~~[0035] Video coding typically refers to the processing of a sequence of pictures, which form the video or video sequence. The term picture, image or frame may be used/are used synonymously in the field of video coding as well as in this application. Each picture is typically partitioned into a set of non-overlapping blocks. The encoding/decoding of the picture is typically performed on a block level where e.g. inter frame prediction or intra frame prediction are used to generate a prediction block, to subtract the prediction block from the current block (block currently processed/to be processed) to obtain a residual block, which is further

transformed and quantized to reduce the amount of data to be transmitted (compression) whereas at the decoder side the inverse processing is applied to the encoded/compressed block to reconstruct the block (video block) for representation.

~~[0037]~~[0036] Intra-prediction is a mechanism used in many video coding frameworks to increase compression efficiency in the cases where only a given frame can be involved. The DC mode is one of the most frequently selected intra-prediction modes. The basic idea behind this mode is to calculate the average value DC of adjacent pixels p_i that are typically placed on the top and left sides of a block. In conventional methods, a division operation is used to calculate the DC value, which leads more complicated calculation. To avoid the division operation, two components DC values for the reference samples p_i are calculated to obtain the DC value. Both components DC values contribute equally to the DC value, which leads to less accuracy. The present invention provides a mechanism to obtain average value of adjacent pixels with high accuracy or less complicated, so as to improve coding efficiency. In the following, embodiments of a system, an encoder, a decoder, and corresponding methods are described, which may simplify the calculation of intra-prediction value. The term “sample” in this disclosure is used to include but not limited to sample, pixel, sub-pixel, and etc.

~~[0038]~~[0037] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 100 that may utilize the techniques described in this disclosure, including techniques for encoding and decoding ~~boundary partition~~. The system 100 is not only applied to video encoding and decoding, but also applied to picture encoding and decoding. As shown in FIG. 1, system 100 includes a source device 102 that generates encoded video data to be decoded at a later time by a destination device 104. Video encoder 200 as shown in FIG. 2, is an example of a video encoder 108 of the source device 102. Video decoder 300 as shown in FIG. 3, is an example of a video decoder 116 of the destination device 104. Source device 102 and destination device 104 may comprise any of a wide range of devices, including desktop computers, notebooks (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called “smart” phones, so-called “smart” pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some cases, source device 102 and destination device 104 may be equipped for wireless communication.

~~[0039]~~[0038] Destination device 104 may receive the encoded video data to be decoded via a link 112. Link 112 may include any type of medium or device capable of moving the encoded video data from source device 102 to destination device 104. In one example, link 112 may comprise a communication medium to enable source device 102 to transmit encoded video data directly to destination device 104 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and

transmitted to destination device 104. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from source device 102 to destination device 104.

~~[0040]~~[0039] Alternatively, encoded data may be output from output interface 110 to a storage device (not shown in FIG.1). Similarly, encoded data may be accessed from the storage device by input interface 114. Destination device 104 may access stored video data from storage device via streaming or download. The techniques of this disclosure are not necessarily limited to wireless applications or settings. The techniques may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, streaming video transmissions, e.g., via the Internet, encoding of digital video for storage on a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, system 100 may be configured to support one-way or two-way video transmission to support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

~~[0041]~~[0040] In the example of FIG. 1, source device 102 includes a video source 106, a video encoder 108 and an output interface 110. In some cases, output interface 110 may include a modulator/demodulator (modem) and/or a transmitter. In source device 102, video source 106 may include a source such as a video capture device, e.g., a video camera, a video archive containing previously captured video, a video feed interface to receive video from a video content provider, and/or a computer graphics system for generating computer graphics data as the source video, or a combination of such sources. As one example, if video source 106 is a video camera, source device 102 and destination device 104 may form so-called camera phones or video phones. However, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

~~[0042]~~[0041] The captured, pre-captured, or computer-generated video may be encoded by video encoder 108. The encoded video data may be transmitted directly to destination device 104 via output interface 110 of source device 102. The encoded video data may also (or alternatively) be stored onto the storage device for later access by destination device 104 or other devices, for decoding and/or playback.

~~[0043]~~[0042] Destination device 104 includes an input interface 114, a video decoder 116, and a display device 118. In some cases, input interface 114 may include a receiver and/or a modem. Input interface 114 of destination device 104 receives the encoded video data over link 112. The encoded video data communicated over link 112, or provided on the storage device, may include a variety of syntax elements generated by video encoder 108 for use by a video decoder, such as video decoder 116, in decoding the video data. Such syntax elements may be included with the encoded video data transmitted on a communication medium, stored on a storage medium, or stored a file server.

~~[0044]~~[0043] Display device 118 may be integrated with, or external to, destination device 104. In some examples, destination device 104 may include an integrated display device and also be configured to interface with an external display device. In other examples, destination device 104 may be a display device. In general, display device 118 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

~~[0045]~~[0044] Video encoder 108 and video decoder 116 may operate according to any kind of video compression standard, including but not limited to MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), the High Efficiency Video Coding (HEVC), ITU-T H.266/Next Generation Video Coding (NGVC) standard.

~~[0046]~~[0045] It is generally contemplated that video encoder 108 of source device 102 may be configured to encode video data according to any of these current or future standards. Similarly, it is also generally contemplated that video decoder 116 of destination device 104 may be configured to decode video data according to any of these current or future standards.

~~[0047]~~[0046] Video encoder 108 and video decoder 116 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of video encoder 108 and video decoder 116 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

~~[0048]~~[0047] In video coding specifications, a video sequence typically includes a series of pictures. However, it is noted that the present disclosure is also applicable to fields in case interlacing is applied. Video encoder 108 may output a bitstream that includes a sequence of

bits that forms a representation of coded pictures and associated data. Video decoder 116 may receive a bitstream generated by video encoder 108. In addition, video decoder 116 may parse the bitstream to obtain syntax elements from the bitstream. Video decoder 116 may reconstruct the pictures of the video data based at least in part on the syntax elements obtained from the bitstream. The process to reconstruct the video data may be generally reciprocal to the process performed by video encoder 108.

~~[0049]~~[0048] FIG. 2 shows a schematic diagram illustrating an example of a video encoder 200. The video encoder 200 is not only applied to video encoding, but also applied to picture encoding. The video encoder 200 comprises an input for receiving input blocks of frames or pictures of a video stream and an output for generating an encoded video bit stream. The video encoder 200 is adapted to apply prediction, transformation, quantization, and entropy coding to the video stream. The transformation, quantization, and entropy coding are carried out respectively by a transform unit 201, a quantization unit 202 and an encoding unit 203 so as to generate as an output the encoded video bit stream.

~~[0050]~~[0049] The video stream corresponds to a plurality of frames, wherein each frame is divided into blocks of a certain size that are either intra or inter coded. The blocks of for example the first frame of the video stream are intra coded by means of an intra prediction unit 209. An intra frame is coded using only the information within the same frame, so that it can be independently decoded and it can provide an entry point in the bit stream for random access. Blocks of other frames of the video stream are inter coded by means of an inter prediction unit 210: information from coded frames, which are called reference frames, are used to reduce the temporal redundancy, so that each block of an inter coded frame is predicted from a block of the same size in a reference frame. A mode selection unit 208 is adapted to select whether a block of a frame is to be processed by the intra prediction unit 209 or the inter prediction unit 210.

~~[0051]~~[0050] For performing inter prediction, the coded reference frames are processed by an inverse quantization unit 204, an inverse transform unit 205, a filtering unit 206 (optional) so as to obtain the reference frames that are then stored in a frame buffer 207. Particularly, reference blocks of the reference frame can be processed by these units to obtain reconstructed reference blocks. The reconstructed reference blocks are then recombined into the reference frame.

~~[0052]~~[0051] The inter prediction unit 210 comprises as input a current frame or picture to be inter coded and one or several reference frames or pictures from the frame buffer 207. Motion estimation and motion compensation are applied by the inter prediction unit 210. The motion estimation is used to obtain a motion vector and a reference frame based on certain cost function. The motion compensation then describes a current block of the current frame in

terms of the transformation of a reference block of the reference frame to the current frame. The inter prediction unit 210 outputs a prediction block for the current block, wherein said prediction block minimizes the difference between the current block to be coded and its prediction block, i.e. minimizes the residual block. The minimization of the residual block is based e.g., on a rate-distortion optimization procedure.

~~[0053]~~[0052] The difference between the current block and its prediction, i.e. the residual block, is then transformed by the transform unit 201. The transform coefficients are quantized and entropy coded by the quantization unit 202 and the encoding unit 203. The encoded video bit stream comprises intra coded blocks and inter coded blocks.

~~[0054]~~[0053] FIG. 3 shows a schematic diagram illustrating an example of a video decoder 300. The video decoder 300 is not only applied to video decoding, but also applied to picture decoding. The video decoder 300 comprises particularly a frame buffer 307, an inter prediction unit 310. The frame buffer 307 is adapted to store at least one reference frame obtained from the encoded video bit stream. The inter prediction unit 310 is adapted to generate a prediction block of a current block of a current frame from a reference block of the reference frame.

~~[0055]~~[0054] The decoder 300 is adapted to decode the encoded video bit stream generated by the video encoder 200, and both the decoder 300 and the coder 200 generate identical predictions. The features of the frame buffer 307, the inter prediction unit 310 are similar to the features of the frame buffer 207, the inter prediction unit 210, of FIG. 2.

~~[0056]~~[0055] Particularly, the video decoder 300 comprises units that are also present in the video encoder 200 like e.g., an inverse quantization unit 304, an inverse transform unit 305, a filtering unit 306 (optional) and an intra prediction unit 309, which respectively correspond to the inverse quantization unit 204, the inverse transform unit 205, the filtering unit 206 and the intra prediction unit 209 of the video encoder 200. A decoding unit 303 is adapted to decode the received encoded video bit stream and to correspondingly obtain quantized residual transform coefficients. The quantized residual transform coefficients are fed to the inverse quantization unit 304 and an inverse transform unit 305 to generate a residual block. The residual block is added to a prediction block and the addition is fed to the filtering unit 306 to obtain the decoded video. Frames of the decoded video can be stored in the frame buffer 307 and serve as a reference frame for inter prediction.

~~[0057]~~[0056] The video encoder 200 may split the input video frame into blocks before coding. The term “block” in this disclosure is used for any type block or for any depth block, for example, the term “block” is included but not limited to root block, block, sub-block, leaf node, and etc. The blocks to be coded do not necessarily have the same size. One picture may include blocks of different sizes and the block rasters of different pictures of video sequence may also differ.

~~[0058]~~[0057] According to the HEVC/H.265 standard, 35 intra prediction modes are available. As shown in FIG. 4, this set contains the following modes: planar mode (the intra prediction mode index is 0), DC mode (the intra prediction mode index is 1), and directional (angular) modes that cover the 180° range and have the intra prediction mode index value range of 2 to 34 shown by black arrows in FIG. 4. To capture the arbitrary edge directions present in natural video, the number of directional intra modes is extended from 33, as used in HEVC, to 65. The additional directional modes are depicted as dotted arrows in FIG. 4, and the planar and DC modes remain the same. It is worth noting that the range that is covered by intra prediction modes can be wider than180°. In particular, 62 directional modes with index values of 3 to 64 cover the range of approximately 230°, i.e. several pairs of modes have opposite directionality. In the case of the HEVC Reference Model (HM) and JEM platforms, only one pair of angular modes (namely, modes 2 and 66) has opposite directionality as shown in FIG. 4. For constructing a predictor, conventional angular modes take reference samples and (if needed) filter them to get a sample predictor. The number of reference samples required for constructing a predictor depends on the length of the filter used for interpolation (e.g., bilinear and cubic filters have lengths of 2 and 4, respectively).

~~[0059]~~[0058] The disclosure proposes a mechanism to avoid a division operation with high accuracy while calculating an intra-prediction value, such as a DC value. The disclosure takes into account actual lengths of adjacent sides, which requires 1 multiplication per block, some other elementary arithmetic operations and a look up table (LUT). The size of the LUT is reduced as compared with a straightforward implementation of this approach. Multiplication operation and the LUT are not required for square blocks that can be important for the smallest blocks (4x4 for luma and 2x2 for chroma) to avoid such operations as multiplication for the worst case when a picture or its part is subdivided into the smallest blocks. In addition, the proposed mechanism can be combined with the conventional methods and/or other simplifications. In this case, the proposed approach is used for larger blocks and / or elongate blocks. The technique based on conventional methods is applied to blocks of smaller sizes and with the shape close to square. Besides, DC value can be estimated by taking reference samples only along with a longer side for very elongate blocks (e.g., 64x4). Details on the calculation process are described below.

~~[0060]~~[0059] FIG. 5(a) is an embodiment of a given intra-prediction mode for rectangular blocks, for example, the DC intra-prediction mode. As shown in FIG. 5(a), W≠H, the current block is a rectangular block, where W and H denote width and height of a block, respectively. Taking into account that $W = 2^w$ and $H = 2^h$ (i.e. the lengths of a block sides are powers of 2), the sum of width and height can be written as follows:

$$W + H = 2^w + 2^h = 2^w + 2^w = 2^{w+1} \quad (1)$$

When W > H,

$$W + H = 2^h\left(2^{w-h} + 1\right). \quad (2\text{-}1)$$

When H > W,

$$W + H = 2^w\left(2^{h-w} + 1\right). \quad (2\text{-}2)$$

~~[0061]~~[0060] When W > H, aspect ratios R_A of horizontally and vertically oriented blocks is:

$$R_A = 2^{w-h} \quad (3\text{-}1)$$

~~[0062]~~[0061] When H > W, aspect ratios R_A of horizontally and vertically oriented blocks is:

$$R_A = 2^{h-w} \quad (3\text{-}2)$$

~~[0063]~~[0062] Let $m = \min(w,h)$, then, formulas (2-1) and (2-2) can be rewritten as follows:

$$W + H = 2^m\left(R_A + 1\right). \quad (4)$$

~~[0064]~~[0063] The DC average value of adjacent pixels p_i is calculated as formula (5):

$$DC = \frac{\sum_{i=0}^{W+H-1} p_i}{W + H}, \quad (5)$$

~~[0065]~~[0064] Substituting formula (4) into formula (5),

$$DC = \frac{\sum_{i=0}^{W+H-1} p_i}{2^m\left(R_A + 1\right)}. \quad (6)$$

~~[0066]~~[0065] Formula (6) can be rewritten as follows:

$$DC = \left(\left(\frac{1}{R_A + 1} \cdot \sum_{i=0}^{W+H-1} p_i\right) + \left(1 << \left(m - 1\right)\right)\right) >> m. \quad (7)$$

where << and >> are left and right shift operations, respectively.

~~[0067]~~[0066] Since the partitioning depth is typically limited and known at both encoder and decoder sides a priori, in general, values of $\frac{1}{R_A + 1}$ can be tabulated using a LUT where the quantity of records is computed as follows:

$$N_{\text{LUT}} = \log_2 L_{\max} - \log_2 L_{\min} = \log_2 \frac{L_{\max}}{L_{\min}}, \quad (8)$$

where $L_{\min}$ and $L_{\max}$ are minimal and maximum lengths of a block side, respectively. According to the default settings of QuadTree plus Binary Tree (QTBT), $L_{\min}=4$ and $L_{\max}=128$, i.e. $N_{\mathrm{LUT}}=\log_2\frac{128}{4}=\log_2 32=5$. For square blocks, conventional method may be applied or one more record should be added to the LUT.

~~[0068]~~[0067] Expression $\frac{1}{R_A+1}$ takes the following values: $\frac{1}{2^1+1}=\frac{1}{3}$, $\frac{1}{2^2+1}=\frac{1}{5}$, $\frac{1}{2^3+1}=\frac{1}{9}$, $\frac{1}{2^4+1}=\frac{1}{17}$, $\frac{1}{2^5+1}=\frac{1}{33}$. To keep the calculations of DC value accurate while computing $\frac{1}{R_A+1}\cdot\sum_{i=0}^{W+H-1}p_i$, reformulate it as follows:

$$\left(\left(\frac{(1<<q)}{R_A+1}\cdot\sum_{i=0}^{W+H-1}p_i\right)+(1<<(q-1))\right)>>q, \qquad (9)$$

where q is an integer parameter that takes the maximum possible value that does not cause an overflow of the used data type. The formula (9) may provide a way to calculate DC value without division operation using fixed point multiplications and bit shift operation. Substituting expression (9) into formula (7):

$$DC=\left(\left(\left(\left(\frac{(1<<q)}{R_A+1}\cdot\sum_{i=0}^{W+H-1}p_i\right)+(1<<(q-1))\right)>>q\right)+(1<<(m-1))\right)>>m. \qquad (10).$$

Formula (10) can be simplified as follows:

$$DC=\left(\left(\frac{(1<<q)}{R_A+1}\cdot\sum_{i=0}^{W+H-1}p_i\right)+(1<<(m+q-1))\right)>>(m+q). \qquad (11).$$

Formula (10) can affect calculation accuracy but can reduce the computational complexity of formula (10). In either case, $\frac{(1<<q)}{R_A+1}$ may be tabulated and stored in the LUT. It is worth to note that the value of *q* can be different for different values of R_A to maximize the accuracy of calculations.

~~[0069]~~[0068] As shown above, a multiplication operation is required if formulas (10) or (11) is used. On the one hand, it can be critical if a picture or one of its regions is split into very small blocks (such as 4x4 for luma and 2x2 for chroma). In this situation (for example, the

square block or close to square block), conventional approach may be used. On the other hand, the conventional DC calculation approach can adversely affect the coding efficiency of DC mode if a block is very asymmetric (e.g., 32x4). So, the proposed technique based on formulas (10) and (11) can be combined with the approach based on conventional DC calculation approach, for example formulas (12)-(14).

$$DC_{\text{left}} = \left(\left(\sum_{j=0}^{H-1} p_j \right) + \left(H >> 1 \right) \right) >> h , \quad (12)$$

$$DC_{\text{up}} = \left(\left(\sum_{i=0}^{W-1} p_i \right) + \left(W >> 1 \right) \right) >> w , \quad (13)$$

$$DC = \left(DC_{\text{up}} + DC_{\text{left}} \right) >> 1. \quad (14)$$

where the reference samples p_i above the block DC_{up} and the reference samples p_j left of the block $DC_{\text{left.}}$

~~[0070]~~[0069] Another mechanism that keep the design of DC mode computationally simple for elongate blocks (e.g., 64x4) is based on the assumption:

$$DC = \frac{\left(\sum_{i=0}^{W+H-1} p_i \right) + \left(\left(W + H \right) >> 1 \right)}{W + H} \approx \frac{\left(\sum_{i=0}^{W-1} p_i \right) + \left(W >> 1 \right)}{W} = \left(\left(\sum_{i=0}^{W-1} p_i \right) + \left(W >> 1 \right) \right) >> w \text{ if}$$

$$W > f_{\text{THR}} \cdot H . \quad (15)$$

$$DC = \frac{\left(\sum_{i=0}^{W+H-1} p_i \right) + \left(\left(W + H \right) >> 1 \right)}{W + H} \approx \frac{\left(\sum_{i=0}^{H-1} p_i \right) + \left(H >> 1 \right)}{H} = \left(\left(\sum_{i=0}^{H-1} p_i \right) + \left(H >> 1 \right) \right) >> h \quad \text{if}$$

$$H > f_{\text{THR}} \cdot W . \quad (16)$$

f_{THR} is a threshold value (e.g., $f_{\text{THR}} = 8$). Formulas (15) and (16) may have different or same threshold value. So, only reference samples of the longest block side are used to calculate DC value.

~~[0071]~~[0070] Another embodiment that only longest side component is considered when calculating DC value is shown below:

$$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + \left(W >> 1 \right) \right) >> w , \quad (17)$$

if $W \geq H$,

$$DC = \left(\left(\sum_{i=0}^{H-1} p_i \right) + (H >> 1) \right) >> h, \qquad (18)$$

if $H \geqslant W$,

where $w = \log_2(W), h = \log_2(H)$, W, H are width and height (respectively) of the predicted block. In these examples, the threshold value f_{THR} is 1.

The example for equation (17) is shown at Fig. 5(b), and the example for equation (18) is shown at Fig. 5(c).

~~[0072]~~[0071] Different exemplary values of $\frac{(1<<q)}{R_A+1}$ and q are shown in Tables 1 and 2.

Table 1. The values of $\frac{(1<<q)}{R_A+1}$ for different values of *q* if it does not depend on R_A

q	Aspect ratio (R_A)					
	1 (square block)	**2**	**4**	**8**	**16**	**32**
5	32	11	6	4	2	1
6	64	21	13	7	4	2
7	128	43	26	14	8	4
8	256	85	51	28	15	8
9	512	171	102	57	30	16
10	1024	341	205	114	60	31
11	2048	683	410	228	120	62
12	4096	1365	819	455	241	124

Table 2. The values of q_{R_A} and $\frac{(1<<q)}{R_A+1}$ for different values of q_{const} if $q = q_{\mathrm{const}} + q_{R_A}$ depends on R_A

q_{const}	Aspect ratio (R_A)											
	1 (square block)		2		4		8		16		32	
	q_{R_A}	$\frac{(1<<q)}{R_A+1}$	q_{R_A}	$\frac{(1<<q)}{R_A+1}$	q_{R_A}	$\frac{(1<<q)}{R_A+1}$	q_{R_A}	$\frac{(1<<q)}{R_A+1}$	q_{R_A}	$\frac{(1<<q)}{R_A+1}$	q_{R_A}	$\frac{(1<<q)}{R_A+1}$
5	1	32	1	21	2	23	3	28	4	30	3	31
6	1	64	1	43	2	51	3	57	4	60	3	62
7	1	128	1	85	2	102	3	114	4	120	3	124
8	1	256	1	171	2	205	3	228	4	241	3	248
9	1	512	1	341	2	410	3	455	4	482	3	496
10	1	1024	1	683	2	819	3	910	4	964	3	993
11	1	2048	1	1365	2	1638	3	1820	4	1928	3	1986
12	1	4096	1	2731	2	3277	3	3641	4	3855	3	3972

~~[0073]~~[0072] FIG 6. shows an exemplary embodiment of DC value calculation using the proposed invention. The input of the process are the width and the height of a block and the corresponding logarithms of width and height.

~~[0074]~~[0073] In the embodiment, three block shape cases are considered. Specifically, one case is a square shape when width and height are equal, and ~~tow~~ two rectangular shape cases when width is longer than height and vice versa.

~~[0075]~~[0074] The first step 601 is to distinguish a case of a square shape by comparing width and height values. For the square shape case DC value is computed conventionally at step 620 and the rest of the steps are skipped.

~~[0076]~~[0075] The second step 602, which includes step 6021 and step 6022, is to distinguish between the rest two rectangular shape cases at step 6021 and store the result in “isHor” variable at step 6022. For both of these cases values of aspect ratio R_A~~RA~~, and larger side index s is assigned correspondingly.

~~[0077]~~[0076] The third step 603 is to define whether the block is small by comparing s to a predefined threshold, e.g. 3, and the block is not elongate one. The later condition is checked by thresholding R_A by another threshold value (e.g. 2). If both of these condition are fulfilled,

at step 630, the DC value is calculated using formulas (12)-(14) described above and the rest of the steps are skipped.

~~[0078]~~[0077] The forth step 604 is to define whether the block is elongated and large enough using the conditions shown in the figure 6, for example, comparing s to a predefined threshold, e.g. 4, and the block is not elongate one. The later condition is checked by thresholding 2^{RA} by another threshold value f_{THR}. Denotations of variables are disclosed in the description above. If these conditions are met, the next step 605 is to decide whether variable “isHor” is true or false. DC value is calculated from the reference samples of the left side (if “isHor” is false) at step 607, or from the top side (if “isHor” is true) at step 606.

~~[0079]~~[0078] The last step 608 of the FIG. 6 is to calculate the DC value by default, using formulas (10)-(11).

~~[0080]~~[0079] Another embodiment of this invention considers the colorplane for which prediction is being performed. In specific, condition “W==H?” in FIG.6 could be amended to “W==H or isChroma?”. “isChroma” variable is set to true if the block being predicted belongs to chroma color plane (e.g., Cb or Cr ~~one~~) and is set to false if the block belongs to luma color plane.

~~[0081]~~[0080] FIG. 7 is a block diagram of an apparatus 700 that can be used to implement various embodiments. The apparatus 700 may be the source device 102 as shown in FIG. 1, or the video encoder 200 as shown in FIG. 2, or the destination device 104 as shown in FIG. 1, or the video decoder 300 as shown in FIG. 3. Additionally, the apparatus 700 can host one or more of the described elements. In some embodiments, the apparatus 700 is equipped with one or more input/output devices, such as a speaker, microphone, mouse, touchscreen, keypad, keyboard, printer, display, and the like. The apparatus 700 may include one or more central processing units (CPUs) 710, a memory 720, a mass storage 730, a video adapter 740, and an I/O interface 760 connected to a bus. The bus is one or more of any type of several bus architectures including a memory bus or memory controller, a peripheral bus, a video bus, or the like.

~~[0082]~~[0081] The CPU 710 may have any type of electronic data processor. The memory 720 may have, or be, any type of system memory such as static random access memory (SRAM), dynamic random access memory (DRAM), synchronous DRAM (SDRAM), read-only memory (ROM), a combination thereof, or the like. In an embodiment, the memory 720 may include ROM for use at boot-up, and DRAM for program and data storage for use while executing programs. In embodiments, the memory 720 is non-transitory. The mass storage 730 includes any type of storage device that stores data, programs, and other information and to make the data, programs, and other information accessible via the bus. The mass storage

730 includes, for example, one or more of a solid state drive, hard disk drive, a magnetic disk drive, an optical disk drive, or the like.

~~[0083]~~[0082] The video adapter 740 and the I/O interface 760 provide interfaces to couple external input and output devices to the apparatus 700. For example, the apparatus 700 may provide SQL command interface to clients. As illustrated, examples of input and output devices include a display 790 coupled to the video adapter 740 and any combination of mouse/keyboard/printer 770 coupled to the I/O interface 760. Other devices may be coupled to the apparatus 700, and additional or fewer interface cards may be utilized. For example, a serial interface card (not shown) may be used to provide a serial interface for a printer.

~~[0084]~~[0083] The apparatus 700 also includes one or more network interfaces 750, which includes wired links, such as an Ethernet cable or the like, and/or wireless links to access nodes or one or more networks 780. The network interface 750 allows the apparatus 700 to communicate with remote units via the networks 780. For example, the network interface 750 may provide communication to database. In an embodiment, the apparatus 700 is coupled to a local-area network or a wide-area network for data processing and communications with remote devices, such as other processing units, the Internet, remote storage facilities, or the like.

~~[0085]~~[0084] The advantages of the present disclose are as follows:

~~[0086]~~[0085] Multiplication is used with high accuracy, for example formulas (7), (10) and (11), to replace a division operation in the conventional method;

~~[0087]~~[0086] size of the required LUT is kept small;

~~[0088]~~[0087] the proposed technique can be combined with other approaches to reduce the computational complexity of the worst case for DC mode, for example, combine with formulas (12)-(14). Different blocks may use same or different techniques, such as, block 1 may use formula (7), block 2 may use formula (10), block 3 may use formula (11), block 4 may use formulas (12)-(14), etc.; or blocks use same formula (7), or (10), or (11), or (15) or (16) ; or any combinations between the proposed techniques; or any combinations between the proposed techniques and conventional methods; and/or

~~[0089]~~[0088] for very elongate blocks (e.g., 64x4), the shortest side can be skipped to simplify calculations, for example, formulas (15) and (16).

~~[0090]~~[0089] The embodiments may be applied, for example, to other intra-prediction techniques such as DWDIP (Distance-Weighted Directional Intra-Prediction) or LIP (Linear Intra-Prediction), respectively. The step for gradual component of secondary reference samples for DWDIP uses the same mechanism.

~~[0091]~~[0090] Implementations of the subject matter and the operations described in this disclosure may be implemented in digital electronic circuitry, or in computer software,

firmware, or hardware, including the structures disclosed in this disclosure and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this disclosure may be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions may be encoded on an artificially-generated propagated signal, for example, a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium, for example, the computer-readable medium, may be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium may be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer storage medium may also be, or be included in, one or more separate physical and/or non-transitory components or media (for example, multiple CDs, disks, or other storage devices).

~~[0092]~~[0091] In some implementations, the operations described in this disclosure may be implemented as a hosted service provided on a server in a cloud computing network. For example, the computer-readable storage media may be logically grouped and accessible within a cloud computing network. Servers within the cloud computing network may include a cloud computing platform for providing cloud-based services. The terms “cloud,” “cloud computing,” and “cloud-based” may be used interchangeably as appropriate without departing from the scope of this disclosure. Cloud-based services may be hosted services that are provided by servers and delivered across a network to a client platform to enhance, supplement, or replace applications executed locally on a client computer. The circuit may use cloud-based services to quickly receive software upgrades, applications, and other resources that would otherwise require a lengthy period of time before the resources may be delivered to the circuit.

~~[0093]~~[0092] A computer program (also known as a program, software, software application, script, or code) may be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it may be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program may be stored in a portion of a file that holds other programs or data (for example, one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files

(for example, files that store one or more modules, sub-programs, or portions of code). A computer program may be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

~~[0094]~~[0093] The processes and logic flows described in this disclosure may be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows may also be performed by, and apparatus may also be implemented as, special purpose logic circuitry, for example, an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

~~[0095]~~[0094] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing actions in accordance with instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, for example, magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer may be embedded in another device, for example, a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device (for example, a universal serial bus (USB) flash drive), to name just a few. Devices suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, for example, EPROM, EEPROM, and flash memory devices; magnetic disks, for example, internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory may be supplemented by, or incorporated in, special purpose logic circuitry.

~~[0096]~~[0095] While this disclosure contains many specific implementation details, these should not be construed as limitations on the scope of any implementations or of what may be claimed, but rather as descriptions of features specific to particular implementations ~~of particular implementations~~. Certain features that are described in this disclosure in the context of separate implementations may also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation may also be implemented in multiple implementations separately or in any suitable subcombination. Moreover, although features may be described above as acting in

certain combinations and even initially claimed as such, one or more features from a claimed combination may in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

~~[0097]~~[0096] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems may generally be integrated together in a single software product or packaged into multiple software products.

~~[0098]~~[0097] Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims may be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

CLAIMS

1. A method for calculating a DC value for DC intra prediction mode of a current block of a picture, the method implemented by a decoder comprising:

obtaining an aspect ratio of the current block according to a width and a height of the current block;

obtaining a weighting factor based on the aspect ratio~~, and~~;

determining an intra-prediction value of the current block based on the weighting factor~~.~~

~~2. The method of claim 1~~, wherein the current block is a rectangular block, and the intra-prediction value is a DC value~~.~~; and

reconstructing the current block based on the intra-prediction value,

wherein the width W of the current block is $W = 2^w$ and the height H of the current block is $H = 2^h$,

wherein when W > H, the aspect ratio R_A is $R_A = 2^{w-h}$, and

wherein when H > W, the aspect ratio R_A is $R_A = 2^{h-w}$.

~~3~~2. The method of claim 1~~ or 2~~, wherein determining the intra-prediction value of the current block is based on a multiplication ~~between~~involving the weighting factor and ~~adjacent pixels~~reference samples ~~of~~ the current block in the picture.

~~4~~3. The method of ~~any one of claims~~claim 1~~ to 3~~, wherein the weighting factor ~~corresponds~~is proportional to an inverse of a sum of the aspect ratio of the current block and 1.

~~5~~4. The method of claim ~~4~~3, wherein the weighting factor equals $\frac{1}{R_A + 1}$, wherein R_A is the aspect ratio.

~~6~~5. The method of claim ~~4~~3, wherein the weighting factor equals $\frac{(1<<q)}{R_A+1}$, wherein R_A is the aspect

ratio, and q is an integer having a maximum possible value that does not cause an overflow of a data type.

~~7~~6. The method of ~~any one of claims~~claim 1 ~~to 6~~, wherein a value of the weighting factor is tabulated and stored in a look up table (LUT).

~~8. The method of any one of claims 1 to 7, wherein the width W of the current block and a height H of the current block,~~

~~when W > H, the aspect ratio =;~~

~~When H > W, the aspect ratio R_A =.~~

~~9~~. The method of claim ~~8, when the intra-prediction value is a DC value,~~ 1, wherein the DC value is determined as:

~~,~~

$$DC = \frac{\sum_{i=0}^{W+H-1} p_i}{2^m \left(R_A + 1\right)},$$

wherein ~~,~~ $m = \min\left(w, h\right)$, p_i is ~~an adjacent pixel~~a reference sample of the current block~~,~~ that is on a left side or a top side of the current block in the picture, $\frac{1}{R_A + 1}$ is the weighting factor.

~~10~~8. The method of claim ~~8 or 9, when the intra-prediction value is a DC value,~~1, wherein the DC value is determined as:

~~Wherein ,~~ $DC = \left(\left(\frac{1}{R_A + 1} \cdot \sum_{i=0}^{W+H-1} p_i\right) + \left(1 << \left(m - 1\right)\right)\right) >> m$

wherein $m = \min\left(w, h\right)$, p_i is ~~an adjacent pixel~~a reference sample of the current block~~,~~ that is on a left side or a top side of the current block in the picture , $\frac{1}{R_A + 1}$ is the weighting factor, << and >> are left and right shift operations, respectively.

~~11~~9. The method of claim ~~8, when the intra-prediction value is a DC value,~~1, wherein the DC value is determined as:

$$DC = \left(\left(\left(\left(\frac{(1 << q)}{R_A + 1} \cdot \sum_{i=0}^{W+H-1} p_i \right) + (1 << (q-1)) \right) >> q \right) + (1 << (m-1)) \right) >> m$$

wherein ~~,~~ $m = \min(w, h)$, p_i is ~~an adjacent pixel~~a reference sample of the current block~~,~~ that is on a left side or a top side of the current block in the picture, $\frac{(1<<q)}{R_A+1}$ is the weighting factor, q is an integer parameter that takes a maximum possible value that does not cause an overflow of a data type.

~~12~~10. The method of claim ~~8 or 11, when the intra-prediction value is a DC value,~~1, wherein the DC value is determined as:

$$DC = \left(\left(\frac{(1 << q)}{R_A + 1} \cdot \sum_{i=0}^{W+H-1} p_i \right) + (1 << (m+q-1)) \right) >> (m+q)$$

wherein ~~,~~ $m = \min(w, h)$, p_i is ~~an adjacent pixel~~a reference sample of the current block~~,~~ that is on a left side or a top side of the current block in the picture, $\frac{(1<<q)}{R_A+1}$ is the weighting factor, q is an integer parameter that takes a maximum possible value that does not cause an overflow of a data type.

~~13~~11. The method of ~~any one of claims~~claim 1~~-12,~~ , wherein:

when ~~,~~ $W > f_{\mathrm{THR}} \cdot H$, f_{THR} is a threshold value, the width W of the current block $W = 2^w$ and ~~a~~the height H of the current block~~, the intra-prediction value is a DC value~~ $H = 2^h$, and w and h are integers, the DC value is determined as:

~~14~~ $$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + (W >> 1) \right) >> w$$.

p_i is a reference sample of the current block that is on a top side of the current block in the picture.

12. The method of ~~any one of claims~~claim 1~~-13,~~, wherein

when ~~,~~ $H > f_{THR} \cdot W$, f_{THR} is a threshold value, the width W of the current block $W = 2^w$ and a height H of the current block~~, the intra-prediction value is a DC value,~~ $H = 2^h$, and w and h are integers, the DC value is determined as:

~~.~~

~~15. The method of any one of claims 1-14, when the method~~ $DC = \left(\left(\sum_{i=0}^{H-1} p_i \right) + \left(H >> 1 \right) \right) >> h$,

p_i is ~~applied to blocks only of~~ a ~~color plane of~~reference sample of the current block that is on a left side of the current block in the picture.

~~16~~

13. A method for calculating a DC value for DC intra prediction mode of a current block of a picture, the method implemented by a decoder comprising:

obtaining a width and a height of the current block;

comparing the width with the height;

when the width is greater than the height, determining a DC value of the current block based on ~~the width; and~~

reference samples only in a neighboring block on a top side of the current block, or when the height is greater than the width, determining a DC value of the current block based on ~~the height.~~reference samples only in a neighboring block on a left side of the current block; and

reconstructing the current block based on the DC value,

wherein the width W of the current block is $W = 2^w$ and the height H of the current block is $H = 2^h$,

wherein when W > H, the aspect ratio R_A is $R_A = 2^{w-h}$, and

wherein when H > W, the aspect ratio R_A is $R_A = 2^{h-w}$

~~17.~~

14. The method of claim ~~16~~13, wherein when the width is greater than the height, the DC value is calculated by:

;

$$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + (W >> 1) \right) >> w ,$$

$$w = \log_2 (W) ,$$

wherein DC denotes the DC value, W denotes the width of the current block, and ~~denotes a reference sample~~ p_i, i=0,...,W-1 denote the reference samples only in the neighboring block on the top side of the current block.

~~18~~15. The method of claim ~~16~~13, wherein when the height is greater than the width, the DC value is calculated by:

;

;

$$DC = \left(\left(\sum_{i=0}^{H-1} p_i \right) + (H >> 1) \right) >> h ,$$

$$h = \log_2 (H) ,$$

wherein DC denotes the DC value, H denotes the height of the current block, and ~~denotes a reference sample~~ p_i, i=0,...,H-1, denote the reference samples only in the neighboring block on the left side of the current block.

~~19~~

16. A ~~method~~decoder for calculating a DC value for DC intra prediction mode of a current block of a picture, ~~the method~~ comprising:

~~obtaining a width and a height of the current block;~~

when the width is greater than a product of the height and a first threshold value, determining a DC value of the current block based on the width; and

when the height is greater than a product of the width and a second threshold value, determining a DC value of the current block based on the height.

20. The method of claim 19, wherein the first threshold value is the same as the second threshold value.

21. The method of claim 19 or 20, wherein the first threshold value equals to 1.

22. The method of any one of claims 19-21, wherein when , the DC value is determined as:

;

wherein DC denotes the DC value, denotes the first threshold value, W denotes the width of the current block, H denotes the height of the current block, and denotes a reference sample of the current block.

23. The method of any one of claims 19-21, wherein when , the DC value is determined as:

;

;

wherein DC denotes the DC value, denotes the second threshold value, W denotes the width of the current block, H denotes the height of the current block, and denotes a reference sample of the current block..

---

24. An encoder (108; 200) comprising processing circuitry for carrying out the method according to any one of claims 1 to 15.

25. A decoder (116; 300) comprising processing circuitry for carrying out the method according to any one of claims 1 to 15.

26. A computer program product comprising a program code for performing the method according

to any one of claims 1 to 15.

27. A decoder for boundary partition of a current block of a picture, comprising:

—one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the decoder to carry out the method according to any one of claims 1 to 15.operations comprising :

28. An encoder for boundary partition of a current block of a picture, comprising:

— one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and — storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the encoder to carry out the method according to any one of claims 1 to 15.

29. An encoder (108; 200) comprising processing circuitry for carrying out the method according to any one of claims 16 to 18.

30. A decoder (116; 300) comprising processing circuitry for carrying out the method according to any one of claims 16 to 18.

31. A computer program product comprising a program code for performing the method according to any one of claims 16 to 18.

32. A decoder for boundary partition of a obtaining a width and a height of the current block;

comparing the width with the height;

when the width is greater than the height, determining a DC value of the current block of a picture, comprising:

— one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and — storing programming for execution by the processors, wherein the programming, when executed by the

processors, configures the decoder to carry out the method according to any one of claims 16 to 18.

33. An encoder for boundary partition of a based on reference samples only in a neighboring block on a top side of the current block of a picture, comprising:
one, or more processors; and
a non-transitory computer-readable storage medium coupled to when the processors and storing programming for execution byheight is greater than the processors, wherein width, determining a DC value of the programming, when executed by the processors, configures the encoder to carry out the method according to any one of claims 16 to 18.

34. An encoder (108; 200) comprising processing circuitry for carrying out the method according to any one of claims 19 to 23.

35. A decoder (116; 300) comprising processing circuitry for carrying out the method according to any one of claims 19 to 23.

36. A computer program product comprising a program code for performing the method according to any one of claims 19 to 23

37. A decoder for boundary partition of a current block of a picture, comprising:
one or more processors; and
a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the decoder to carry out the method according to any one of claims 19 to 23.

38. An encoder for boundary partition of a based on reference samples only in a neighboring block on a left side of the current block of a picture, comprising:
one or more processors; and

reconstructing the current block based on the DC value,

wherein the width W of the current block is $W = 2^w$ and the height H of the current block is $H = 2^h$,

wherein when W > H, the aspect ratio R_A is $R_A = 2^{w-h}$, and

wherein when H > W, the aspect ratio R_A is $R_A = 2^{h-w}$ ~~a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the encoder to carry out the method according to any one of claims 19 to 23.~~ .

17. The decoder of claim 16, wherein when the width is greater than the height, the DC value is determined by:

$$DC = \left(\left(\sum_{i=0}^{W-1} p_i \right) + (W >> 1) \right) >> w,$$

$$w = \log_2(W),$$

wherein DC denotes the DC value, W denotes the width of the current block, and p_i, i=0,...,W-1, denote the reference samples only in the neighboring block on the top side of the current block.

18. The decoder of claim 16, wherein when the height is greater than the width, the DC value is determined by:

$$DC = \left(\left(\sum_{i=0}^{H-1} p_i \right) + (H >> 1) \right) >> h,$$

$$h = \log_2(H),$$

wherein DC denotes the DC value, H denotes the height of the current block, and p_i, i=0,...,H-1, denote the reference samples only in the neighboring block on the left side of the current block.