(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 311 569 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
06.10.2021 Bulletin 2021/40

(21) Application number: **16734804.4**

(22) Date of filing: **17.06.2016**

(51) Int Cl.:
H04N 19/176 $^{(2014.01)}$
H04N 19/11 $^{(2014.01)}$
H04N 19/136 $^{(2014.01)}$
H04N 19/59 $^{(2014.01)}$
H04N 19/593 $^{(2014.01)}$
H04N 19/463 $^{(2014.01)}$
H04N 19/80 $^{(2014.01)}$

(86) International application number:
PCT/US2016/038182

(87) International publication number:
WO 2016/205712 (22.12.2016 Gazette 2016/51)

(54) **INTRA PREDICTION AND INTRA MODE CODING**
INTRAPRÄDIKTIONS- UND INTRAMODUSCODIERUNG
INTRAPRÉDICTION ET CODAGE À MODES INTRA

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **18.06.2015 US 201562181744 P**
14.07.2015 US 201562192310 P
16.06.2016 US 201615184103

(43) Date of publication of application:
25.04.2018 Bulletin 2018/17

(73) Proprietor: **Qualcomm Incorporated**
San Diego, CA 92121-1714 (US)

(72) Inventors:
- **ZHAO, Xin**
  San Diego, California 92121-1714 (US)
- **CHEN, Jianle**
  San Diego, California 92121-1714 (US)
- **ZHANG, Li**
  San Diego, California 92121-1714 (US)
- **KARCZEWICZ, Marta**
  San Diego, California 92121-1714 (US)

(74) Representative: **Loveless, Ian Mark**
Reddie & Grose LLP
The White Chapel Building
10 Whitechapel High Street
London E1 8QS (GB)

(56) References cited:
- **Jani Lainema ET AL: "Chapter 4 Intra-Picture Prediction in HEVC" In: "High Efficiency Video Coding (HEVC)", 1 August 2014 (2014-08-01), Springer International Publishing, XP055292569, ISBN: 978-3-319-06894-7 vol. 29, pages 91-112, DOI: 10.1007/978-3-319-06895-4_4, page 91, paragraph 4.1 - page 93; table 4.1 page 97, paragraph 4.3.1 - page 100 page 103, paragraph 4.4 - page 104 page 106, paragraph 4.5 - page 107**
- **LIANG ZHAO ET AL: "Fast mode decision algorithm for intra prediction in HEVC", VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2011 IEEE, IEEE, 6 November 2011 (2011-11-06), pages 1-4, XP032081373, DOI: 10.1109/VCIP.2011.6115979 ISBN: 978-1-4577-1321-7**
- **JIANLE CHEN ET AL: "Further improvements to HMKTA-1.0", 52. VCEG MEETING; 19-6-2015 - 26-6-2015; WARSAW; (VIDEO CODING EXPERTS GROUP OF ITU-T SG.16),, no. VCEG-AZ07, 19 June 2015 (2015-06-19), XP030003885,**
- **ANONYMOUS: "The H.264 Advanced Video Compression Standard, 2nd Edition, chapter 6, H.264 Prediction, Iain E. Richardson", NOT KNOWN,, 20 April 2010 (2010-04-20), XP030001637,**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

[0001] This disclosure relates to video encoding and decoding.

BACKGROUND

[0002] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, tablet computers, e-book readers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, so-called "smart phones," video teleconferencing devices, video streaming devices, and the like. Digital video devices implement video compression techniques, such as those described in the standards defined by 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) standard, and extensions of such standards. The video devices may transmit, receive, encode, decode, and/or store digital video information more efficiently by implementing such video compression techniques.

[0003] Video compression techniques perform spatial (intra-picture) prediction and/or temporal (inter-picture) prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video slice (i.e., a video frame or a portion of a video frame) may be partitioned into video blocks. Video blocks in an intra-coded (I) slice of a picture are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same picture. Video blocks in an inter-coded (P or B) slice of a picture may use spatial prediction with respect to reference samples in neighboring blocks in the same picture or temporal prediction with respect to reference samples in other reference pictures. Pictures may be referred to as frames, and reference pictures may be referred to as reference frames.

[0004] Spatial or temporal prediction results in a predictive block for a block to be coded. Residual data represents pixel differences between the original block to be coded and the predictive block. An inter-coded block is encoded according to a motion vector that points to a block of reference samples forming the predictive block, and the residual data indicates the difference between the coded block and the predictive block. An intra-coded block is encoded according to an intra-coding mode and the residual data. For further compression, the residual data may be transformed from the pixel domain to a transform domain, resulting in residual coefficients, which then may be quantized. The quantized coefficients, initially arranged in a two-dimensional array, may be scanned in order to produce a one-dimensional vector of coefficients, and entropy coding may be applied to achieve even more compression.

[0005] Chapter 4 of the book "High Efficiency Video Coding (HEVC), Algorithms and Architectures" by Vivienne Sze et al describes intra-picture prediction in HEVC. The paper "Fast Mode Decision Algorithm for Intra Prediction in HEVC" by Zhao et al, Proceedings the IEEE Conference on Visual Communication and Image Processing (VCIP), 2011, describes techniques for speeding up the mode selection process by reducing the candidate modes tested using rate-distortion optimization.

SUMMARY

[0006] In general, this disclosure describes techniques for intra prediction and intra prediction mode coding, which may be used in the context of advanced video codecs, such as extensions of the High Efficiency Video Coding (HEVC) standard, or the next generation of video coding standards. The scope of protection is defined by the independent claims, to which reference should now be made. Optional features are included in the dependent claims.

[0007] According to one aspect of the claimed invention, there is provided a method of encoding block of video data that includes selecting, based on intra prediction modes used to encode one or more neighboring blocks of the block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes; encoding syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and encoding the block of video data based on the selected intra prediction mode using 1/32 pel accuracy. Encoding the block of video data comprises: for each respective sample of a predictive block of the current PU: determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and calculating a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel interpolation accuracy.

[0008] According to another aspect of the claimed invention, there is provided a method of decoding a block of video data that includes selecting, based on intra prediction modes used to decode one or more neighboring blocks of the block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes; decoding syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes;; and reconstructing the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy. Reconstructing the block of video data comprises: for each respective sample of a predictive block of the current PU: determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and calculating a value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

[0009] According to another aspect of the claimed invention, there is provided a device for encoding video data that includes a memory configured to store the video data; and one or more processors configured to: select, based on intra prediction modes used to encode one or more neighboring blocks of a block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes; encode syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and encode the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy. Encoding the block of video data comprises: for each respective sample of a predictive block of the current PU: determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and calculating a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

[0010] According to another aspect of the claimed invention, there is provided a device for decoding video data that includes a memory configured to store the video data; and one or more processors configured to: select, based on intra prediction modes used to decode one or more neighboring blocks of a block of video data in a current picture of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes; decode syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and reconstruct the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy. Reconstructing the block of video data comprises: for each respective sample of a predictive block of the current PU: determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and calculating a value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

[0011] According to another aspect of the claimed invention, there is provided a computer readable medium that stores instructions that, when executed by one or more processors cause the one or more processors to perform the method of encoding or decoding.

[0012] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0013]

FIG. 1 is a block diagram illustrating an example video coding system that may utilize the techniques described in this disclosure.

FIG. 2 is a conceptual diagram illustrating an example of intra prediction of a block of video data.

FIG. 3 is a conceptual diagram illustrating an example of intra prediction modes and corresponding mode indexes.

FIG. 4 is a conceptual diagram illustrating an example technique for generating a prediction sample for a block of video data according to a planar intra prediction mode.

FIG. 5 is a conceptual diagram illustrating an example technique for generating a prediction sample for a block of video data according to an angular intra prediction mode.

FIG. 6 is a block diagram illustrating an example video encoder that may implement the techniques described in this disclosure.

FIG. 7 is a block diagram illustrating an example video decoder that may implement the techniques described in this disclosure.

FIG. 8 is a conceptual diagram illustrating an example of intra prediction modes and corresponding mode indexes according to the techniques of this disclosure.

FIG. 9 is a conceptual diagram illustrating example intra prediction angles according to the techniques of this disclosure.

FIG. 10 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 11 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

FIG. 12 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 13 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

Fig. 14 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 15 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

FIG. 16 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 17 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

FIG. 18 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 19 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

FIG. 20 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure.

FIG. 21 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure.

DETAILED DESCRIPTION

[0014] In general, this disclosure describes techniques for intra prediction and intra prediction mode coding, which may be used in the context of advanced video codecs, such as extensions of the HEVC standard, or the next generation of video coding standards. For example, this disclosure describes techniques for deriving, selecting and/or signaling most-probable modes (MPMs) for intra prediction. Examples described in this disclosure also include techniques for intra prediction using an increased number of angular modes. The techniques for intra prediction using an increased number of angular modes may include techniques for applying an *N*-tap Intra interpolation filter, where N is larger than 2. The disclosure also describes techniques that may allow multiple Intra prediction directions per block, e.g. respective directions for sub-blocks of the block, which may not require additional signaling of multiple intra prediction modes.

[0015] Intra prediction and intra mode coding are techniques that may be used in the context of advanced video codecs, such as extensions of the High Efficiency Video Coding (HEVC) standard, or the next generation of video coding standards.

[0016] In intra mode coding in HEVC, for each intra prediction unit (PU), a selected intra prediction mode is signaled. To select the intra prediction mode, three intra modes are first identified, which are assumed to have higher probability to be actually selected, namely the Most Probable Modes (MPMs). In HEVC, there are 35 modes for the intra prediction of a luma block, including a planar intra prediction mode, a DC intra prediction mode and 33 different prediction directions associated with respectively angular intra prediction modes.

[0017] The 35 fixed prediction angles of the existing design of Intra prediction in HEVC may be inefficient to capture very flexible edge direction distributions. However, applying more prediction angles may result in undesired encoder complexity increase. For example, a direct extension to 65 prediction angles based on the HEVC reference software may require an approximately doubled number of SATD checks. Furthermore, the efficiency of current intra mode coding, using three MPMs, in HEVC can be limited because it may not accurately correspond to the actual probability distribution of all available intra prediction modes.

[0018] This disclosure describes techniques that may remediate one or more of these deficiencies in HEVC. For example, in accordance with some techniques of this disclosure, a video coder may derive more than three MPMs. In some such examples, the video coder may use context modeling for decoding one or more bins of an MPM index indicating which of the MPMs is a selected MPM for a current PU. In some examples where the video coder derives more than three MPMs, the video coder may define, among the MPMs, a representative intra prediction mode for the left neighboring column and using the representative intra prediction mode for the left neighboring column as the MPM for the left neighboring column and/or define a representative intra prediction mode for the above neighboring row and using the representative intra prediction mode for the above neighboring row as the MPM for the above neighboring row. Furthermore, in some examples where the video coder derives more than three MPMs, the video coder may select one or more additional angular MPMs based on similarity with an angular mode already among the MPMs. In this example, the similarity is determined based on at least one of intra prediction mode index differences or intra prediction angle differences. Furthermore, in some example techniques of this disclosure, a non-MPM index may be encoded in the bitstream as a code word shorter than $\lceil \log_2 N \rceil$ bits if the non-MPM index satisfies a criterion and is encoded as a fixed length code with $\lceil \log_2 N \rceil$ bits otherwise, wherein there is a total of N available values of the non-MPM index.

[0019] In some examples of this disclosure where the video coder may use more than 33 angular intra prediction modes, the video coder may use an interpolation filter on neighboring reconstructed samples in which the interpolation has 1/32-pel accuracy. In some examples of this disclosure, the video coder may calculate a value of a respective sample by applying an N-tap intra interpolation filter to neighboring reconstructed samples to interpolate a value at the determined fractional position, wherein N is greater than 2.

[0020] FIG. 1 is a block diagram illustrating an example video coding system 10 that may utilize the techniques of this disclosure. As used herein, the term "video coder" refers generically to both video encoders and video decoders. In this disclosure, the terms "video coding" or "coding" may refer generically to either video encoding or video decoding.

[0021] As shown in FIG. 1, video coding system 10 includes a source device 12 and a destination device 14. Source device 12 generates encoded video data. Accordingly, source device 12 may be referred to as a video encoding device or a video encoding apparatus. Destination device 14 may decode the encoded video data generated by source device 12. Accordingly, destination device 14 may be referred to as a video decoding device or a video decoding apparatus. Source device 12 and destination device 14 may be examples of video coding devices or video coding apparatuses. Source device 12 and destination device 14 may comprise a wide range of devices, including desktop computers, mobile computing devices, notebook (e.g., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called "smart" phones, televisions, cameras, display devices, digital media players, video gaming consoles, in-car computers, or the like.

[0022] Destination device 14 may receive encoded video data from source device 12 via a channel 16. Channel 16 may comprise one or more media or devices capable of moving the encoded video data from source device 12 to destination device 14. In some examples, channel 16 may comprise one or more communication media that enable source device 12 to transmit encoded video data directly to destination device 14 in real-time. In this example, source device 12 may modulate the encoded video data according to a communication standard, such as a wireless communication protocol, and may transmit the modulated video data to destination device 14. The one or more communication media may include wireless and/or wired communication media, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The one or more communication media may form part of a packet-based network, such as a local area network, a wide-area network, or a global network (e.g., the internet). The one or more communication media may include routers, switches, base stations, or other equipment that facilitate communication from source device 12 to destination device 14.

[0023] In some examples, channel 16 may include a storage medium that stores encoded video data generated by source device 12. In this example, destination device 14 may access the storage medium, e.g., via disk access or card access. The storage medium may include a variety of locally-accessed data storage media such as Blu-ray discs, DVDs, CD-ROMs, flash memory, or other suitable digital storage media for storing encoded video data.

[0024] In some examples, channel 16 may include a file server or another intermediate storage device that stores encoded video data generated by source device 12. In this example, destination device 14 may access encoded video data stored at the file server or other intermediate storage device via streaming or download. The file server may be a type of server capable of storing encoded video data and transmitting the encoded video data to destination device 14. Example file servers include web servers (e.g., for a website), file transfer protocol (FTP) servers, network attached storage (NAS) devices, and local disk drives.

[0025] Destination device 14 may access the encoded video data through a standard data connection, such as an internet connection. Example types of data connections may include wireless channels (e.g., Wi-Fi connections), wired connections (e.g., digital subscriber line (DSL), cable modem, etc.), or combinations of both that are suitable for accessing encoded video data stored on a file server. The transmission of encoded video data from the file server may be a streaming transmission, a download transmission, or a combination of both.

[0026] The techniques of this disclosure are not limited to wireless applications or settings. The techniques may be applied to video coding in support 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 video data for storage on a data storage medium, decoding of video data stored on a data storage medium, or other applications. In some examples, video coding system 10 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.

[0027] FIG. 1 is merely an example and the techniques of this disclosure may apply to video coding settings (e.g., video encoding or video decoding) that do not necessarily include any data communication between the encoding and decoding devices. In other examples, data is retrieved from a local memory, streamed over a network, or the like. A video encoding device may encode and store data to memory, and/or a video decoding device may retrieve and decode data from memory. In many examples, the encoding and decoding is performed by devices that do not communicate with one another, but simply encode data to memory and/or retrieve and decode data from memory.

[0028] In the example of FIG. 1, source device 12 includes a video source 18, a video encoder 20, and an output interface 22. In some examples, output interface 22 may include a modulator/demodulator (modem) and/or a transmitter. Video source 18 may include a video capture device, e.g., a video camera, a video archive containing previously-captured video data, a video feed interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources of video data.

[0029] Video encoder 20 may encode video data from video source 18. In some examples, source device 12 directly transmits the encoded video data to destination device 14 via output interface 22. In other examples, the encoded video data may also be stored onto a storage medium or a file server for later access by destination device 14 for decoding and/or playback.

[0030] In the example of FIG. 1, destination device 14 includes an input interface 28, a video decoder 30, and a display device 32. In some examples, input interface 28 includes a receiver and/or a modem. Input interface 28 may receive encoded video data over channel 16. Video decoder 30 may decode encoded video data. Display device 32 may display the decoded video data. Display device 32 may be integrated with or may be external to destination device 14. Display device 32 may comprise 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.

[0031] Video encoder 20 and video decoder 30 each may be implemented as any of a variety of suitable circuitry, such as one or more microprocessors, digital signal processors (DSPs), application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), discrete logic, hardware, or any combinations thereof. If the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable storage medium and may execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Any of the foregoing (including hardware, software, a combination of hardware and software, etc.) may be considered to be one or more processors. Each of video encoder 20 and video decoder 30 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. An apparatus including video encoder 20 and/or video decoder 30 may comprise an integrated circuit, a microprocessor, and/or a wireless communication device, such as a cellular telephone.

[0032] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, and/or a video feed interface to receive video from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In some cases, if video source 18 is a video camera, source device 12 and destination device 14 may form so-called camera phones or video phones. As mentioned above, 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. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 20. The encoded video information may then be output by output interface 22 onto the channel 16.

[0033] Many of the techniques described in this disclosure can be performed by both video encoder 20 and video decoder 30. Therefore, for ease of explanation, the techniques may be described with respect to a video coder, which may be a video encoder and/or a video decoder, such as video encoder 20 and video decoder 30. This disclosure may generally refer to video encoder 20 "signaling" certain information to another device, such as video decoder 30. The term "signaling" may generally refer to the communication of syntax elements and/or other data used to decode the compressed video data. Such communication may occur in real- or near-real-time. Alternately, such communication may occur over a span of time, such as might occur when storing syntax elements to a computer-readable storage medium in an encoded bitstream at the time of encoding, which then may be retrieved by a decoding device at any time after being stored to this medium.

[0034] Video coding standards include ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual and ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC), including its Scalable Video Coding (SVC) and Multi-View Video Coding (MVC) extensions. In addition, High Efficiency Video Coding (HEVC) has recently been developed by the Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG). A draft of the HEVC specification is available from: http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip, hereinafter, "JCTVC-N1003."

[0035] In HEVC and other video coding specifications, a video sequence typically includes a series of pictures. Pictures may also be referred to as "frames." A picture may include three sample arrays, denoted S_L, S_{Cb}, and S_{Cr}. S_L is a two-dimensional array (i.e., a block) of luma samples. S_{Cb} is a two-dimensional array of Cb chrominance samples. S_{Cr} is a two-dimensional array of Cr chrominance samples. Chrominance samples may also be referred to herein as "chroma" samples. In other instances, a picture may be monochrome and may only include an array of luma samples.

[0036] To generate an encoded representation of a picture, video encoder 20 may generate a set of coding tree units (CTUs). Each of the CTUs may comprise a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples, and syntax structures used to code the samples of the coding tree blocks. In monochrome pictures or pictures having three separate color planes, a CTU may comprise a single coding tree block and syntax structures used to code the samples of the coding tree block. A coding tree block may be an NxN block of samples. A CTU may also be referred to as a "tree block" or a "largest coding unit" (LCU). The CTUs of HEVC may be broadly analogous to the macroblocks of other standards, such as H.264/AVC. However, a CTU is not necessarily limited to a particular size and may include one or more coding units (CUs). A slice may include an integer number of CTUs ordered consecutively in a raster scan order. For instance, a slice may be an integer number of CTUs contained in one independent slice segment and all subsequent dependent slice segments, if any, that precede the next independent slice segment, if any, within the same access unit.

[0037] To generate a coded CTU, video encoder 20 may recursively perform quad-tree partitioning on the coding tree blocks of a CTU to divide the coding tree blocks into coding blocks, hence the name "coding tree units." A coding block is an NxN block of samples. A CU may comprise a coding block of luma samples and two corresponding coding blocks of chroma samples of a picture that has a luma sample array, a Cb sample array, and a Cr sample array, and syntax structures used to code the samples of the coding blocks. In monochrome pictures or pictures having three separate color planes, a CU may comprise a single coding block and syntax structures used to code the samples of the coding block.

[0038] Video encoder 20 may partition a coding block of a CU into one or more prediction blocks. A prediction block is a rectangular (i.e., square or non-square) block of samples on which the same prediction is applied. A prediction unit (PU) of a CU may comprise a prediction block of luma samples, two corresponding prediction blocks of chroma samples, and syntax structures used to predict the prediction blocks. In monochrome pictures or pictures having three separate color planes, a PU may comprise a single prediction block and syntax structures used to predict the prediction block. Video encoder 20 may generate predictive luma, Cb, and Cr blocks for luma, Cb, and Cr prediction blocks of each PU of the CU.

[0039] Video encoder 20 may use intra prediction or inter prediction to generate the predictive blocks for a PU. If video encoder 20 uses intra prediction to generate the predictive blocks of a PU, video encoder 20 may generate the predictive blocks of the PU based on decoded samples of the picture associated with the PU.

[0040] If video encoder 20 uses inter prediction to generate the predictive blocks of a PU, video encoder 20 may generate the predictive blocks of the PU based on decoded samples of one or more pictures other than the picture associated with the PU. Inter prediction may be uni-directional inter prediction (i.e., uni-prediction) or bi-directional inter prediction (i.e., bi-prediction). To perform uni-prediction or bi-prediction, video encoder 20 may generate a first reference picture list (RefPicListO) and a second reference picture list (RefPicList1) for a current slice. Each of the reference picture lists may include one or more reference pictures. When using uni-prediction, video encoder 20 may search the reference pictures in either or both RefPicListO and RefPicList1 to determine a reference location within a reference picture. Furthermore, when using uni-prediction, video encoder 20 may generate, based at least in part on samples corresponding to the reference location, the predictive sample blocks for the PU. Moreover, when using uni-prediction, video encoder 20 may generate a single motion vector that indicates a spatial displacement between a prediction block of the PU and the reference location. To indicate the spatial displacement between a prediction block of the PU and the reference location, a motion vector may include a horizontal component specifying a horizontal displacement between the prediction block of the PU and the reference location and may include a vertical component specifying a vertical displacement between the prediction block of the PU and the reference location.

[0041] When using bi-prediction to encode a PU, video encoder 20 may determine a first reference location in a reference picture in RefPicListO and a second reference location in a reference picture in RefPicList1. Video encoder 20 may then generate, based at least in part on samples corresponding to the first and second reference locations, the predictive blocks for the PU. Moreover, when using bi-prediction to encode the PU, video encoder 20 may generate a first motion vector indicating a spatial displacement between a sample block of the PU and the first reference location and a second motion vector indicating a spatial displacement between the prediction block of the PU and the second reference location.

[0042] Typically, a reference picture list construction for the first or the second reference picture list (e.g., RefPicListO or RefPicList1) of a B picture includes two steps: reference picture list initialization and reference picture list reordering (modification). The reference picture list initialization is an explicit mechanism that puts the reference pictures in the reference picture memory (also known as decoded picture buffer) into a list based on the order of POC (Picture Order Count, aligned with display order of a picture) values. The reference picture list reordering mechanism can modify the position of a picture that was put in the list during the reference picture list initialization to any new position, or put any reference picture in the reference picture memory in any position even the picture doesn’t belong to the initialized list. Some pictures after the reference picture list reordering (modification) may be put in a very further position in the list. However, if a position of a picture exceeds the number of active reference pictures of the list, the picture is not considered as an entry of the final reference picture list. The number of active reference pictures may be signaled in the slice header for each list. After reference picture lists are constructed (namely RefPicListO and RefPicList1, if available), a reference index to a reference picture list can be used to identify any reference picture included in the reference picture list.

[0043] Video encoder 20 may encode certain blocks of video data using intra prediction mode encoding, and provide information indicating a selected intra prediction mode used to encode the block. Video encoder 20 may intra prediction encode blocks of any type of frame or slice (e.g., I-frames or I-slices, in addition to P-frames or P-slices and B-frames or B-slices) using an intra prediction mode. When video encoder 20 determines that a block should be intra prediction mode encoded, video encoder 20 may perform a rate-distortion analysis to select a most appropriate intra prediction mode. Intra prediction modes may also be referred to as "intra modes." For example, video encoder 20 may calculate rate-distortion values for one or more intra prediction modes, and select one of the modes having acceptable rate-distortion characteristics.

[0044] Video encoder 20 may, in some examples, be configured to begin analysis for selection of an intra prediction mode with the most probable mode, based on the context. When the most probable mode achieves suitable rate-distortion characteristics, in some examples, video encoder 20 may select the most probable mode. In other examples, video encoder 20 need not begin the selection process with the most probable mode.

[0045] After video encoder 20 generates predictive block (e.g., a predictive luma, Cb, and Cr block) for one or more PUs of a CU, video encoder 20 may generate a luma residual block for the CU. Each sample in a residual block may indicate a difference between a sample in a predictive block and a corresponding sample in an original coding block. Each sample in the luma residual block of the CU indicates a difference between a luma sample in one of the predictive luma blocks of the CU and a corresponding sample in the original luma coding block of the CU. In addition, video encoder 20 may generate a Cb residual block for the CU. Each sample in the CU's Cb residual block may indicate a difference between a Cb sample in one of the CU's predictive Cb blocks and a corresponding sample in the CU's original Cb coding block. Video encoder 20 may also generate a Cr residual block for the CU. Each sample in the CU's Cr residual block may indicate a difference between a Cr sample in one of the CU's predictive Cr blocks and a corresponding sample in the CU's original Cr coding block.

[0046] Furthermore, video encoder 20 may use quad-tree partitioning to decompose the luma, Cb, and Cr residual blocks of a CU into one or more luma, Cb, and Cr transform blocks. A transform block is a rectangular (e.g., square or non-square) block of samples on which the same transform is applied. A transform unit (TU) of a CU may comprise a transform block of luma samples, two corresponding transform blocks of chroma samples, and syntax structures used to transform the transform block samples. Thus, each TU of a CU may be associated with a luma transform block, a Cb transform block, and a Cr transform block. The luma transform block associated with the TU may be a sub-block of the CU's luma residual block. The Cb transform block may be a sub-block of the CU's Cb residual block. The Cr transform block may be a sub-block of the CU's Cr residual block. In monochrome pictures or pictures having three separate color planes, a TU may comprise a single transform block and syntax structures used to transform the samples of the transform block.

[0047] Video encoder 20 may apply one or more transforms to a luma transform block of a TU to generate a luma coefficient block for the TU. A coefficient block may be a two-dimensional array of transform coefficients. A transform coefficient may be a scalar quantity. Video encoder 20 may apply one or more transforms to a Cb transform block of a TU to generate a Cb coefficient block for the TU. Video encoder 20 may apply one or more transforms to a Cr transform block of a TU to generate a Cr coefficient block for the TU.

[0048] After generating a coefficient block (e.g., a luma coefficient block, a Cb coefficient block or a Cr coefficient block), video encoder 20 may quantize the coefficient block. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the transform coefficients, providing further compression.

[0049] Thus, following intra-predictive or inter-predictive coding to produce predictive data and residual data, and following any transforms (such as the 4x4 or 8x8 integer transform used in H.264/AVC or a discrete cosine transform DCT) to produce transform coefficients, quantization of transform coefficients may be performed. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the coefficients. The quantization process may reduce the bit depth associated with some or all of the coefficients. For example, an n-bit value may be rounded down to an m-bit value during quantization, where n is greater than *m*.

[0050] After video encoder 20 quantizes a coefficient block, video encoder 20 may entropy encode syntax elements indicating the quantized transform coefficients. For example, video encoder 20 may perform Context-Adaptive Binary Arithmetic Coding (CABAC) on the syntax elements indicating the quantized transform coefficients. For instance, following quantization, entropy coding of the quantized data may be performed, e.g., according to content adaptive variable length coding (CAVLC), CABAC, or another entropy coding methodology. A processing unit configured for entropy coding, or another processing unit, may perform other processing functions, such as zero run length coding of quantized coefficients and/or generation of syntax information such as coded block pattern (CBP) values, macroblock type, coding mode, maximum macroblock size for a coded unit (such as a frame, slice, macroblock, or sequence), or the like.

[0051] Video encoder 20 may also be configured to determine an encoding context for a block. The context may be determined based on various characteristics of the block such as, for example, a size of the block, which may be determined in terms of pixel dimensions, prediction unit (PU) type such as, in the example of HEVC, 2Nx2N, Nx2N, 2NxN, NxN, short-distance intra prediction (SDIP) types such as 2NxN/2, N/2x2N, 2Nx1, 1x2N, a macroblock type in the example of H.264, a coding unit (CU) depth for the block, or other measurements of size for a block of video data.

In some examples, video encoder 20 may determine the context based on how any or all of intra prediction modes for an above-neighboring block, a left-neighboring block, an above-left neighboring block, an above-right neighboring block, or other neighboring blocks. In some examples, video encoder 20 determines the context based on both intra prediction modes for one or more blocks as well as size information for the current block being encoded.

[0052] Video encoder 20 may output a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data. The bitstream may comprise a sequence of network abstraction layer (NAL) units. A NAL unit is a syntax structure containing an indication of the type of data in the NAL unit and bytes containing that data in the form of a raw byte sequence payload (RBSP) interspersed as necessary with emulation prevention bits. Each of the NAL units includes a NAL unit header and encapsulates a RBSP. The NAL unit header may include a syntax element that indicates a NAL unit type code. The NAL unit type code specified by the NAL unit header of a NAL unit indicates the type of the NAL unit. A RBSP may be a syntax structure containing an integer number of bytes that is encapsulated within a NAL unit. In some instances, an RBSP includes zero bits.

[0053] Different types of NAL units may encapsulate different types of RBSPs. For example, a first type of NAL unit may encapsulate an RBSP for a picture parameter set (PPS), a second type of NAL unit may encapsulate an RBSP for a coded slice, a third type of NAL unit may encapsulate an RBSP for SEI messages, and so on. NAL units that encapsulate RBSPs for video coding data (as opposed to RBSPs for parameter sets and SEI messages) may be referred to as video coding layer (VCL) NAL units. NAL units that contain parameter sets (e.g., video parameter sets (VPSs), sequence parameter sets (SPSs), picture parameter sets (PPSs), etc.) may be referred to as parameter set NAL units.

[0054] Video decoder 30 may receive a bitstream generated by video encoder 20. In addition, video decoder 30 may parse the bitstream to obtain syntax elements from the bitstream. Video decoder 30 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 20. For instance, video decoder 30 may use motion vectors of PUs to determine predictive blocks for the PUs of a current CU. In addition, video decoder 30 may inverse quantize coefficient blocks associated with TUs of the current CU. Video decoder 30 may perform inverse transforms on the coefficient blocks to reconstruct transform blocks associated with the TUs of the current CU. Video decoder 30 may reconstruct the coding blocks of the current CU by adding the samples of the predictive blocks for PUs of the current CU to corresponding samples of the transform blocks of the TUs of the current CU. By reconstructing the coding blocks for each CU of a picture, video decoder 30 may reconstruct the picture.

[0055] Thus, in some examples of this disclosure, video decoder 30 may ultimately receive encoded video data, e.g., from input interface 28. In accordance with some techniques of this disclosure, video decoder 30 may receive a codeword or other syntax representative of an intra prediction mode used to encode a block of video data. Video decoder 30 may be configured to determine a coding context for the block in a manner substantially similar to video encoder 20.

[0056] As mentioned above, particular NAL units of the bitstream may include VPSs, SPS, and PPSs. In some examples, a VPS is a syntax structure comprising syntax elements that apply to zero or more entire coded video sequences (CVSs). In some examples, an SPS is a syntax structure containing syntax elements that apply to zero or more entire CVSs. An SPS may include a syntax element that identifies a VPS that is active when the SPS is active. Thus, the syntax elements of a VPS may be more generally applicable than the syntax elements of a SPS. In some examples, a PPS is a syntax structure containing syntax elements that apply to zero or more entire coded pictures as determined by a syntax element found in each slice segment header.

[0057] A parameter set (e.g., a VPS, SPS, PPS, etc.) may contain an identification that is referenced, directly or indirectly, from a slice header of a slice. In some examples, a slice header is the slice segment header of the independent slice segment that is a current slice segment or the most recent independent slice segment that precedes a current dependent slice segment in decoding order. In such examples, a slice segment header is an integer number of coding tree units ordered consecutively in the tile scan and contained in a single NAL unit. The referencing process is known as "activation." Thus, when video decoder 30 is decoding a particular slice, a parameter set referenced, directly or indirectly, by a syntax element in a slice header of the particular slice is said to be "activated." Depending on the parameter set type, the activation may occur on a per picture basis or a per sequence basis. For example, a slice header of a slice may include a syntax element that identifies a PPS. Thus, when a video coder codes the slice, the PPS may be activated. Furthermore, the PPS may include a syntax element that identifies a SPS. Thus, when a PPS that identifies the SPS is activated, the SPS may be activated. The SPS may include a syntax element that identifies a VPS. Thus, when a SPS that identifies the VPS is activated, the VPS is activated.

[0058] As mentioned briefly above, video encoder 20 may encode syntax elements using CABAC encoding. To apply CABAC encoding to a syntax element, video encoder 20 may binarize the syntax element to form a series of one or more bits, which are referred to as "bins." In addition, video encoder 20 may identify a coding context. The coding context may identify probabilities of coding bins having particular values. For instance, a coding context may indicate a 0.7 probability of coding a 0-valued bin and a 0.3 probability of coding a 1-valued bin. After identifying the coding context, video encoder 20 may divide an interval into a lower sub-interval and an upper sub-interval. One of the sub-intervals may be associated with the value 0 and the other sub-interval may be associated with the value 1. The widths of the sub-intervals may be proportional to the probabilities indicated for the associated values by the identified coding context. If a bin of the syntax element has the value associated with the lower sub-interval, the encoded value may be equal to the lower boundary of the lower sub-interval. If the same bin of the syntax element has the value associated with the upper sub-interval, the encoded value may be equal to the lower boundary of the upper sub-interval. To encode the next bin of the syntax element, video encoder 20 may repeat these steps with the interval being the sub-interval associated with the value of the encoded bit. When video encoder 20 repeats these steps for the next bin, video encoder 20 may use modified probabilities based on the probabilities indicated by the identified coding context and the actual values of bins encoded. In some examples, when video encoder 20 repeats these steps for the next bin, video encoder 20 may select another coding context.

[0059] When video decoder 30 performs CABAC decoding on a syntax element, video decoder 30 may identify a coding context. Video decoder 30 may then divide an interval into a lower sub-interval and an upper sub-interval. One of the sub-intervals may be associated with the value 0 and the other sub-interval may be associated with the value 1. The widths of the sub-intervals may be proportional to the probabilities indicated for the associated values by the identified coding context. If the encoded value is within the lower sub-interval, video decoder 30 may decode a bin having the value associated with the lower sub-interval. If the encoded value is within the upper sub-interval, video decoder 30 may decode a bin having the value associated with the upper sub-interval. To decode a next bin of the syntax element, video decoder 30 may repeat these steps with the interval being the sub-interval that contains the encoded value. When video decoder 30 repeats these steps for the next bin, video decoder 30 may use modified probabilities based on the probabilities indicated by the identified coding context and the decoded bins. Video decoder 30 may then de-binarize the bins to recover the syntax element.

[0060] Rather than performing regular CABAC encoding on all syntax elements, video encoder 20 may encode some bins using bypass CABAC coding. It may be computationally less expensive to perform bypass CABAC coding on a bin than to perform regular CABAC coding on the bin. Furthermore, performing bypass CABAC coding may allow for a higher degree of parallelization and throughput. Bins encoded using bypass CABAC coding may be referred to as "bypass bins." Grouping bypass bins together may increase the throughput of video encoder 20 and video decoder 30. The bypass CABAC coding engine may be able to code several bins in a single cycle, whereas the regular CABAC coding engine may be able to code only a single bin in a cycle. The bypass CABAC coding engine may be simpler because the bypass CABAC coding engine does not select contexts and may assume a probability of ½ for both symbols (0 and 1). Consequently, in bypass CABAC coding, the intervals are split directly in half.

[0061] FIG. 2 is a conceptual diagram illustrating an example of intra prediction of a block of video data. For intra prediction, a block of video data (e.g., a PU) is predicted using reconstructed image samples that are spatially neighboring. A typical example of the intra prediction for an image block 40, e.g., a 16x16 image block, is shown in FIG. 2. With intra prediction, image block 40 is predicted by the above and left neighboring reconstructed samples (reference samples) along a selected prediction direction (as indicated by arrow 42).

[0062] FIG. 3 is a conceptual diagram illustrating an example of intra prediction modes and corresponding mode indexes. In HEVC, there are 35 modes for the intra prediction of a luma block, including a planar mode (i.e., a planar intra prediction mode), a DC mode and 33 angular modes (i.e., angular intra prediction modes), as indicated in FIG 3. The 35 modes of the intra prediction, as defined in HEVC, are indexed as below table

Table 1 - Specification of intra prediction mode and associated names

Intra prediction mode	Associated name
0	INTRA_PLANAR
1	INTRA_DC
2..34	INTRA_ANGULAR2..INTRA_ANGULAR34

[0063] FIG. 4 is a conceptual diagram illustrating an example technique for generating a prediction sample for a block of video data according to a planar intra prediction mode. Planar mode is typically the most frequently used Intra prediction mode. To perform a planar prediction for an NxN block, for each sample p_{xy} located at (x, y), as illustrated in FIG. 4, the prediction value is calculated using four specific neighboring reconstructed samples, i.e., reference samples, with a bilinear filter. The four reference samples include a top-right reconstructed sample TR (50), a bottom-left reconstructed sample BL (52), and the two reconstructed samples (54, 56) located at the same column ($r_{x,-1}$) and row ($r_{-1,y}$) of the current sample, as illustrated in FIG. 4. The planar mode can be formulated as: $p_{xy} = (N\text{-}x\text{-}1)\cdot L + (N\text{-}y\text{-}1)\cdot T + x\cdot\mathrm{TR} + y\cdot\mathrm{BL}$. In this formula, *N* is the height and width of the block.

[0064] For a DC mode, the prediction block is simply filled with the average value of the neighboring reconstructed samples. For instance, to generate a predictive block for a PU using a DC intra prediction mode, the video coder may set each sample of the predictive block for the PU equal to an average value of neighboring reconstructed samples. Generally, both the planar and DC intra prediction modes are applied for modeling smoothly varying and constant image regions.

[0065] HEVC specifies 33 different prediction directions for its angular intra prediction modes. For each given angular intra prediction, the intra prediction direction can be identified, for example, according to FIG. 3. In the example of FIG. 3, intra prediction mode 10 corresponds to a pure horizontal prediction direction, and intra prediction mode 26 corresponds to a pure vertical prediction direction.

[0066] Given a specific intra prediction direction, for each sample of the prediction block, its coordinate (x, y) is first projected to the row/column of neighboring reconstructed samples along the prediction direction, as shown in an example in FIG. 3. Suppose, (x, y) is projected to the fractional position α between two neighboring reconstructed samples L and R, then the prediction value for (x, y) is calculated using a two-tap bi-linear interpolation filter, formulated as follows: $p_{xy} = (1- \alpha)\cdot L + \alpha\cdot R$. To avoid floating point operations, in HEVC, the preceding calculation is actually approximated using integer arithmetic as: $p_{xy} = ((32-a)\cdot L + a\cdot R + 16)>>5$, where a is an integer equal to $32*\alpha$.

[0067] For intra mode coding in HEVC, for each intra PU, a selected intra prediction mode is signaled. To signal the selected intra prediction mode, three intra prediction modes are first identified which are assumed to have higher probability to be actually selected, namely the Most Probable Modes (MPM). In HEVC, the MPMs, labeled candModeList[x], x=0, 1, 2, are derived as follows.

[0068] Firstly, a left neighboring mode (candIntraPredModeA) and an above neighboring mode (candIntraPredModeB) are derived as follows according to subclause 8.4.2 of JCTVC-N1003:

Input to this process is a luma location (xPb, yPb) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture.

1. The neighbouring locations (xNbA, yNbA) and (xNbB, yNbB) are set equal to (xPb - 1, yPb) and (xPb, yPb - 1), respectively.

2. For X being replaced by either A or B, the variables candIntraPredModeX are derived as follows:

- The availability derivation process for a block in z-scan order as specified in subclause 6.4.1 is invoked with the location (xCurr, yCurr) set equal to (xPb, yPb) and the neighbouring location (xNbY, yNbY) set equal to (xNbX, yNbX) as inputs, and the output is assigned to availableX.
- The candidate intra prediction mode candIntraPredModeX is derived as follows:
  - If availableX is equal to FALSE, candIntraPredModeX is set equal to INTRA_DC.
  - Otherwise, if CuPredMode[xNbX][yNbX] is not equal to MODE_INTRA or pcm_flag[xNbX][yNbX] is equal to 1, candIntraPredModeX is set equal to INTRA_DC,
  - Otherwise, if X is equal to B and yPb - 1 is less than ((yPb >> CtbLog2SizeY) « CtbLog2SizeY), candIntraPredModeB is set equal to INTRA_DC.
  - Otherwise, candIntraPredModeX is set equal to IntraPredModeY[xNbX][yNbX].

[0069] Thus, if the neighboring sample left of the top left sample of the luma prediction block of a current PU is not available, if the neighboring sample left of the top left sample of the luma prediction block of the current PU is not predicted using intra prediction, or if the neighboring sample left of the top left sample of the luma prediction block of the current PU is encoded using pulse code modulation (PCM), candIntraPredModeA is set to the DC intra prediction mode. Otherwise, candIntraPredModeA is set to the intra prediction mode of a PU whose prediction block contains the neighboring sample left of the top left sample of the luma prediction block of the current PU. Similarly, if the neighboring sample above the top left sample of the luma prediction block of a PU is not available, if the neighboring sample above the top left sample of the luma prediction block of the current PU is not predicted using intra prediction, if the neighboring sample above the top left sample of the luma prediction block of the current PU is encoded using PCM, or if the neighboring sample above the top left sample of the luma prediction block of the current PU is in a different coding tree block from the current PU, candIntraPredModeB is set to the DC intra prediction mode. Otherwise, candIntraPredModeB is set to the luma intra prediction mode of a PU whose prediction block contains the neighboring sample above the top left sample of the luma prediction block of the current PU (i.e., IntraPredModeY[xNbX][yNbX].

[0070] Then, in subclause 8.4.2 of JCTVC-N1003, using the derived left neighboring mode (candIntraPredModeA) and above neighboring mode (candIntraPredModeB), the three MPMs are derived as follows:

3. The candModeList[x] with x = 0..2 is derived as follows:

- If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
    - If candIntraPredModeA is less than 2 (i.e. equal to INTRA_PLANAR or INTRA_DC), candModeList[x] with x = 0..2 is derived as follows:

$$\text{candModeList[0]} = \text{INTRA\_PLANAR} \qquad (8\text{-}15)$$

$$\text{candModeList[1]} = \text{INTRA\_DC} \qquad (8\text{-}16)$$

$$\text{candModeList[2]} = \text{INTRA\_ANGULAR26} \qquad (8\text{-}17)$$

    - Otherwise, candModeList[x] with x = 0..2 is derived as follows:

$$\text{candModeList[0]} = \text{candIntraPredModeA} \qquad (8\text{-}18)$$

$$\text{candModeList[1]} = 2 + ((\text{candIntraPredModeA} + 29) \,\%\, 32) \qquad (8\text{-}19)$$

$$\text{candModeList[2]} = 2 + ((\text{candIntraPredModeA} - 2 + 1) \,\%\, 32) \qquad (8\text{-}20)$$

- Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the following applies:
    - candModeList[0] and candModeList[1] are derived as follows:

$$\text{candModeList[0]} = \text{candIntraPredModeA} \qquad (8\text{-}21)$$

$$\text{candModeList[1]} = \text{candIntraPredModeB} \qquad (8\text{-}22)$$

    - If neither of candModeList[0] and candModeList[1] is equal to INTRA_PLANAR, candModeList[2] is set equal to INTRA_PLANAR,
    - Otherwise, if neither of candModeList[0] and candModeList[1] is equal to INTRA_DC, candModeList[2] is set equal to INTRA_DC,
    - Otherwise, candModeList[2] is set equal to INTRA_ANGULAR26.

[0071] After the three MPMs are decided, in subclause 8.4.2 of JCTVC-N1003, for each PU, a one-bit flag prev_intra_luma_pred_flag[xPb][yPb] is first signaled to indicate whether the selected intra mode of the current PU is same as one of the three MPMs.

- If prev_intra_luma_pred_flag[xPb][yPb] being signaled as 1, i.e., one of the 3 MPMs is selected for coding the current PU, a index mpm_idx (may equal to 0, 1 or 2), indicating which MPM is selected for coding the current PU, is further signaled. The mpm_idx is binarized using truncated unary code and by-pass coded using no context modeling
- Otherwise (prev intra_luma_pred_flag[xPb][yPb] being signaled as 0), i.e., non-MPM is used for current PU, an index rem_intra_luma_pred_mode[xPb][yPb], indicating which non-MPM is selected for the current PU, is further

signaled. The value of rem_intra_luma_pred_mode[xPb][yPb] could be 0, 1, ...,31, and fixed length (5 bits) binarization is used with bypass coding.

[0072] As previously discussed, video encoder 20 selects the intra prediction mode for a current PU. The process of selecting the intra prediction mode for a PU may be referred to as an encoder-side intra mode decision. When selecting an intra prediction mode for a current PU, video encoder 20 may select one of the three MPMs determined for the current PU or one of the non-MPM intra prediction modes. If video encoder 20 selects one of the three MPMs for the current PU as the intra prediction mode for the current PU, video encoder 20 may signal, in the bitstream, an MPM index for the current PU. In JCTVC-N1003, the MPM index for a PU having a top left luma sample at coordinates (x, y) is denoted mpm_ind[x][y]. Otherwise, if video encoder 20 does not select one of the three MPMs for the current PU as the intra prediction mode for the current PU, video encoder 20 may signal, in the bitstream, a non-MPM index for the current PU. In JCTVC-N1003, the non-MPM index for a PU having a top left luma sample at coordinates (x, y) is denoted rem_intra_pred_mode[x][y]. Furthermore, video encoder 20 signals, in the bitstream, a syntax element indicating whether an MPM index or a non-MPM index is signaled for the current PU. In JCTVC-N1003, this syntax element for a PU having a top left luma sample at coordinates (x, y) is denoted prev_intra_luma_pred_flag[x][y].

[0073] In JCTVC-N1003, an MPM index may only have three potential different values, while a non-MPM index may have many more potential values. Consequently, fewer bits may be required to signal an MPM index than a non-MPM index.

[0074] To select an intra prediction mode for a current PU as efficiently as possible, in the design of HEVC reference software (referred to as "the HM"), several fast intra mode decision methods have been integrated. The HM reference software may be downloaded from: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-14.0/.

[0075] Since there are as many as 35 available intra prediction modes for a luma block, a full rate-distortion optimization based encoder mode decision may be too expensive for a practical implementation. In the latest version of the HM, the intra mode decision is performed as two stages. In the first stage, *N* Intra mode candidates list is first roughly decided using a much cheaper cost criterion commonly known as "Sum of Absolute Transform Difference" (SATD). The value of *N* depends on block size, and the default setting in the HM is: *N* equals to 8 for 4x4 and 8x8, and *N* equals to 3 for 16x16 and larger block sizes. After that, the left and above neighboring modes, i.e., either candIntraPredModeA or both candIntraPredModeA and candIntraPredModeB (if candIntraPredModeA does not equal to candIntraPredModeA), are appended to the intra mode candidate list, if not already included. In the second stage, the intra mode candidate list is fed into the expensive rate-distortion cost calculation process, and the final best intra prediction mode is decided for the current PU. With this two stage intra mode decision process, the majority of intra prediction modes are skipped for the expensive rate-distortion cost calculation, and the best intra prediction mode is still selected without much penalty of coding performance drop.

[0076] In Matsuo et al., "Improved intra angular prediction by DCT-based interpolation filter," Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, pp. 1568-1572. IEEE, 2012., it is proposed to apply a 4-tap DCT based interpolation filter for 4x4 and 8x8 block sizes and the Intra smoothing filter is also turned off when 4-tap filter is applied, for block sizes larger than or equal to 16x16, the 2-tap bilinear interpolation filter is applied. In Maani, Ehsan, "Interpolation filter for intra prediction of HEVC," U.S. Patent Application 13/312,946, filed December 6, 2011, a 4-tap interpolation filter can be used when the intra smoothing filter is off, while the 4-tap interpolation filter could be obtained based on a CUBIC interpolation process, a DCT-based interpolation process or a Hermite interpolation process. In M. Guo, X. Guo, and S. Lei, "Improved Intra Mode Coding", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th Meeting: Daegu, Korea, 20-28 January, 2011, the binarization of the intra mode is adaptively selected from a set of pre-defined coding trees according to the modes of its neighboring blocks.

[0077] There may be a number of problems associated with existing techniques for intra prediction for coding video data, e.g., according to the HEVC standard. For example, the 35 fixed prediction angles of the existing design of intra prediction in HEVC may be inefficient to capture very flexible edge direction distributions. However, applying more prediction angles may result in undesired encoder complexity increases. For example, a direct extension to 65 prediction angles based on the HEVC reference software may require an approximately doubled number of SATD checks relative to the 35 fixed prediction angles of the existing design of intra prediction in HEVC. Additionally, the efficiency of the current intra mode coding, e.g., using three MPMs in HEVC, can be limited because it may not accurately correspond to the actual probability distribution of all available intra prediction modes. In other words, the three MPMs determined in HEVC for a particular PU may not actually be the intra prediction modes that are most likely to be used for the particular PU.

[0078] The techniques of this disclosure may resolve the above-identified problems associated with existing techniques for intra prediction for coding video data, e.g., according to the HEVC standard, as well as other problems in the technical fields of intra prediction and video coding and compression. The following itemized methods may be applied individually. Alternatively, any combination of them may be applied.

[0079] In accordance with some techniques of this disclosure, more than three MPMs can be used for signaling an intra prediction mode. For instance, a typical example of the number of MPM, denoted by *M*, can be 4, 5 or 6. The use of additional MPMs may reduce the probability of using non-MPM indexes to signal intra prediction modes of PUs. Because MPM indexes are typically represented using fewer bits than non-MPM indexes, the use of additional MPMs may increase decrease bitstream size.

[0080] In some examples where more than three MPMs are used for signaling an intra prediction mode of a PU, video encoder 20 may use an entropy encoding process, such as CABAC, to encode a MPM index and video decoder 30 may use an entropy decoding process, such as CABAC, to decode the MPM index. The MPM index for a PU may be a numerical value identifying which of the MPMs is the intra prediction mode of the PU. As part of using the entropy encoding process to encode the MPM index, video encoder 20 may binarize the MPM index. In other words, video encoder 20 may convert the MPM index into a binary code. In different examples, video encoder 20 may use various binarization methods to convert the MPM index into a binary code. For instance, a video coder may binarize the MPM index using at least one of: a fixed truncated unary code or a fixed Huffman code. Furthermore, in some examples, when coding the MPM index, the binarization method can be a fixed truncated unary code, or a fixed Huffman code or decided based on the coded neighboring intra modes. For instance, a video coder may determine the binarization method based on intra prediction modes used to code one or more neighboring blocks. For instance, for an MPM index of a PU, the video coder may determine the binarization method based on intra prediction modes used to code one or more neighboring blocks adjacent to the PU. After binarizing the MPM index, video encoder 20 may CABAC encode the binarized MPM index and include the CABAC-encoded binarized MPM index in the bitstream. As part of decoding the MPM index, video decoder 30 may obtain the CABAC-encoded binarized MPM index from the bitstream, apply CABAC decoding to the CABAC-encoded binarized MPM index to recover the binarized MPM index, and de-binarize the MPM index to recover the MPM index. In this disclosure, the phrase "bins of an MPM index" may refer to bins of the binarized MPM index.

[0081] In some examples of this disclosure, video encoder 20 and video decoder 30 may use context modeling when coding (e.g., CABAC coding) one or more bins of an MPM index. In other words, video encoder 20 and video decoder 30 may select, for each respective bin of one or more bins of the MPM index, a respective coding context for the respective bin. Video encoder 20 and video decoder 30 may use the respective coding context for the respective bin to code the respective bin. For example, when coding the MPM index, the first or selected N coded bins can be coded using context modeling, while the context index is decided using the coded neighboring intra prediction modes. For instance, for an MPM index of a PU in a current picture, the video coder may determine the context index based on intra prediction modes used to code one or more neighboring PUs adjacent in the current picture to the PU. In this example, *N* is an integer. In some various examples, *N* may be equal to 1, 2, or 3. Thus, in some examples, the one or more context-modeled bins consists of the first bin of the MPM index. In other examples, the one or more the one or more context-modeled bins consists of a selected *N* bins of the MPM index, where *N* is set equal to *M* - *K,* wherein *K* is a pre-defined number.

[0082] In one example where video encoder 20 and video decoder 30 use context modeling when coding one or more bins of an MPM index of a current PU, the context index is dependent on the cases of left and above neighboring intra prediction modes. In case 0 of this example, left and above neighboring intra prediction modes are same and neither of the intra prediction modes is DC or planar. In case 1 of this example, the left and the above neighboring intra prediction modes are same and at least one of the intra prediction modes is DC or planar. In case 2 of this example, the left and the above neighboring intra prediction modes are different and neither is planar. In case 3 of this example, left and above neighboring intra prediction modes are different and at least one of the intra prediction modes is planar. The left and above neighboring intra prediction modes may be the intra prediction modes of PUs left and above the current PU. In this example, video encoder 20 and video decoder 30 may calculate the context index based on a pre-defined mapping table which maps from the case index (e.g., case 0, case 1, case 2, or case 3) to a context index number. In other words, a video coder may use a case index to context index mapping table to determine a context index from the case index. The context index number may identify a predefined coding context in a set of predefined coding contexts.

[0083] Thus, in this example, when a video coder selects a context index for a context-modeled bin of the MPM index, the video coder may select a first context index if intra prediction modes used to decode a left neighboring block and an above neighboring block are the same and neither of the intra prediction modes used to decode the left neighboring block nor the above neighboring block is DC or planar. Furthermore, in this example, the video coder may select a second context index if intra prediction modes used to decode the left neighboring block and the above neighboring block are same and at least one of the intra prediction modes used to decode the left neighboring block and the above neighboring block is DC or planar. In this example, the video coder may select a third context index if intra prediction modes used to decode the left neighboring block and the above neighboring block are different and neither of the intra prediction modes used to decode the left neighboring block nor the above neighboring block is planar. In this example, the video coder may select a fourth context index if intra prediction modes used to decode the left neighboring block and the above neighboring block are different and at least one of the intra prediction modes used to decode the left neighboring block and the above neighboring block is DC or planar.

[0084] In some examples, different case index to context index mapping tables are used for different bins. For example, a video coder, such as video encoder 20 or video decoder 30, may use a first case index to context index mapping table for bin 0 of the MPM index, a second case index to context index mapping table for bin 1 of the MPM index, a third case index to context index mapping table for bin 2 of the MPM index, and so on. The different case index to context index mapping tables of this example may assign different contexts to the cases (e.g., case 0, case 1, case 2, and case 3) described above. For instance, the mapping described in the previous paragraph may apply for the first bin of the MPM index, but for a second bin of the MPM index, the video coder may select the second context index if intra prediction modes used to decode a left neighboring block and an above neighboring block are the same and neither of the intra prediction modes used to decode the left neighboring block nor the above neighboring block is DC or planar; select the third context index if intra prediction modes used to decode the left neighboring block and the above neighboring block are same and at least one of the intra prediction modes used to decode the left neighboring block and the above neighboring block is DC or planar; and so on.

[0085] In some examples where video encoder 20 and video decoder 30 use context modeling when coding one or more bins of an MPM index of a current PU, only one context is applied for coding each of the first or selected N bins, and the context may not depend on the neighboring intra prediction modes. For instance, in such examples, video encoder 20 and video decoder 30 perform the coding context selection process once for the MPM index and do not repeat for the coding context selection process for subsequent bins of the MPM index. In other words, a video coder may select a common context index for each of the one or more context-modeled bins of the MPM index.

[0086] As mentioned above, in some examples, video encoder 20 and video decoder 30 may use context modeling when coding *N* coded bins of an MPM index. In one such example, *N* is set equal to (*M-K*), where *K* is a pre-defined number and *M* is the number of MPMs. For example, *K* may be equal to 1, 2 or 3.

[0087] Thus, in the examples above, video encoder 20 may derive *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In this example, *M* is greater than 3. Furthermore, encoding a syntax element that indicates whether a most probable mode index or a non-most probable mode index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode the non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Based on the indicated one of the MPM index or the non-MPM index being the MPM index, video encoder 20 may select, for each of one or more context-modeled bins of the MPM index, based on intra prediction modes used to decode one or more neighboring blocks, a context index for the context-modeled bin. Video encoder 20 may encode the block of video data based on the selected intra prediction mode.

[0088] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, to encode the syntax element that indicates whether a MPM index or a non-MPM index is used to indicate the selected intra prediction mode, video encoder 20 may encode a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the PU. Video encoder 20 may for each of the one or more context-modeled bins of the MPM index, use the selected context index for the context-modeled bin to perform CABAC encoding of the context-modeled bin. Furthermore, as part of encoding the block of video data based on the selected intra prediction mode, video encoder 20 may generate, based on the selected intra prediction mode of the current PU, a predictive block for the current PU. Video encoder 20 may generate residual data that represents pixel differences between the current CU and the predictive block. In this example, video encoder 20 may output a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data, the coded pictures including the current picture.

[0089] Similarly, in some examples, video decoder 30 may derive *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In this example, *M* is greater than 3. Furthermore, video decoder 30 may decode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode and the non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Based on the indicated one of the MPM index or the non-MPM index being the MPM index, video decoder 30 may select, for each of one or more context-modeled bins of the MPM index, based on intra prediction modes used to decode one or more neighboring blocks, a context index for the context-modeled bin. Furthermore, video decoder 30 may reconstruct the block of video data based on the selected intra prediction mode.

[0090] For instance, in this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, to decoded the syntax element that indicates whether the MPM index or the non-MPM index is used to indicate the selected intra prediction mode, video decoder 30 may decode, from a bitstream, a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the current PU. In this example, for each of the one or more context-modeled bins of the MPM index, video decoder 30 may use the selected context index for the context-modeled bin to perform CABAC decoding of the context-modeled bin of the MPM index. Furthermore, as part of reconstructing the block of video data based on the selected intra prediction mode, video decoder 30 may generate a predictive block for the current PU using the selected intra prediction mode. In this example, video decoder 30 may reconstruct the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU. In this example, video decoder 30 may receive a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data, the coded pictures including the current picture.

[0091] As briefly discussed above, the efficiency of intra prediction mode coding using the three MPMs specified in HEVC may be limited because the three MPMs specified in HEVC frequently do not accurately correspond to the actual probability distribution of all available intra prediction modes. For instance, the probability of the intra prediction mode of a PU being a particular non-MPM intra prediction mode may be greater than the probabilities of the intra prediction mode of the PU being any of the three MPMs specified in HEVC.

[0092] Thus, in accordance with some examples of this disclosure, when deriving the MPMs, a representative intra prediction mode is defined for left (or above) neighboring column (or row) and the representative intra prediction mode is used as the MPM candidate from the left (or above) neighboring column. For instance, a video coder may define a representative intra prediction mode for a left neighboring column and use the representative mode for the left neighboring column as an MPM candidate. The video coder may use the representative mode for the left neighboring column as an MPM candidate in addition to the MPM defined in HEVC for the left neighboring block or in place of the MPM defined in HEVC for the left neighboring block (candIntraPredModeA). Moreover, the video coder may define a representative intra prediction mode for an above neighboring row and use the representative mode for the above neighboring row as an MPM candidate. The video coder may use the representative mode for the above neighboring row as an MPM candidate in addition to the MPM defined in HEVC for the above neighboring block or in place of the MPM defined in HEVC for the above neighboring block (candIntraPredModeB).

[0093] The left neighboring column may include a column of samples immediately left in a picture of a column of samples containing a current sample of a current PU for a CU of the picture. For instance, the left neighboring column may include (or consist of) each sample immediately left of the current PU. The above neighboring row may include a row of samples immediately above in a picture of a row of samples containing a current sample of a current PU for a CU of the picture. For instance, the above neighboring row may include (or consist of) each sample immediately above the current PU.

[0094] For example, as part of encoding video data, video encoder 20 may encode a block of video data in part by deriving *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In some examples, *M* is greater than 3. In this example, the MPMs may include an MPM for the left neighboring column and an MPM for the above neighboring row. In some examples, video encoder 20 defines a representative intra prediction mode for the left neighboring column and uses the representative intra prediction mode for the left neighboring column as the MPM for the left neighboring column. In some examples, video encoder 20 defines a representative intra prediction mode for the above neighboring row and using the representative intra prediction mode for the above neighboring row as the MPM for the above neighboring row.

[0095] Furthermore, video encoder 20 may encode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data, and encodes the indicated one of the MPM index or the non-MPM index. For example, video encoder 20 may include, in the bitstream, data indicating a prev_intra_luma_pred_flag syntax element that indicates whether the bitstream includes data indicating an mpm_index or a rem_intra_luma_pred_mode syntax element. The MPM index indicates which of the MMPMs is the selected intra prediction mode. The non-MPM index indicates which of the intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Video encoder 20 encodes the block of video data based on the selected intra prediction mode.

[0096] In this example, the block of video data is a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. To encode the block of video data based on the selected in intra prediction mode, video encoder 20 may reconstruct the block of video data in part by generating a predictive block for the current PU using the selected intra prediction mode. Furthermore, video encoder 20 may reconstruct the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU.

[0097] In a similar example of selecting an intra prediction mode to be used for intra prediction for coding video data, video decoder 30 may decode a block of video data in part by deriving *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In this example, *M* may be greater than 3. The MPMs may include an MPM for the left neighboring column and an MPM for the above neighboring row. In some examples, video decoder

30 defines a representative intra prediction mode for the left neighboring column and uses the representative intra prediction mode for the left neighboring column as the MPM for the left neighboring column. In some examples, video decoder 30 defines a representative intra prediction mode for the above neighboring row and uses the representative intra prediction mode for the above neighboring row as the MPM for the above neighboring row.

[0098] Furthermore, video decoder 30 may decode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data, and decodes the indicated one of the most probable mode index or the non-most probable mode index. For instance, video decoder 30 may obtain a previ_ntrapred_flag from a bitstream. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode.

[0099] Video decoder 30 may then reconstruct a video block based on the resulting selected intra prediction mode. In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, as part of reconstructing the video block based on the resulting selected intra prediction mode, video decoder 30 may use the selected intra prediction mode to generate a predictive block for the video block. For instance, the video block may be a CU and video decoder 30 may use the selected intra prediction mode to generate a predictive block for a PU of the CU. Furthermore, video decoder 30 may reconstruct at least some samples of the video block by adding samples of the predictive block to corresponding residual samples for the video block. For instance, the video block may be a CU and video decoder 30 may add samples of the predictive block to corresponding residual samples in one or more transform blocks of TUs of the CU.

[0100] In various examples, a video coder may determine a representative intra prediction mode in different ways. In one example, it is defined as a function of all the intra prediction modes used by the neighboring blocks. For example, the number of actually used intra prediction modes of the left (or above) neighboring column (or row), in unit of the smallest PU size (e.g., 4), are counted, and the most frequently used one is defined as the representative intra prediction mode. Thus, in the examples above, video encoder 20 and video decoder 30 may define the representative intra prediction mode for the left neighboring column based on a most frequent intra prediction mode of the left neighboring column, and define the representative intra prediction mode for the above neighboring column based on a most frequent intra prediction mode of the above neighboring row.

[0101] Alternatively, the counting is based on the PU size of the current block. For instance, a video coder may determine the most frequent intra prediction mode of the neighboring blocks based on at least one of: (i) a smallest prediction unit size, or (ii) a prediction unit size of the block of video data. In one example, instead of counting based on the unit of a smallest PU size (e.g., 4x4), a different unit size may be used for a different PU size of current block. For example, if the PU size of the current block is 64x64, the counting is based on an 8x8 PU size or a 16x16 PU size, instead of the smallest PU size (4x4).

[0102] In some examples, a representative intra prediction mode may be defined as one of the intra prediction modes used by a selected neighboring block in the left (or above) neighboring column (or row). For example, the intra prediction mode of the sample (or block equal to the smallest PU size) located at the middle position of the left(right) neighboring column(row), which is not the left (top)-most or the right(bottom)-most one, is defined as the representative intra prediction mode of the left (or above) neighboring mode. Furthermore, in the examples above, the most frequent intra prediction mode of the neighboring blocks may be determined based on at least one of a smallest prediction unit size, or a prediction unit size of the block of video data. In another example, video encoder 20 and video decoder 30 may define the representative intra prediction mode for the left neighboring column as an intra prediction mode used by a selected neighboring block in the left neighboring column, and may define the representative intra prediction mode for the above neighboring row as an intra prediction mode used by a selected neighboring block in the above neighboring row.

[0103] As mentioned above, particular examples of this disclosure allow for *M* MPMs, where *M* is greater than three. Thus, in accordance with such examples of this disclosure, video encoder 20 may derive *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In other words, video encoder 20 may derive, from among the plurality of intra prediction modes, *M* MPMs for intra prediction of a PU of a CU of a current picture of the video data. In this example, *M* is greater than 3. In this example, *M* is less than a total number of intra prediction modes in the plurality of intra prediction modes. Furthermore, in this example, video encoder 20 may encode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. In this example, video encoder 20 may encode the indicated one of the MPM index or the non-MPM index. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode and the non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Video encoder 20 may encode the block of video data based on the selected intra prediction mode.

[0104] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, video encoder 20 may encode the syntax element that indicates whether a MPM index or a non-MPM index is used to indicate the selected intra prediction mode, video encoder 20 may encode a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the PU. Furthermore, as part of encoding the block of video data based on the selected intra prediction mode, video encoder 20 may generate, based on the selected intra prediction mode of the current PU, a predictive block for the current PU. Video encoder 20 may generate residual data that represents pixel differences between the current CU and the predictive block. In this example, video encoder 20 may output a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data, the coded pictures including the current picture.

[0105] Moreover, in accordance with such examples of this disclosure, video decoder 30 may derive *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In this example *M* is greater than 3. *M* may be less than the total number of intra prediction modes in the plurality of intra prediction modes. Furthermore, in this example, video decoder 30 may decode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. In this example, video decoder 30 may decode the indicated one of the MPM index or the non-MPM index. Video decoder 30 may reconstruct the block of video data based on the selected intra prediction mode.

[0106] For instance, in this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, to decoded the syntax element that indicates whether the MPM index or the non-MPM index is used to indicate the selected intra prediction mode, video decoder 30 may decode, from a bitstream, a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the current PU. Furthermore, as part of reconstructing the block of video data based on the selected intra prediction mode, video decoder 30 may generate a predictive block for the current PU using the selected intra prediction mode. In this example, video decoder 30 may reconstruct the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU. In this example, video decoder 30 may receive a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data, the coded pictures including the current picture.

[0107] In some examples, when deciding the *M* (*M* > 3) MPMs, the original three MPMs defined in HEVC are maintained in the MMPMs, and additional (*M*-3) intra prediction modes are added. For instance, in some examples, the additional three intra prediction modes can be selected such that, planar and DC modes are always included in the MPMs. In other words, the process of deriving the *M* MPMs may comprise including a planar mode and a DC mode among the *M* MPMs.

[0108] Alternatively, in some examples, the additional (M-3) intra prediction modes can be selected such that planar, DC, VERTICAL (i.e., INTRA_ANGULAR26) and/or HORIZONTAL (i.e., INTRA_ANGULAR10), and/or DIAGONAL (i.e., INTRA_ANGULAR18) modes are always included in the MPMs. In other words, the process of deriving the *M* MPMs may comprise including a Planar mode, a DC mode, and at least one of a VERTICAL mode, a HORIZONTAL mode or a DIAGONAL mode among the *M* MPMs.

[0109] In some examples, the additional (*M*-3) intra prediction modes may be selected from angular modes which are closest to the angular mode already included in the first three MPMs. In this example, the term 'closest' is measured by the difference of intra prediction mode indices or the absolute value of the difference of intra prediction mode indices, or the difference of intra prediction angles. For instance, in this example, as part of deriving the *M* MPMs, a video coder (e.g., video encoder 20 or video decoder 30) may derive three MPMs, wherein the three MPMs include at least one angular mode. Furthermore, in this example, the video coder may select one or more additional angular MPMs based on similarity with the at least angular mode. In this example, similarity is determined based on at least one of intra prediction mode index differences or intra prediction angle differences

[0110] In accordance with some examples of this disclosure, in order to reduce the number of bits used for coding index values, certain variations in the number of bits that are utilized to code index values may be included when coding the non-MPM index. Thus, a video coder, such as video encoder 20 and video decoder 30, may determine whether an MPM index or a non-MPM index is being used to indicate a selected intra prediction mode of a plurality of intra prediction modes and, when a non-MPM index is identified, use either a fixed length code to represent the non-MPM index or a variable length code to represent the non-MPM index, depending on whether one or more criteria are met. In various examples, a video coder may determine the various criteria in different ways. For example, the criteria may be met if the non-MPM index is one of a predetermined number of first non-MPM indices of the plurality of intra prediction modes. In some examples, the criteria are met if the non-MPM index is one of a predetermined number of last non-MPM indices. In some examples, the criteria are met if the non-MPM index is one of a predetermined number of selected non-MPM indices.

[0111] Thus, in accordance with one example of this disclosure, video encoder 20 may derive *M* most probable modes for intra prediction of a block of video data from among a plurality of intra prediction modes. In this example, *M* may be greater than 3. Furthermore, video encoder 20 may encode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. Furthermore, in this example, video encoder 20 may encode, based on the syntax element indicating the non-MPM index is used to indicate the selected intra prediction mode, the non-MPM index. In this example, the non-MPM index may be encoded in the bitstream as a code word shorter than $\lceil \log_2 N \rceil$ bits if the non-MPM index satisfies a criterion and is encoded as a fixed length code with $\lceil \log_2 N \rceil$ bits otherwise. In this example, there may be a total of N available values of the non-MPM index. Furthermore, video encoder 20 may encode the block of video data based on the selected intra prediction mode. In this disclosure, $\lceil\ \rceil$ represents the rounding up operation.

[0112] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data may be a selected intra prediction mode for a current PU of the current CU. Furthermore, in this example, to encode a syntax element that indicates whether an MPM index or a non-MPM index is used to indicate the selected intra prediction mode, video encoder 20 may include, in a bitstream, data indicating a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate, for the current PU, the selected intra prediction mode of a plurality of intra prediction modes. Furthermore, in some such examples, to encode the block of video data based on the selected intra prediction mode, video encoder 20 may generate a predictive block for the PU using the selected intra prediction mode and video encoder 20 may generate residual data that represents pixel differences between the current CU and the predictive block. In some examples, video encoder 20 may transform and quantize the residual data and include, in a bitstream, entropy encoded syntax elements representing the resulting quantized transform coefficients.

[0113] In one example, the criterion may be met if the non-MPM index is one of a predetermined number of first non-MPM indices of the plurality of intra prediction modes. For example, for non-MPM modes, both the encoder and decoder may assign an index value to each of the available non-MPM modes. With the decoded index value, the decoder will know which non-MPM mode is signaled. In one example, when a non-MPM index is one of the first M index values, the non-MPM index may be within the range of 0 ~ (M-1). In another example, the criterion may be met if the non-MPM index is one of a predetermined number of last non-MPM indices. In another example, the criterion may be met if the non-MPM index is one of a predetermined number of selected non-MPM indices.

[0114] In a similar example, video decoder 30 may derive *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes. In this example, *M* may be greater than 3. Furthermore, in this example, video decoder 30 may decode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data. As before, the MPM index indicates which of the *M* most probable modes is the selected intra prediction mode, and the non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Furthermore, in this example, based on the MPM index indicating the selected intra prediction mode, video decoder 30 may decode the non-MPM index. In this example, the non-MPM index is encoded in the bitstream as a code word shorter than $\lceil \log_2 N \rceil$ bits if the non-MPM index satisfies a criterion and is encoded in the bitstream as a fixed length code with $\lceil \log_2 N \rceil$ bits otherwise. In this example, there is a total of *N* available values of the non-MPM index. Video decoder 30 may reconstruct the block of video data based on the selected intra prediction mode.

[0115] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data may be a selected intra prediction mode for a current PU of the current CU. Furthermore, in this example, to decode the syntax element that indicates whether a MPM index or a non-MPM index is used to indicate the selected intra prediction mode for intra prediction of the current PU, video decoder 30 may decode or otherwise obtain, from a bitstream, a syntax element (e.g., prev_intra_luma_pred_flag) that indicates whether a MPM index (e.g., mpm_idx) or a non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate a selected intra prediction mode for the current PU. Furthermore, in some examples, as part of reconstructing the block of video data based on the selected intra prediction mode, video decoder 30 may generate a predictive block for the current PU using the selected intra prediction mode. In such examples, video decoder 30 may reconstruct the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU

[0116] In one example, the criterion may be met if the non-MPM index is one of a predetermined number of first non-MPM indices of the plurality of intra prediction modes. For instance, the criterion may be that the non-MPM index is one of the first *X* non-MPM indices of the plurality of intra prediction modes, *X* being an integer. In another example, the criterion may be met if the non-MPM index is one of a predetermined number of last non-MPM indices. For instance, the criterion may be that the non-MPM index is one of the last *X* non-MPM indices of the plurality of intra prediction modes, *X* being an integer. In another example, the criterion may be met if the non-MPM index is one of a predetermined number of selected non-MPM indices. For instance, the criterion may be that the non-MPM index is one of *X* selected non-MPM indices of the plurality of intra prediction modes, *X* being an integer.

[0117] Thus, when coding the non-MPM index, all the available *N* non-MPM indexes, except for the first *M* or the last *M* or selected *M,* index values may be coded using fixed length code with $\lceil \log_2 N \rceil$ bits, while the M index values is signaled using code word shorter than $\lceil \log_2 N \rceil$ bits. In this example, *M* may or may not be equal to the number of MPMs. Alternatively, all the available *N* non-MPM indexes are coded using fixed length code with $\lceil \log_2 N \rceil$ bits.

[0118] Alternatively, in some examples, a one-bit flag, indicating whether the non-MPM index refers to a horizontal or vertical mode, is first signaled with context modeling, then an index further specifying which non-MPM is selected is further signaled. Thus, in this example, video encoder 20 may encode, based on a syntax element (e.g., prev_intra_luma_pred_flag) indicating the non-MPM index (e.g., rem_intra_luma_pred_mode) is used to indicate the selected intra prediction mode for the block of video data, a second syntax element indicating whether the non-MPM index refers to one of a horizontal mode or a vertical mode. Similarly, video decoder 30 may decode, from the bitstream, based on a syntax element (e.g., prev_intra_luma_pred_flag) indicating a non-MPM index is used, a second syntax element indicating whether the non-MPM index refers to one of a horizontal mode or a vertical mode.

[0119] In some examples, when more than three MPMs are used for signaling the intra prediction mode, during the encoder mode decision, for each PU, the first *K* (*K*>=2) MPMs are inserted into the candidate intra prediction mode list which is to be checked by rate-distortion optimization. The value of *K* may depend on whether the left neighboring intra prediction mode *IntraModeL* and above neighboring intra prediction mode *IntraModeA* are the same. For example, *K* = (*IntraModeL*== *IntraModeA*)? 2 : 3. In other words, in this example, *K* is set equal to 2 if *IntraModeL* is equal to *IntraModeA* and set equal to 3 otherwise.

[0120] In some examples, the number of MPMs may depend on CU or PU sizes. For example, the number of MPMs may be greater for larger CU or PU sizes. Alternatively, in some examples, the number of MPMs may be signaled in a sequence parameter set, a picture parameter set, or a slice header.

[0121] In accordance with some examples of this disclosure, a video coder, such as video encoder 20 and video decoder 30, may use more than 33 angular intra prediction modes. For example, the video coder may use 65 angular modes. In this example, the 65 prediction angles (directions) may be defined such that the interpolation is performed in 1/32 pel (pixel) accuracy. As discussed above, when a PU is encoded using an angular intra prediction mode, a video coder may determine a fractional position between two reconstructed neighboring samples by projecting, along a prediction direction associated with the angular intra prediction mode, the coordinates of a sample of the predictive block of the PU. When the interpolation is performed in 1/32 pel accuracy, 31 equidistant positions are defined between the two adjacent reconstructed neighboring pixels and the video coder sets the determined fractional position as to the closest one of the reconstructed neighboring pixels or the 31 positions. Using a limited number of positions instead of a real number line may simplify the processing involved in performing the interpolation calculations. As briefly described above, performing interpolation using 1/32 pel accuracy despite more than 33 angular intra prediction modes being available may increase accuracy while keeping the level of complexity the same. In particular, in order to perform interpolations with increased accuracy, an increased number of different interpolation filters would need to be designed and stored at both the encoder and the decoder. Therefore, for example, by avoiding the need for increased interpolation accuracy, the complexity (only 32 different interpolation filters still being utilized) of the video coder may remain the same using 1/32 pel interpolations.

[0122] Thus, in the examples above where more than 33 angular intra prediction modes are used (e.g., 65 angular intra prediction modules are used) and interpolation is performed in 1/32 pel accuracy, as part of encoding a block of the video data, video encoder 20 may encode syntax information (e.g., mpm_idx or rem_intra_luma_pred_mode) that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes that includes more than 33 angular intra prediction modes. In this example, the angular intra prediction modes are defined such that interpolation is performed in 1/32 pel accuracy. Furthermore, in this example, video encoder 20 may encode the block of video data based on the selected intra prediction mode.

[0123] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. Furthermore, in this example, as part of encoding the block of video data, video encoder 20 may, for each respective sample of a predictive block of the current PU, video encoder 20 may determine a fractional position between two neighboring reconstructed samples of the current picture. FIG. 5 is a conceptual diagram illustrating an example technique for generating a prediction sample for a block of video data according to an angular intra prediction mode. For instance, in the example of FIG. 5, video encoder 20 may determine a fractional position α between neighboring samples L and R. The two neighboring reconstructed samples neighbor the current PU. Video encoder 20 may determine the fractional position by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples. For instance, in the example of FIG. 5, line 60 shows the prediction direction associated with the selected intra prediction mode and (x,y) is the coordinate of the respective sample 62. In this example, video encoder 20 may calculate a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position. This interpolation is in 1/32 pel accuracy. Additionally, in this example, as part of encoding the block of video data, video encoder 20 may generate residual data that represents pixel differences between the current block and the predictive block. Video encoder 20 may transform and quantize the residual data and include, in a bitstream, entropy encoded syntax elements representing the resulting quantized transform coefficients.

[0124] Similarly, in one example, video decoder 30 may decode syntax information (e.g., mpm_idx or rem_intra_luma_pred_mode) that indicates a selected intra prediction mode for a block of video data from among a plurality of intra prediction modes. In this example, the plurality of intra prediction modes includes greater than 33 angular intra prediction modes. The angular intra prediction modes may be defined such that interpolation is performed in 1/32 pel accuracy. Video decoder 30 may reconstruct the block of video data based on the selected intra prediction mode.

[0125] Thus, in this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data may be a selected intra prediction mode for a current PU of the current CU. As part of reconstructing the block of video data, for each respective sample of a predictive block of the current PU, video decoder 30 may determine a fractional position between two neighboring reconstructed samples of the current picture by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples. Additionally, video decoder 30 may calculate a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position. In this example, this interpolation is in 1/32 pel accuracy. Video decoder 30 may reconstruct a coding block of the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU.

[0126] In the examples above, video encoder 20 and video decoder 30 may use more than 33 angular intra prediction modes, according to the invention. For instance, the plurality of angular intra prediction modes may include 65 angular intra prediction modes. Furthermore, in the examples above, the interpolation filter may be a two-tap bi-linear interpolation filter. In some such examples, the interpolation filter is formulated as:

$$p_{xy} = (1-\alpha)\cdot L + \alpha\cdot R$$

where p_{xy} is the calculated value of the respective sample, L and R are values of the two reconstructed neighboring samples, and α is the determined fractional position.

[0127] As indicated above, in accordance with some examples of this disclosure, a video coder may use more than 33 angular intra prediction modes, according to the invention. In one such example, for each PU, it is restricted that only N angular modes are available to be used, which are adaptively selected from all the available angular modes using the coded information of the left and above neighboring blocks. For instance, in this example, video encoder 20 is prohibited from (i.e., restricted from) selecting particular angular intra prediction modes when encoding a block (e.g., PU) of a picture. In this example, video encoder 20 may be prohibited from selecting the particular angular intra prediction modes despite the particular angular intra prediction modes being available to video encoder 20 for use in encoding other blocks of the same picture, and despite the fact that the particular angular intra prediction modes may offer better compression performance than angular intra prediction modes that video encoder 20 is allowed to use for encoding the block.

[0128] Thus, in an example according to the invention where only N angular intra prediction modes are available to be used, video encoder 20 and video decoder 30 may select, based on intra prediction modes used to decode one or more neighboring blocks, a subset of N angular intra prediction modes from among the plurality of angular intra prediction modes. In this example, the syntax information that indicates the selected intra prediction mode for the current PU may comprise an index that indicates the selected intra prediction mode. In this example, the index is a fixed length code consisting of $\lceil \log_2 N \rceil$ bits. In this example, *N* may be various integers, such as 33.

[0129] In some examples, the number of total intra prediction modes depends on CU or PU sizes. For instance, a video coder (e.g., video encoder 20 or video decoder 30) may determine, based on at least one of a CU size or a PU size, a total number of intra prediction modes in the plurality of intra prediction modes. For example, if a CU size or a PU size is below a particular threshold only angular intra prediction modes with index values less than a particular number are available. In some examples, a total number of intra prediction modes may be signaled in an SPS, a PPS, or a slice header.

[0130] As discussed above, video encoder 20 and video decoder 30 may apply an interpolation filter to reconstructed neighboring samples to determine values of samples in a predictive block. In accordance with some techniques of this disclosure, an *N*-tap intra interpolation filter can be applied, where *N* is larger than 2. For instance, *N* may be 3 and 4. When a video coder, such as video encoder 20 or video decoder 30, uses an *N*-tap intra interpolation filter to determine the value of a sample in a predictive block, the video coder may calculate the value based on the values of *N* reconstructed neighboring samples. Using N-tap (N>2) interpolation may be advantageous over existing 2-tap interpolation since it may provide more accurate interpolation results by including more reference samples (N reference samples) during the interpolation. Statistically, using more reference samples will introduce better interpolation accuracy with higher complexity. The *N*-tap intra interpolation filter may include a sinc interpolation filter, a Gaussian interpolation filter and an interpolation filter derived using an image correlation model. A sinc filter removes all frequency components above a given cutoff frequency, without affecting lower frequencies, and has linear phase response. Thus, the *N*-tap interpolation filter may comprise at least one of a sinc interpolation filter, a Gaussian interpolation filter and an interpolation filter derived using an image correlation model.

[0131] In this way, video encoder 20 may encode syntax information that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes. For example, video encoder 20 may signal, in a bitstream, a mpm_idx or a rem_intra_luma_pred_mode syntax element for the current PU. Furthermore, video encoder 20 may apply an *N*-tap intra interpolation filter to neighboring reconstructed samples of the block of video data according to the selected intra prediction mode, where *N* is greater than 2. Video encoder 20 may apply the *N*-tap intra interpolation filter to determine filtered neighboring reconstructed samples that may correspond to positions between full-pel neighboring reconstructed samples. Video encoder 20 may encode the block of video data based on the filtered neighboring reconstructed samples according to the selected intra prediction mode. For instance, video encoder 20 may use the filtered neighboring reconstructed samples to generate a predictive block and generate residual data for the block of video data based on the predictive block.

[0132] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. Furthermore, in this example, for each respective sample of a prediction block of the current PU, video encoder 20 may determine a fractional position between two neighboring reconstructed samples of a set of neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples. The set of neighboring reconstructed samples including reconstructed samples above and left of the current PU in the current picture. Furthermore, video encoder 20 may calculate a value of the respective sample by applying an *N*-tap intra interpolation filter to neighboring reconstructed samples to interpolate a value at the determined fractional position, wherein N is greater than 2. As part of encoding the block of video data based on the filtered neighboring reconstructed samples, video encoder 20 may generate residual data that represents pixel differences between the current CU and the predictive block. Video encoder 20 may transform and quantize the residual data and include, in a bitstream, entropy encoded syntax elements representing the resulting quantized transform coefficients.

[0133] Similarly, video decoder 30 may decode syntax information that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes. For example, video decoder 30 may obtain, from a bitstream, a mpm idx or a rem_intra_luma_pred_mode syntax element. Furthermore, video decoder 30 may apply an *N*-tap intra interpolation filter to neighboring reconstructed samples of the block of video data according to the selected intra prediction mode, wherein N is greater than 2. Furthermore, video decoder 30 may reconstruct the block of video data based on the filtered neighboring reconstructed samples according to the selected intra prediction mode.

[0134] In this example, the block of video data may be a current CU in a current picture of the video data and the selected intra prediction mode for the block of video data is a selected intra prediction mode for a current PU of the current CU. In this example, for each respective sample of a predictive block of the current PU, video decoder 30 may determine a fractional position between two neighboring reconstructed samples of a set of neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples. The set of neighboring reconstructed samples including reconstructed samples above and left of the current PU in the current picture. Video decoder 30 may calculate a prediction value of the respective sample by applying an N-tap intra interpolation filter to neighboring reconstructed samples to interpolate a value at the determined fractional position, wherein N is greater than 2. In this example, as part of reconstructing the block of video data (e.g., the current CU), video decoder 30 may reconstruct a coding block of the current CU using residual values by adding samples of the predictive blocks of PUs of the CU to corresponding samples of transform blocks of TUs of the current CU.

[0135] When the interpolation filter is derived using an image correlation model, an image correlation function based on Generalized Gaussian function may be applied, and the interpolation function for each fractional position is further derived using the least mean square estimate. For example, when deriving the interpolation filter using an image correlation model, assuming the image correlation model is $R(i,i) = \rho_x^{|i|} \cdot \rho_y^{|j|}$, where i and j indicates the distance between two image samples in the horizontal and vertical axis, ρ_x is the image correlation factor in horizontal direction, ρ_y is the image correlation factor in vertical direction, both ρ_x and ρ_y may be within the range of [0, 1], meaning that a larger image distance outputs a smaller correlation. In another example, assuming there are four samples, a, b, c and d, and the fractional sample e is to be interpolated. To derive a desirable interpolation filter [f0, f1, f2, f3], the interpolation process $[a, b, c, d] * [f0\ f1, f2, f3]^T = [e]$ may be rewritten as $A * F = E$, where $A = [a, b, c, d]$, $F = [f0\ f1, f2, f3]^T$, and $E = [e]$. According to a least mean square estimate, the optimal filter F is $F^* = (A^T*A)^{-1}*A^TE$, and "$(A^T*A)^{-1}*A^TE$" may be determined by the assumed image correlation model. Therefore, using the image correlation model, the interpolation filter for each fractional sample may be derived using a least mean square estimate.

[0136] Thus, in some examples where a video coder derives the intra interpolation filter using an image correlation model, the video coder may apply an image correlation function based on Generalized Gaussian function. Additionally, the video coder may derive an interpolation function for each fractional position using a least mean square estimate.

[0137] In some examples, when a video coder applies a Gaussian interpolation filter, the parameter controlling the smoothing strength of the filter may vary depending on block size. Therefore, in such examples, the intra interpolation filter may comprise a Gaussian interpolation filter and a parameter controlling a smoothing strength of the Gaussian interpolation filter varies based on block sizes. For example, the Gaussian interpolation filter may be formulated as:

$$G(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}},$$

G(*x, σ*), x indicates the fractional position, and *σ* controls the smoothing strength. Therefore, a larger value of *σ* generates a Gaussian interpolation filter with increased smoothing strength.

[0138] Furthermore, in some examples, a video coder may select the interpolation filter to apply. In such examples, the selection of the filter may depend on the block sizes and or relative sample position inside the block. Therefore, in such examples, the video coder may select the intra interpolation filter based on at least on of block size or relative sample position inside the block. For example, assuming a set of different interpolation filters have been designed, namely f1, f2, ... fy, different interpolation filters may be used for different pixel positions inside the current block. For example, interpolation filter f1 may be utilized for the first two rows of pixels inside the current block, and interpolation filter f2 may be utilized for the remaining rows of pixels.

[0139] Different interpolation filters may be available for intra prediction of a single block, and the selection of the interpolation filter may be explicitly signaled or derived from neighboring decoded information, including the reconstructed sample values and intra prediction modes. Thus, in such examples, the video coder may select the intra interpolation filter based on at least one of reconstructed sample values or intra prediction modes of neighboring blocks. For example, assuming a set of different interpolation filters have been designed, namely f1, f2, ... fy, different interpolation filters may be used for different intra prediction modes. For example, interpolation filter f1 may be utilized for the diagonal Intra prediction mode, and interpolation filter f2 may be utilized for the vertical Intra prediction mode.

[0140] In accordance with some examples of this disclosure, multiple intra prediction directions can be allowed for one PU without additional signaling of multiple intra prediction modes. For example, video decoder 30 may decode syntax information that indicates a selected intra prediction mode for a block of video data (e.g., a PU) from among a plurality of intra prediction modes. Furthermore, video decoder 30 may determine one or more additional intra prediction modes for the block without decoding additional syntax information indicting additional intra prediction modes. In this example, video decoder 30 may reconstruct the block of video data according to the selected intra prediction mode and the one or more additional intra prediction modes. For instance, video decoder 30 may reconstruct a CU containing the block based on residual data for the CU and a predictive block for the block generated based on the selected intra prediction mode and the one or more additional intra prediction modes.

[0141] In some examples where multiple intra prediction directions are allowed for one PU without additional signaling of multiple intra prediction modes, the PU comprises a plurality of sub-blocks and each sub-block within the PU may have its own intra prediction mode. In this example, a sub-block within a PU is a 4x4 block. In this way, in one example, as part of determining the one or more additional intra prediction modes for the block of video data mentioned above, video decoder 30 may determine respective intra prediction modes for each of a plurality of sub-blocks of the block of video data without decoding additional syntax information indicating additional intra prediction modes. Furthermore, in this example, as part of reconstructing the block of video data, video decoder 30 may reconstruct the sub-blocks according to the respective determined intra prediction modes.

[0142] Furthermore, in some examples, the intra prediction direction for one sub-block can be derived as the intra prediction mode of the closest above or left reference block. In this way, as part of determining respective intra prediction modes for each of a plurality of sub-blocks of the block of video data, video decoder 30 may derive a respective intra prediction mode for at least one of the sub-blocks as an intra prediction mode of a closest above or left reference block. In some examples, for each sub-block of the current PU, the prediction is a weighted sum of two prediction directions, which come from the closest block in the above reference row and the left reference column. In this way, as part of determining respective intra prediction modes for each of a plurality of sub-blocks of the block of video data, video decoder 30 may derive a respective intra prediction mode for at least one of the sub-blocks based on one or both of an above reference block and a left reference block.

[0143] Similarly, as part of a process of encoding video data, video encoder 20 may encode syntax information that indicates a selected intra prediction mode for a block of video data (e.g., a PU) from among a plurality of intra prediction modes. Video encoder 20 may determine one or more additional intra prediction modes for the block of video data without encoding additional syntax information indicting additional intra prediction modes. Video encoder 20 may encode the block of video data according to the selected intra prediction mode and the one or more additional intra prediction modes. In some examples, as part of determining one or more additional intra prediction modes for the block of video data, video encoder 20 may determine respective intra prediction modes for each of a plurality of sub-blocks of the block of video data without encoding additional syntax information indicting additional intra prediction modes, and, as part of encoding the block of video data, video encoder 20 may encode the sub-blocks according to the respective determined intra prediction modes. In some examples, as part of determining respective intra prediction modes for each of a plurality of sub-blocks of the block of video data, video encoder 20 may derive a respective intra prediction mode for at least one of the sub-blocks as an intra prediction mode of a closest above or left reference block. In some examples, as part of determining respective intra prediction modes for each of a plurality of sub-blocks of the block of video data, video encoder 20 may derive a respective intra prediction mode for at least one of the sub-blocks based on one or both of an above reference block and a left reference block.

[0144] As mentioned above, video encoder 20 may perform a process to decide which intra prediction mode to select for a PU. Because performing a full rate-distortion optimization process for all 35 available intra prediction modes in HEVC may be impractical in some instances, video encoder 20 may determine a list of *N* intra prediction mode candidates by checking intra prediction modes using a "Sum of Absolute Transform Difference" (SATD) criterion. When more than 35 intra prediction modes are available, as is the case in some examples of this disclosure, even checking the intra prediction modes using the SATD criterion may become excessively time consuming or resource intensive.

[0145] Hence, in accordance with an example of this disclosure, to reduce, e.g., largely reduce, the number of SATD checks for intra angular prediction at video encoder 20, a *K* stage SATD check scheme is proposed. In this example, *K* is greater than 2. In this example, for each stage of SATD check, only a selected number of intra prediction modes, which come from a pre-defined subset of intra prediction angles, may be checked with SATD cost. For instance, in a first stage of SATD check, video encoder 20 may check each of the intra prediction angles from the pre-defined subset of intra prediction modes with SATD cost, and may decide the best *N* intra prediction modes. In this example, *N* is a pre-defined constant value depending on the block size. After that, for each remaining stage of the SATD check, video encoder 20 only checks the intra prediction angles which are close, in terms of prediction angle difference, to the best *N* intra prediction modes decided in the previous stage with SATD cost, and video encoder 20 decides the best *N* intra prediction modes for the next stage.

[0146] In one example, when the proposed 65 angular modes are applied, video encoder 20 applies a two stage SATD check scheme to decide which of the intra prediction modes will be further examined with more efficient cost criterion (e.g., rate-distortion (R-D) cost) at the encoder side. In the first stage, video encoder 20 checks only the original 35 intra prediction modes as defined in HEVC, including Planar, DC and 33 angular modes, with SATD cost, then the best *N* (*N* can be the same as used in HEVC reference software) intra prediction modes are decided. In the second stage, video encoder 20 further checks only the intra angular modes which are the direct neighbors (angular mode index ±1) of, i.e., adjacent to, the best N angular modes with SATD cost. Video encoder 20 may decide to further check the final best *N* intra prediction modes with R-D cost.

[0147] Thus, in some examples, video encoder 20 may determine a respective SATD cost for encoding the block of video data according to each of a first subset of intra prediction modes. Additionally, video encoder 20 may determine a second subset of the intra prediction modes based on the determined SATD costs of the first subset of intra prediction modes. Video encoder 20 may determine a respective SATD cost for encoding the block of video data according to each of the second subset of intra prediction modes. Furthermore, video encoder 20 may select one of the intra prediction modes based on the determined SATD costs of the first and second subsets of intra prediction modes. Video encoder 20 may encode the block of video data according to the selected one of the intra prediction modes.

[0148] In this example, as part of determining the second subset of the intra prediction modes based on the determined SATD costs of the first subset of intra prediction modes comprises, video encoder 20 may identify *N* intra prediction modes of the first subset of intra prediction modes having a lowest *N* of the determined SATD costs. Video encoder 20 may include intra prediction modes having intra prediction angles proximate to intra prediction angles of the identified *N* intra prediction modes in the second subset of intra prediction modes. As part of including intra prediction modes having intra prediction angles proximate to intra prediction angles of the identified N intra prediction modes in the second subset of intra prediction modes, video encoder 20 may include intra prediction modes having intra prediction angles adjacent to intra prediction angles of the identified N intra prediction modes in the second subset of intra prediction modes.

[0149] Furthermore, in some examples, as part of selecting one of the intra prediction modes based on the determined SATD costs of the first and second subsets of intra prediction modes, video encoder 20 may identify *N* intra prediction modes of the first and second subsets of intra prediction modes having a lowest *N* of the determined SATD costs. In this example, video encoder 20 may determine a respective rate-distortion cost for each of the identified N intra prediction modes. Moreover, in this example, video encoder 20 may select one of the identified N intra prediction modes based on the determined rate-distortion costs.

[0150] In the examples of this disclosure related to the SATD checks, the value of *N* depends on a block size of the block of video data. Furthermore, in some such examples, the first subset of intra prediction modes may include a Planar intra prediction mode, a DC intra prediction mode, and 33 angular prediction modes.

[0151] In some examples of this disclosure related to the SATD checks, the first and second subsets of intra prediction modes comprise first and second ones *of K* subsets of intra prediction modes and *K* is greater than or equal to 2. In such examples, for each current subset of the *K* subsets of intra prediction modes, video encoder 20 may determine a respective SATD cost for encoding the block of video data according to each intra prediction mode of the current subset of intra prediction modes. Additionally, video encoder 20 may determine a next subset of the *K* subsets of intra prediction modes based on the determined SATD costs of the current subset of intra prediction modes.

[0152] FIG. 6 is a block diagram illustrating an example video encoder 20 that may implement the techniques of this disclosure. FIG. 6 is provided for purposes of explanation and should not be considered limiting of the techniques as broadly exemplified and described in this disclosure. The techniques of this disclosure may be applicable to any of a variety of coding standards or methods.

[0153] In the example of FIG. 6, video encoder 20 includes a prediction processing unit 100, video data memory 101, a residual generation unit 102, a transform processing unit 104, a quantization unit 106, an inverse quantization unit 108, an inverse transform processing unit 110, a reconstruction unit 112, a filter unit 114, a decoded picture buffer 116, and an entropy encoding unit 118. Prediction processing unit 100 includes an inter-prediction processing unit 120 and an intra prediction processing unit 126. Inter-prediction processing unit 120 includes a motion estimation (ME) unit 122 and a motion compensation (MC) unit 124. In other examples, video encoder 20 may include more, fewer, or different functional components.

[0154] Video data memory 101 may store video data to be encoded by the components of video encoder 20. The video data stored in video data memory 101 may be obtained, for example, from video source 18. Decoded picture buffer 116 may be a reference picture memory that stores reference video data for use in encoding video data by video encoder 20, e.g., in intra- or inter-coding modes. Video data memory 101 and decoded picture buffer 116 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 101 and decoded picture buffer 116 may be provided by the same memory device or separate memory devices.

[0155] Video encoder 20 may receive video data. Video encoder 20 may encode each CTU in a slice of a picture of the video data. Each of the CTUs may be associated with equally-sized luma coding tree blocks (CTBs) and corresponding CTBs of the picture. As part of encoding a CTU, prediction processing unit 100 may perform quad-tree partitioning to divide the CTBs of the CTU into progressively-smaller blocks. The smaller blocks may be coding blocks of CUs. For example, prediction processing unit 100 may partition a CTB associated with a CTU into four equally-sized sub-blocks, partition one or more of the sub-blocks into four equally-sized sub-sub-blocks, and so on.

[0156] Video encoder 20 may encode CUs of a CTU to generate encoded representations of the CUs (i.e., coded CUs). As part of encoding a CU, prediction processing unit 100 may partition the coding blocks associated with the CU among one or more PUs of the CU. Thus, each PU may be associated with a luma prediction block and corresponding chroma prediction blocks. Video encoder 20 and video decoder 30 may support PUs having various sizes. As indicated above, the size of a CU may refer to the size of the luma coding block of the CU and the size of a PU may refer to the size of a luma prediction block of the PU. Assuming that the size of a particular CU is 2Nx2N, video encoder 20 and video decoder 30 may support PU sizes of 2Nx2N or NxN for intra prediction, and symmetric PU sizes of 2Nx2N, 2NxN, Nx2N, NxN, or similar for inter prediction. Video encoder 20 and video decoder 30 may also support asymmetric partitioning for PU sizes of 2NxnU, 2NxnD, nLx2N, and nRx2N for inter prediction.

[0157] Inter-prediction processing unit 120 may generate predictive data for a PU by performing inter prediction on each PU of a CU. The predictive data for the PU may include predictive blocks of the PU and motion information for the PU. Intra prediction processing unit 126 may generate predictive data for a PU by performing intra prediction on the PU. The predictive data for the PU may include predictive blocks for the PU and various syntax elements. Intra prediction processing unit 126 may perform intra prediction on PUs in I slices, P slices, and B slices.

[0158] To perform intra prediction on a PU, intra prediction processing unit 126 may use multiple intra prediction modes to generate multiple sets of predictive blocks for the PU. When performing intra prediction using a particular intra prediction mode, intra prediction processing unit 126 may generate predictive blocks for the PU using a particular set of samples from neighboring blocks. The neighboring blocks may be above, above and to the right, above and to the left, or to the left of the prediction blocks of the PU, assuming a left-to-right, top-to-bottom encoding order for PUs, CUs, and CTUs. Intra prediction processing unit 126 may use various numbers of intra prediction modes, e.g., DC, planar, or directional/angular prediction modes, as described herein. In some examples, the number of intra prediction modes may depend on the size of the prediction blocks of the PU.

[0159] Prediction processing unit 100 may select the predictive data for PUs of a CU from among the predictive data generated by inter-prediction processing unit 120 for the PUs or the predictive data generated by intra prediction processing unit 126 for the PUs. In some examples, prediction processing unit 100 selects the predictive data for the PUs of the CU based on rate/distortion metrics of the sets of predictive data. The predictive blocks of the selected predictive data may be referred to herein as the selected predictive blocks.

[0160] Residual generation unit 102 may generate, based on the luma, Cb and Cr coding block of a CU and the selected predictive luma, Cb and Cr blocks of the PUs of the CU, luma, Cb and Cr residual blocks of the CU. For instance, residual generation unit 102 may generate the residual blocks of the CU such that each sample in the residual blocks has a value equal to a difference between a sample in a coding block of the CU and a corresponding sample in a corresponding selected predictive block of a PU of the CU.

[0161] Transform processing unit 104 may perform quad-tree partitioning to partition the residual blocks of a CU into transform blocks associated with TUs of the CU. Thus, a TU may be associated with a luma transform block and two corresponding chroma transform blocks. The sizes and positions of the luma and chroma transform blocks of TUs of a CU may or may not be based on the sizes and positions of prediction blocks of the PUs of the CU.

[0162] Transform processing unit 104 may generate transform coefficient blocks for each TU of a CU by applying one or more transforms to the transform blocks of the TU. Transform processing unit 104 may apply various transforms to a transform block associated with a TU. For example, transform processing unit 104 may apply a discrete cosine transform (DCT), a directional transform, or a conceptually-similar transform to a transform block. In some examples, transform processing unit 104 does not apply transforms to a transform block. In such examples, the transform block may be treated as a transform coefficient block.

[0163] Quantization unit 106 may quantize the transform coefficients in a coefficient block. The quantization process may reduce the bit depth associated with some or all of the transform coefficients. For example, an n-bit transform coefficient may be rounded down to an m-bit transform coefficient during quantization, where n is greater than $m.$ Quantization unit 106 may quantize a coefficient block associated with a TU of a CU based on a quantization parameter (QP) value associated with the CU. Video encoder 20 may adjust the degree of quantization applied to the coefficient blocks associated with a CU by adjusting the QP value associated with the CU. Quantization may introduce loss of information, thus quantized transform coefficients may have lower precision than the original ones.

[0164] Inverse quantization unit 108 and inverse transform processing unit 110 may apply inverse quantization and inverse transforms to a coefficient block, respectively, to reconstruct a residual block from the coefficient block. Reconstruction unit 112 may add the reconstructed residual block to corresponding samples from one or more predictive blocks generated by prediction processing unit 100 to produce a reconstructed transform block associated with a TU. By reconstructing transform blocks for each TU of a CU in this way, video encoder 20 may reconstruct the coding blocks of the CU.

[0165] Filter unit 114 may perform one or more deblocking operations to reduce blocking artifacts in the coding blocks associated with a CU. Decoded picture buffer 116 may store the reconstructed coding blocks after filter unit 114 performs the one or more deblocking operations on the reconstructed coding blocks. Inter-prediction processing unit 120 may use a reference picture that contains the reconstructed coding blocks to perform inter prediction on PUs of other pictures. In addition, intra prediction processing unit 126 may use reconstructed coding blocks in decoded picture buffer 116 to perform intra prediction on other PUs in the same picture as the CU.

[0166] Entropy encoding unit 118 may receive data from other functional components of video encoder 20. For example, entropy encoding unit 118 may receive coefficient blocks from quantization unit 106 and may receive syntax elements from prediction processing unit 100. Entropy encoding unit 118 may perform one or more entropy encoding operations on the data to generate entropy-encoded data. For example, entropy encoding unit 118 may perform a context-adaptive variable length coding (CAVLC) operation, a CABAC operation, a variable-to-variable (V2V) length coding operation, a syntax-based context-adaptive binary arithmetic coding (SBAC) operation, a Probability interval Partitioning Entropy (PIPE) coding operation, an Exponential-Golomb encoding operation, or another type of entropy encoding operation on the data. Video encoder 20 may output a bitstream that includes entropy-encoded data generated by entropy encoding unit 118.

[0167] Video encoder 20 is an example of a video encoder configured to perform any of the techniques described in this disclosure. In accordance with one or more techniques of this disclosure, one or more units within video encoder 20 may perform the techniques described herein as part of a video encoding process. For example, intra prediction processing unit 126 and/or entropy encoding unit 118, may be configured to perform the techniques described herein as part of a video encoding process. In some examples, intra prediction processing unit 126 may be configured to perform the techniques described herein for deriving or selecting intra prediction modes and/or MPMs for intra prediction, the techniques using an increased number of MPMs and/or angular modes, and the techniques for applying an N-tap intra interpolation filter, where N is larger than 2. Intra prediction processing unit 126 may be configured to perform the techniques described herein for allowing multiple Intra prediction directions per block, e.g. respective directions for sub-blocks of the block, which may not require additional signaling of multiple intra prediction modes.

[0168] Intra prediction processing unit 126 may provide syntax information to entropy encoding unit 118, such as a selected intra prediction mode and/or MPM index for intra prediction. Intra prediction processing unit 126 and/or entropy encoding unit 118 may be configured to implement the techniques described herein for encoding or signaling the syntax information related to intra prediction, e.g., the selected intra prediction mode and/or MPM index for intra prediction, in an encoded video bitstream, which may be decoded by video decoder 30.

[0169] FIG. 7 is a block diagram illustrating an example video decoder 30 that is configured to implement the techniques of this disclosure. FIG. 7 is provided for purposes of explanation and is not limiting on the techniques as broadly exemplified and described in this disclosure. The techniques of this disclosure may be applicable to a variety of coding standards or methods.

[0170] In the example of FIG. 7, video decoder 30 includes an entropy decoding unit 150, video data memory 151, a prediction processing unit 152, an inverse quantization unit 154, an inverse transform processing unit 156, a reconstruction unit 158, a filter unit 160, and a decoded picture buffer 162. Prediction processing unit 152 includes a motion compensation unit 164 and an intra prediction processing unit 166. In other examples, video decoder 30 may include more, fewer, or different functional components.

[0171] Video data memory 151 may store video data, such as an encoded video bitstream, to be decoded by the components of video decoder 30. The video data stored in video data memory 151 may be obtained, for example, from computer-readable medium 16, e.g., from a local video source, such as a camera, via wired or wireless network communication of video data, or by accessing physical data storage media. Video data memory 151 may form a coded picture buffer (CPB) that stores encoded video data from an encoded video bitstream. Decoded picture buffer 162 may be a reference picture memory that stores reference video data for use in decoding video data by video decoder 30, e.g., in intra- or inter-coding modes. Video data memory 151 and decoded picture buffer 162 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 151 and decoded picture buffer 162 may be provided by the same memory device or separate memory devices.

[0172] A coded picture buffer (CPB) formed by video data memory 151 may receive and store encoded video data (e.g., NAL units) of a bitstream. Entropy decoding unit 150 may receive NAL units from CPB formed by video data memory 151 and parse the NAL units to obtain syntax elements from the bitstream. Entropy decoding unit 150 may entropy decode entropy-encoded syntax elements in the NAL units. Prediction processing unit 152, inverse quantization unit 154, inverse transform processing unit 156, reconstruction unit 158, and filter unit 160 may generate decoded video data based on the syntax elements extracted from the bitstream.

[0173] The NAL units of the bitstream may include coded slice NAL units. As part of decoding the bitstream, entropy decoding unit 150 may extract and entropy decode syntax elements from the coded slice NAL units. Each of the coded slices may include a slice header and slice data. The slice header may contain syntax elements pertaining to a slice.

[0174] In addition to decoding syntax elements from the bitstream, video decoder 30 may perform a decoding operation on a CU. By performing the decoding operation on a CU, video decoder 30 may reconstruct coding blocks of the CU.

[0175] As part of performing a decoding operation on a CU, inverse quantization unit 154 may inverse quantize, i.e., de-quantize, coefficient blocks associated with TUs of the CU. Inverse quantization unit 154 may use a QP value associated with the CU of the TU to determine a degree of quantization and, likewise, a degree of inverse quantization for inverse quantization unit 154 to apply. That is, the compression ratio, i.e., the ratio of the number of bits used to represent original sequence and the compressed one, may be controlled by adjusting the value of the QP used when quantizing transform coefficients. The compression ratio may also depend on the method of entropy coding employed.

[0176] After inverse quantization unit 154 inverse quantizes a coefficient block, inverse transform processing unit 156 may apply one or more inverse transforms to the coefficient block in order to generate a residual block associated with the TU. For example, inverse transform processing unit 156 may apply an inverse DCT, an inverse integer transform, an inverse Karhunen-Loeve transform (KLT), an inverse rotational transform, an inverse directional transform, or another inverse transform to the coefficient block.

[0177] If a PU is encoded using intra prediction, intra prediction processing unit 166 may perform intra prediction to generate predictive blocks for the PU. Intra prediction processing unit 166 may use an intra prediction mode to generate the predictive luma, Cb, and Cr blocks for the PU based on reconstructed sample values of spatially-neighboring blocks. Intra prediction processing unit 166 may determine the intra prediction mode for the PU based on one or more syntax elements decoded from the bitstream.

[0178] Prediction processing unit 152 may construct a first reference picture list (RefPicListO) and a second reference picture list (RefPicList1) based on syntax elements extracted from the bitstream. Furthermore, if a PU is encoded using inter prediction, entropy decoding unit 150 may obtain motion information for the PU. Motion compensation unit 164 may determine, based on the motion information of the PU, one or more reference regions for the PU. Motion compensation unit 164 may generate, based on samples at the one or more reference blocks for the PU, predictive luma, Cb, and Cr blocks for the PU.

[0179] Reconstruction unit 158 may use the residual values from the luma, Cb, and Cr transform blocks associated with TUs of a CU and the predictive luma, Cb, and Cr blocks of the PUs of the CU, i.e., either intra prediction data or inter-prediction data, as applicable, to reconstruct the luma, Cb, and Cr coding blocks of the CU. For example, reconstruction unit 158 may add samples of the luma, Cb, and Cr transform blocks to corresponding samples of the predictive luma, Cb, and Cr blocks to reconstruct the luma, Cb, and Cr coding blocks of the CU.

[0180] Filter unit 160 may perform a deblocking operation to reduce blocking artifacts associated with the luma, Cb, and Cr coding blocks of the CU. Video decoder 30 may store the luma, Cb, and Cr coding blocks of the CU in decoded picture buffer 162. Decoded picture buffer 162 may provide reference pictures for subsequent motion compensation, intra prediction, and presentation on a display device, such as display device 32 of FIG. 1. For instance, video decoder 30 may perform, based on the luma, Cb, and Cr blocks in decoded picture buffer 162, intra prediction or inter prediction operations on PUs of other CUs. In this way, video decoder 30 may extract, from the bitstream, transform coefficient levels of the significant luma coefficient block, inverse quantize the transform coefficient levels, apply a transform to the transform coefficient levels to generate a transform block, generate, based at least in part on the transform block, a coding block, and output the coding block for display.

[0181] Video decoder 30 is an example of a video decoder configured to perform any of the techniques described in this disclosure. In accordance with one or more techniques of this disclosure, one or more units within video decoder 30 may perform the techniques described herein as part of a video decoding process. For example, intra prediction processing unit 166 and/or entropy decoding unit 150 may be configured to perform the techniques described herein as part of a video decoding process. In some examples, intra prediction processing unit 166 may be configured to perform the techniques described herein for deriving or selecting intra prediction modes and/or MPMs for intra prediction, the techniques using an increased number of MPMs and/or angular modes, and the techniques for applying an N-tap Intra interpolation filter, where N is larger than 2. Intra prediction processing unit 166 may be configured to perform the techniques described herein for allowing multiple intra prediction directions per block, e.g. respective directions for sub-blocks of the block, which may not require additional signaling of multiple intra prediction modes.

[0182] Intra prediction processing unit 166 may receive syntax information from entropy decoding unit 150, such as a selected intra prediction mode and/or MPM index for intra prediction. Intra prediction processing unit 166 and/or entropy decoding unit 150 may be configured to implement the techniques described herein for decoding the syntax information related to intra prediction, e.g., the selected intra prediction mode and/or MPM index for intra prediction, from an encoded video bitstream.

[0183] In following subsections, examples implementations of particular techniques of this disclosure are be provided. In practice, any combination of any part of the examples may be used as a new example. The example implementations are shown in the form of edits to the JCTVC-N1003. In the following, text added to JCTVC-N1003 is shown between tags <ins> and </ins>, "ins" being short for "insert". Text deleted from JCTVC-N1003 is shown between tags <dlt> and </dlt>, "dlt" being short for "delete."

[0184] Several examples were described above in which video encoder 20 and video decoder 30 use more than three MPMs for signaling an intra prediction mode of a PU. In some such examples, video encoder 20 and video decoder 30 use CABAC coding to encode and decode MPM indexes. As part of CABAC encoding, video encoder 20 applies a binarization process to an MPM index to convert the MPM index into a binary code and video decoder 30 de-binarizes the binary code to recover the MPM index. In accordance with a technique of this disclosure, as shown in Table 9-32 below, the MPM index (mpm_idx) is binarized using a truncated Rice binarization process (TR) and the input parameter cMax is changed from 2 to 5 to accommodate larger values of the MPM index. Section 9.3.3.2 of JCTVC-N1003 describes an implementation of the Truncated Rice binarization process.

Table 9-32 - Syntax elements and associated binarizations

Syntax element	Binarization	
	Process	Input parameters
...	...	...
cu_transquant_bypass_flag	FL	cMax = 1
cu_skip_flag	FL	cMax = 1
pred_mode_flag	FL	cMax = 1
part_mode	9.3.3.5	(xCb, yCb) = (x0, y0), log2CbSize
pcm_flag[ ][ ]	FL	cMax = 1
prev_intra_luma_pred _flag[ ][ ]	FL	cMax = 1
mpm_idx[ ][ ]	TR	cMax = <dlt>2</dlt><ins>5</ins>, cRiceParam = 0
rem_intra_luma_pred _mode[ ][ ]	FL	cMax = 31
intra_chroma_pred_m ode[ ][ ]	9.3.3.6	-
rqt_root_cbf	FL	cMax = 1

[0185] Furthermore, as described above, video encoder 20 and video decoder 30 may use context modeling when coding (e.g., CABAC coding) one or more bins of an MPM index. In JCTVC-N1003, a binarized MPM index (i.e., mpm_idx) consists of three bins. However, in this disclosure, with larger numbers of MPMs, the binarized MPM index may include five or more bins. As specified in subclause 9.3.4.2.1 of JCTVC-N1003, a variable ctxInc is specified by the corresponding entry in Table 9-37 and when more than one value is listed in Table 9-37 for a binIdx, the assignment process for ctxInc for that binIdx is further specified in the subclauses given in parenthesis. As further described in subclause 9.3.4.2.1. of JCTVC-N1003, ctxInc is used to determine a value of ctxIdx, which is an index of a coding context. As shown in the edited version of Table 9-37 below, the first three bins of the MPM index may be coded using coding contexts selected using the context modeling methods of the added subclause 9.3.4.x.x below and remaining bins of the MPM index may be coded using bypass coding.

Table 9-37 - Assignment of ctxInc to syntax elements with context coded bins

Syntax element	binIdx					
	0	1	2	3	4	>= 5
...	...	...	...	...	...	...
pcm_flag [ ][ ]	terminate	na	na	na	na	na
prev_intra_luma_pred_flag[ ] [ ]	0	na	Na	na	na	na
mpm_idx [ ][ ]	<dlt> bypass </dlt> <ins>0,1,2 (subclause 9.3.4.x.x) </ins>	<del> bypass </del> <ins>3,4,5 (subclause 9.3.4.x.x) </ins>	<del> bypass </del> <ins>7,8 (subclause 9.3.4.x.x) </ins>	<dlt> na </dlt> <ins> bypass </ins>	<dlt> na </dlt> <ins> bypass </ins>	<dlt> na </dlt> <ins> bypass </ins>
rem_intra_luma_pred_mode [ ][ ]	bypass	bypass	bypass	bypass	bypass	bypass
intra_chroma_pred_mode[ ] [ ]	0	bypass	bypass	na	na	na
...	...	...	...	...	...	...

[0186] <ins>**9.3.3.x Binarization process for rem_intra_luma_pred_mode** Input to this process is a request for a binarization for the syntax element rem_intra_luma_pred_mode.

[0187] Output of this process is the binarization of the syntax element.

- If rem_intra_luma_pred_mode equals to 28, the binarization string is "111";

- Otherwise, the binarization is fixed-length code with 5 bins.

9.3.4.x.x Derivation process of ctxInc for the syntax element mpm_idx

[0188] Inputs to this process are the intra modes of the left neighboring block candIntraPredModeA, and the above neighboring block candIntraPredModeB.

[0189] Output of this process is the variable ctxInc.

[0190] The variable csbfCtx is derived using the current location (xS, yS), two previously decoded bins of the syntax element coded_sub_block_flag in scan order, and the transform block size log2TrafoSize, as follows:

- When candIntraPredModeA equals to candIntraPredModeB, ctxInc is derived as follows
  ctxInc = candIntraPredModeA > 1 ? 0 : 1

- Otherwise
  ctxInc = (candIntraPredModeA && candIntraPredModeB)? 2 : 3</ins>

[0191] In accordance with some examples of this disclosure, more than three MPMs are defined. The following text describes example changes to JCTVC-N1003 to implement six MPMs.

8.4.2 Derivation process for luma intra prediction mode

[0192] Input to this process is a luma location (xPb, yPb) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture<ins>, a prediction block size nPbS</ins>.

[0193] In this process, the luma intra prediction mode IntraPredModeY[xPb][yPb] is derived. Table 8-1 specifies the value for the intra prediction mode and the associated names.

Table 8-1 - Specification of intra prediction mode and associated names

Intra prediction mode	Associated name
0	INTRA_PLANAR
1	INTRA_DC
2..34	INTRA_ANGULAR2..INTRA_ANGULAR34

[0194] IntraPredModeY[xPb][yPb] labelled 0..34 represents directions of predictions as illustrated in Figure 8-1, which is FIG. 8 of this disclosure.
IntraPredModeY[xPb][yPb] is derived by the following ordered steps:

4. The neighbouring locations (xNbA, yNbA) and (xNbB, yNbB) are set equal to (xPb - 1, yPb) and (xPb, yPb - 1), respectively.
5. For X being replaced by either A or B, the variables candIntraPredModeX are derived as follows:
<ins>

- candIntraPredModeX is initialized as INTRA_DC.
- Initialize the counts of intra mode usage as 0: cntIntraPredModeX[i]=0, i=0, 1, ...,34
- Initalize the maximum count of intra mode usage as 0:
  cntMaxIntraPredModeX = 0;
- For x = 0..nPbS - 1, the following applies
  - (xCurr, yCurr) = (X==A) ? (xNbA+x, yNbA) ? (xNbA, yNbA +x)
  - The availability derivation process for a block in z-scan order as specified in subclause 6.4.1 is invoked with the location (xCurr, yCurr) set equal to (xPb, yPb) and the neighbouring location (xNbY, yNbY) set equal to (xNbX, yNbX) as inputs, and the output is assigned to availableX.
  - If availableX is equal to TRUE and CuPredMode[xNbX][yNbX] 1 is equal to MODE INTRA, the following applies.
    - cntIntraPredModeX[CuPredMode[xCurr][yCurr]] ++
    - if cntIntraPredModeX[CuPredMode[xCurr][yCurr]] > cntMaxIntraPredModeX, the following applies:
      - cntMaxIntraPredModeX = cntIntraPredModeX[CuPredMode[xCurr][yCurr]];
      - candIntraPredModeX = CuPredMode[xCurr][yCurr]
        </ins>

<dlt>

- The availability derivation process for a block in z-scan order as specified in subclause 6.4.1 is invoked with the location (xCurr, yCurr) set equal to (xPb, yPb) and the neighbouring location (xNbY, yNbY) set equal to (xNbX, yNbX) as inputs, and the output is assigned to availableX.

- The candidate intra prediction mode candIntraPredModeX is derived as follows:
  - If availableX is equal to FALSE, candIntraPredModeX is set equal to INTRA_DC.
  - Otherwise, if CuPredMode[xNbX][yNbX] is not equal to MODE_INTRA or pcm_flag[xNbX][yNbX] is equal to 1, candIntraPredModeX is set equal to INTRA_DC,
  - Otherwise, if X is equal to B and yPb - 1 is less than ((yPb >> CtbLog2SizeY) << CtbLog2SizeY), candIntraPredModeB is set equal to INTRA_DC.
  - Otherwise, candIntraPredModeX is set equal to IntraPredModeY[xNbX][yNbX].</dlt>

6. The candModeList[x] with x = 0..<dlt> 2</dlt><ins>5</ins> is derived as follows:

- <ins> idxPlanar is set as 0, idxDC is set as 1;</ins>
- If candIntraPredModeB is equal to candIntraPredModeA, the following applies:
  - If candIntraPredModeA is less than 2 (i.e. equal to INTRA_PLANAR or INTRA DC), candModeList[x] with x = 0..<dlt>2</dlt><ins>5</ins> is derived as follows:

$\text{candModeList}[\,0\,] = \text{INTRA\_PLANAR}$ (8-15)

$\text{candModeList}[\,1\,] = \text{INTRA\_DC}$ (8-16)

$\text{candModeList}[\,2\,] = \text{INTRA\_ANGULAR26}$ (8-17)

<ins>$\text{candModeList}[\,3\,] = \text{INTRA\_ANGULAR10}$ (8-17)

$\text{candModeList}[\,4\,] = \text{INTRA\_ANGULAR2}$ (8-17)

$\text{candModeList}[\,5\,] = \text{INTRA\_ANGULAR18}$ (8-17)</ins>

</ins>

  - Otherwise, candModeList[x] with x = 0..5 is derived as follows:

$\text{candModeList}[\,0\,] = \text{candIntraPredModeA}$ (8-18)

<dlt>$\text{candModeList}[\,2\,] = 2 + (\,(\,\text{candIntraPredModeA} + 29\,)\,\%\,32\,)$ (8-19)

$\text{candModeList}[\,2\,] = 2 + (\,(\,\text{candIntraPredModeA} - 2 + 1\,)\,\%\,32\,)$ (8-20)</dlt>

</dlt>

<ins>$\text{candModeList}[\,1\,] = \text{INTRA\_PLANAR}$ (8-20)

$$\text{candModeList[2]} = 2 + (\ (\ \text{candIntraPredModeA} - 1\ )\ \%\ 32\ ) \qquad (8\text{-}19)$$

$$\text{candModeList[3]} = 2 + (\ (\ \text{candIntraPredModeA} + 29\ )\ \%\ 32\ ) \qquad (8\text{-}20)$$

$$\text{candModeList[4]} = 2 + (\ (\ \text{candModeList[2]} - 1\ )\ \%\ 32\ ) \qquad (8\text{-}20)$$

$$\text{candModeList[5]} = \text{INTRA\_DC} \qquad (8\text{-}20)$$

idxPlanar = 1
idxDC = 5</ins>

- Otherwise (candIntraPredModeB is not equal to candIntraPredModeA), the following applies:
    - candModeList[0] and candModeList[1] are derived as follows:

$$\text{candModeList[0]} = \text{candIntraPredModeA} \qquad (8\text{-}21)$$

$$\text{candModeList[1]} = \text{candIntraPredModeB} \qquad (8\text{-}22)$$

- If neither of candModeList[0] and candModeList[1] is equal to INTRA_PLANAR, candModeList[2] is set equal to INTRA_PLANAR, <ins>and the following applies:
    - maxDir = max(candModeList[0], candModeList[1])
    - minDir = min(candModeList[0], candModeList[1])
    - idxPlanar = 2
    - If either of candModeList[0] and candModeList[1] is equal to INTRA_DC, the following applies:

$$\text{candModeList[3]} = 2 + (\ (\ \text{maxDir} + 29\ )\ \%\ 32\ ) \qquad (8\text{-}21)$$

$$\text{candModeList[4]} = 2 + (\ (\ \text{maxDir} - 1\ )\ \%\ 32\ ) \qquad (8\text{-}22)$$

$$\text{candModeList[5]} = 2 + (\ (\ \text{candModeList[4]} - 1\ )\ \%\ 32\ ) \qquad (8\text{-}22)$$

idxDC = (candModeList[0]== INTRA_DC) ? 0 : 1

- Otherwise,

$$\text{candModeList[3]} = \text{INTRA\_DC} \qquad (8\text{-}21)$$

$$\text{candModeList[4]} = 2 + (\ (\ \text{maxDir} - 1\ )\ \%\ 32\ ) \qquad (8\text{-}22)$$

- If candModeList[4] equals to minDir, candModeList[4]++ candModeList[5] = 2 + ((candModeList[4] + 29) % 32) (8-22)
- If candModeList[5] equals to maxDir, candModeList[5]--
- If candModeList[5] equals to candModeList[4], candModeList[5]= minDir+1
  idxDC = 3</ins>

- Otherwise, if neither of candModeList[0] and candModeList[1] is equal to INTRA DC, candModeList[2] is set equal to INTRA_DC, <ins>the following applies:

  candModeList[3] = 2 + ((candIntraPredModeA + 29) % 32) candModeList[4] = 2 + ((candIntraPredModeA - 1) % 32) candModeList[5] = 2 + ((candModeList[4] - 1) % 32) idxPlanar = (candModeList[0]== INTRA_PLANAR) ? 0 : 1 idxDC = 2</ins>

- Otherwise, candModeList[2] is set equal to INTRA_ANGULAR26, the following applies: <ins>candModeList[3] = INTRA_ANGULAR10 candModeList[4] = INTRA_ANGULAR2 candModeList[5] = INTRA_ANGULAR18 idxPlanar = (candModeList[0]== INTRA_PLANAR) ? 0 : 1 idxDC = 1- idxPlanar</ins>

7. IntraPredModeY[xPb][yPb] is derived by applying the following procedure:

- If prev_intra_luma_pred_flag[xPb][yPb] is equal to 1, the IntraPredModeY[xPb][yPb] is set equal to candModeList[mpm_idx].
- Otherwise, IntraPredModeY[xPb][yPb] is derived by applying the following ordered steps:

  1) The array candModeList[x], x = 0...<dlt>2</dlt><ins>5</ins> is modified as the following ordered steps:<ins>

  i. candModeList[idxPlanar] = candModeList[0]
  ii. candModeList[idxDC] = candModeList[1]
  iii. candModeList[0] = INTRA PLANAR
  iv. candModeList[1] = INTRA_DC</ins>
  v. When candModeList[<dlt>0</dlt><ins>2</ins>] is greater than candModeList[<dlt>1</dlt><ins>3</ins>], both values are swapped as follows: (candModeList[<dlt>0</dlt><ins>2</ins>], candModeList[<dlt>1</dlt><ins>3</ins>]) = Swap(candModeList[<dlt>0</dlt><ins>2</ins>], candModeList[<dlt>1</dlt><ins>3</ins>]) (8-23)
  vi. When candModeList[<dlt>0</dlt><ins>2</ins>] is greater than candModeList[<dlt>2</dlt><ins>4</ins>], both values are swapped as follows: (candModeList[<dlt>0</dlt><ins>2</ins>], candModeList[<dlt>2</dlt><ins>4</ins>]) = Swap(candModeList[<dlt>0</dlt><ins>2</ins>], candModeList[<dlt>2</dlt><ins>4</ins>]) (8-24)
  vii. <ins>When candModeList[2 1 is greater than candModeList[5], both values are swapped as follows: (candModeList[2], candModeList[5]) = Swap(candModeList[2], candModeList[5])</ins>
  viii. When candModeList[<dlt>1</dlt><ins>3</ins>] is greater than candModeList[<dlt>2</dlt><ins>4</ins>], both values are swapped as follows: (candModeList[<dlt>1</delete ><ins>3</ins>], candModeList[<dlt>2</dlt><ins>4</ins>]) = Swap(candModeList[<dlt>1</dlt><ins>3</ins>], candModeList[<dlt>2</dlt><ins>4</ins>]) (8-25)
  ix. <ins>When candModeList[3] is greater than candModeList[5], both values are swapped as follows: (candModeList[3], candModeList[5]) = Swap(candModeList[3], candModeList[5])
  x. When candModeList[4] is greater than candModeList[5], both values are swapped as follows: (candModeList[4], candModeList[5]) = Swap(candModeList[4], candModeList[5])</ins>

  2) IntraPredModeY[xPb][yPb] is derived by the following ordered steps:

  i. IntraPredModeY[xPb][yPb] is set equal to rem_intra_luma_pred_mode[xPb][yPb].
  ii. For i equal to 0 to <dlt>2</dlt><ins>5</ins>, inclusive, when IntraPredModeY[xPb][yPb] is greater than or equal to candModeList[i], the value of IntraPredModeY[xPb][yPb] is incremented by one.

[0195] In the example adapted version of subclause 8.4.2 of JCTVC-N1003 shown above, the derivation process for the MPMs of the current PU includes a derivation process for a representative intra prediction mode (candIntraPredModeA) for a left neighboring column and a representative intra prediction mode (candIntraPredModeB) for an above neighboring row. In this example, is left neighboring column corresponds to samples (xNbA, yNbA + x) for x = x..nPbs and the above neighboring row corresponds to samples (xNbA+x, yNbA) for x = 0..nPbs), where nPbs is the width and height of the prediction block of the current PU.

[0196] Furthermore, in the example adapted version of subclause 8.4.2 of JCTVC-N1003 shown above, the list of MPM candidates (i.e., candModeList) always includes the planar, DC, vertical (i.e., INTRA_ANGULAR26), horizontal (i.e., INTRA_ANGULAR10), and diagonal (i.e., INTRA_ANGULAR18) intra prediction modes if candIntraPredModeA is the same as candIntraPredModeB and candIntraPredModeA is planar or DC. However, in this example, if candIntraPredModeA is not planar and not DC, the list of MPM candidates includes candIntraPredModeA, the planar intra prediction, three angular intra prediction modes closest to candIntraPredModeA, and the DC intra prediction mode. The three angular intra prediction modes closest to candIntraPredModeA are calculated as:

$$2 + ((\text{candIntraPredModeA} - 1) \% 32),$$

$$(2 + ((\text{candIntraPredModeA} + 29) \% 32),$$

and

$$2 + ((\text{candModeList}[2]- 1) \% 32)$$

In the formulae above, % denotes the modulo operation.

[0197] Similarly, in the example adapted version of subclause 8.4.2 of JCTVC-N1003 shown above, if candIntraPredModeA is not equal to candIntraPredModeB, the video coder may include in the list of MPM candidates, other angular intra prediction modes determined to be closest either candIntraPredModeA or candIntraPredModeB using the formulae shown in the example adapted version of subclause 8.4.2 of JCTVC-N1003 that include modulo operations.

[0198] As described above, in some examples of this disclosure, video encoder 20 and video decoder 30 may use more than 33 angular intra prediction modes. The following text describes examples changes to JCTVC-N1003 to implement 65 angular intra prediction modes.

8.4.4.2.6 Specification of intra prediction mode in the range of INTRA_ANGULAR2.. INTRA_ANGULAR34

[0199] Inputs to this process are:

- the intra prediction mode predModeIntra,
- the neighbouring samples p[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1,
- a variable nTbS specifying the transform block size,
- a variable cIdx specifying the colour component of the current block.

[0200] Outputs of this process are the predicted samples predSamples[x][y], with x, y = 0..nTbS - 1.

[0201] FIG. 8-2, which is FIG. 9 of this disclosure, illustrates the total 33 intra angles and Table 8-4 specifies the mapping table between predModeIntra and the angle parameter intraPredAngle.

<dlt>**Table 8-4 - Specification of intraPredAngle**

predMode Intra	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
intraPred Angle	-	32	26	21	17	13	9	5	2	0	-2	-5	-9	-13	-17	-21	-26
predMode Intra	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**
intraPred Angle	-32	-26	-21	-17	-13	-9	-5	-2	0	2	5	9	13	17	21	26	32

</dlt>

<ins>**Table 8-4 - Specification of intraPredAngle**

predMode Intra	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
intraPred Angle	-	-	32	29	26	23	21	19	17	15	13	11	9	7	5	3	2
predMode Intra	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**
intraPred Angle	1	0	1	2	3	5	7	9	11	13	15	17	19	21	23	26	29
predMode Intra	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
intraPred Angle	32	29	26	23	21	19	17	15	13	11	9	7	5	3	2	1	0

(continued)

predMode Intra	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
intraPred Angle	1	2	3	5	7	9	11	13	15	17	19	21	23	26	29	32	-

</ins>

[0202] Table 8-5 further specifies the mapping table between predModeIntra and the inverse angle parameter invAngle.

Table 8-5 - Specification of invAngle

predModeIntra	11	12	13	14	15	16	17	18
invAngle	-4096	-1638	-910	-630	-482	-390	-315	-256
predModeIntra	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
invAngle	-315	-390	-482	-630	-910	-1638	-4096	-

[0203] The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

- If predModeIntra is equal or greater than 18, the following ordered steps apply:

1. The reference sample array ref[x] is specified as follows:

- The following applies:

$$\text{ref}[\ x\ ] = \text{p}[\ -1 + x\ ][\ -1\ ],\ \text{with}\ x = 0..\text{nTbS} \qquad (8\text{-}47)$$

- If intraPredAngle is less than 0, the main reference sample array is extended as follows:

- When (nTbS * intraPredAngle) » 5 is less than -1,

$$\text{ref}[\ x\ ] = \text{p}[\ -1\ ][\ -1 + (\ (\ x * \text{invAngle} + 128\ )\ >>\ 8\ )\ ],$$
$$\text{with}\ x = -1..(\ \text{nTbS} * \text{intraPredAngle}\ )\ >>\ 5 \qquad (8\text{-}48)$$

- Otherwise,

$$\text{ref}[\ x\ ] = \text{p}[\ -1 + x\ ][\ -1\ ],\ \text{with}\ x = \text{nTbS} + 1..2 * \text{nTbS} \qquad (8\text{-}49)$$

2. The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

a. The index variable iIdx and the multiplication factor iFact are derived as follows:

$$\text{iIdx} = (\ (\ y + 1\ ) * \text{intraPredAngle}\ )\ >>\ 5 \qquad (8\text{-}50)$$

$$\text{iFact} = (\ (\ y + 1\ ) * \text{intraPredAngle}\ )\ \&\ 31 \qquad (8\text{-}51)$$

b. Depending on the value of iFact, the following applies:

- If iFact is not equal to 0, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\ x\ ][\ y\ ] = ((32 - \text{iFact}) * \text{ref}[\ x + \text{iIdx} + 1\ ] + \text{iFact} * \text{ref}[\ x + \text{iIdx} + 2\ ] + 16) >> 5 \qquad (8\text{-}52)$$

- Otherwise, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\ x\ ][\ y\ ] = \text{ref}[\ x + \text{iIdx} + 1\ ] \qquad (8\text{-}53)$$

c. When predModeIntra is equal to 26 (vertical), cIdx is equal to 0 and nTbS is less than 32, the following filtering applies with x = 0, y = 0..nTbS - 1:

$$\text{predSamples}[\ x\ ][\ y\ ] = \text{Clip1}_{\text{Y}}(\ p[\ x\ ][\ -1\ ] + ((\ p[\ -1\ ][\ y\ ] - p[\ -1\ ][\ -1\ ]\ ) >> 1\ )) \qquad (8\text{-}54)$$

- Otherwise (predModeIntra is less than 18), the following ordered steps apply:

1. The reference sample array ref[x] is specified as follows:

- The following applies:

$$\text{ref}[\ x\ ] = p[\ -1\ ][\ -1 + x\ ], \text{ with } x = 0..\text{nTbS} \qquad (8\text{-}55)$$

- If intraPredAngle is less than 0, the main reference sample array is extended as follows:

- When (nTbS * intraPredAngle) >> 5 is less than -1,

$$\text{ref}[\ x\ ] = p[\ -1 + ((\ x * \text{invAngle} + 128\ ) >> 8\ )][\ -1\ ], \text{ with } x = -1..(\ \text{nTbS} * \text{intraPredAngle}\ ) >> 5 \qquad (8\text{-}56)$$

- Otherwise,

$$\text{ref}[\ x\ ] = p[\ -1\ ][\ -1 + x\ ], \text{ with } x = \text{nTbS} + 1..2 * \text{nTbS} \qquad (8\text{-}57)$$

2. The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

a. The index variable iIdx and the multiplication factor iFact are derived as follows:

$$\text{iIdx} = ((\ x + 1\ ) * \text{intraPredAngle}\ ) >> 5 \qquad (8\text{-}58)$$

$$\text{iFact} = ((\ x + 1\ ) * \text{intraPredAngle}\ ) \ \&\ 31 \qquad (8\text{-}59)$$

b. Depending on the value of iFact, the following applies:

- If iFact is not equal to 0, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\,x\,][\,y\,] = ((\,32 - \text{iFact}\,) * \text{ref}[\,y + \text{iIdx} + 1\,] + \text{iFact} * \text{ref}[\,y + \text{iIdx} + 2\,] + 16\,) >> 5 \qquad (8\text{-}60)$$

- Otherwise, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\,x\,][\,y\,] = \text{ref}[\,y + \text{iIdx} + 1\,] \qquad (8\text{-}61)$$

c. When predModeIntra is equal to 10 (horizontal), cIdx is equal to 0 and nTbS is less than 32, the following filtering applies with x = 0..nTbS - 1, y = 0:

$$\text{predSamples}[\,x\,][\,y\,] = \text{Clip1}_{Y}(\,p[\,-1\,][\,y\,] + (\,(\,p[\,x\,][\,-1\,] - p[\,-1\,][\,-1\,]\,) >> 1\,)\,) \qquad (8\text{-}62)$$

[0204] Table 8-5 further specifies the mapping table between predModeIntra and the inverse angle parameter invAngle.

Table 8-5 - Specification of invAngle

predModeIntra	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
invAngle	-4096	-1638	-910	-630	-482	-390	-315	-256
predModeIntra	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
invAngle	-315	-390	-482	-630	-910	-1638	-4096	-

[0205] The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

- If predModeIntra is equal or greater than 18, the following ordered steps apply:

7.4.9.11 Residual coding semantics

[0206] For intra prediction, different scanning orders are used. The variable scanIdx specifies which scan order is used where scanIdx equal to 0 specifies an up-right diagonal scan order, scanIdx equal to 1 specifies a horizontal scan order, and scanIdx equal to 2 specifies a vertical scan order. The value of scanIdx is derived as follows:

- If CuPredMode[x0][y0] is equal to MODE_INTRA and one or more of the following conditions are true:
  - log2TrafoSize is equal to 2.
  - log2TrafoSize is equal to 3 and cIdx is equal to 0.

  predModeIntra is derived as follows:

  - If cIdx is equal to 0, predModeIntra is set equal to IntraPredModeY[x0][y0].
  - Otherwise, predModeIntra is set equal to IntraPredModeC.

  scanIdx is derived as follows:

  - If predModeIntra is in the range of <dlt>6</dlt><ins>10</ins> to <dlt>14</dlt><ins>26</ins>, inclusive, scanIdx is set equal to 2.
  - Otherwise if predModeIntra is in the range of <dlt>22</dlt><ins>42</ins> to <dlt>30</dlt><ins>58</ins>, inclu-

sive, scanIdx is set equal to 1.

  - Otherwise, scanIdx is set equal to 0.

- Otherwise, scanIdx is set equal to 0.

8.4.4.2.3 Filtering process of neighbouring samples

[0207] Inputs to this process are:

- the neighbouring samples p[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1,

- a variable nTbS specifying the transform block size.

[0208] Outputs of this process are the filtered samples pF[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1.

[0209] The variable filterFlag is derived as follows:

- If one or more of the following conditions are true, filterFlag is set equal to 0:

  - predModeIntra is equal to INTRA_DC.

  - nTbS is equal 4.

- Otherwise, the following applies:

  - The variable minDistVerHor is set equal to Min(Abs(predModeIntra - 26), Abs(predModeIntra - 10)).

  - The variable intraHorVerDistThres[nTbS] is specified in Table 8-3.

  - The variable filterFlag is derived as follows:

    - If minDistVerHor is greater than intraHorVerDistThres[nTbS], filterFlag is set equal to 1.

    - Otherwise, filterFlag is set equal to 0.

7.4.9.11 Residual coding semantics

[0210] For intra prediction, different scanning orders are used. The variable scanIdx specifies which scan order is used where scanIdx equal to 0 specifies an up-right diagonal scan order, scanIdx equal to 1 specifies a horizontal scan order, and scanIdx equal to 2 specifies a vertical scan order. The value of scanIdx is derived as follows:

- If CuPredMode[x0][y0] is equal to MODE_INTRA and one or more of the following conditions are true:

  - log2TrafoSize is equal to 2.

  - log2TrafoSize is equal to 3 and cIdx is equal to 0.

  predModeIntra is derived as follows:

  - If cIdx is equal to 0, predModeIntra is set equal to IntraPredModeY[x0][y0].

  - Otherwise, predModeIntra is set equal to IntraPredModeC.

  scanIdx is derived as follows:

  - If predModeIntra is in the range of <dlt>6</dlt><ins>10</ins> to <dlt>14</dlt><ins>26</ins>, inclusive, scanIdx is set equal to 2.

    - Otherwise if predModeIntra is in the range of <dlt>22</dlt><ins>42</ins> to <dlt>30</dlt><ins>58</ins>, inclusive, scanIdx is set equal to 1.
    - Otherwise, scanIdx is set equal to 0.
- Otherwise, scanIdx is set equal to 0.

8.4.4.2.3 Filtering process of neighbouring samples

[0211] Inputs to this process are:

- the neighbouring samples p[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1,
- a variable nTbS specifying the transform block size.

[0212] Outputs of this process are the filtered samples pF[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1.

[0213] The variable filterFlag is derived as follows:

- If one or more of the following conditions are true, filterFlag is set equal to 0:
    - predModeIntra is equal to INTRA_DC.
    - nTbS is equal 4.
- Otherwise, the following applies:
    - The variable minDistVerHor is set equal to Min(Abs(predModeIntra - 26), Abs(predModeIntra - 10)).
    - The variable intraHorVerDistThres[nTbS] is specified in Table 8-3.
    - The variable filterFlag is derived as follows:
        - If minDistVerHor is greater than intraHorVerDistThres[nTbS], filterFlag is set equal to 1.
        - Otherwise, filterFlag is set equal to 0.

Table 8-3 - Specification of intraHorVerDistThres[nTbS] for various transform block sizes

	nTbS = 8	**nTbS = 16**	**nTbS = 32**
intraHorVerDistThres [nTbS]	<dlt>7 </dlt> <ins>14</ins>	<dlt>1 </dlt> <ins>2 </ins>	0

[0214] When filterFlag is equal to 1, the following applies:

[0215] As described above, in accordance with some techniques of this disclosure, video encoder 20 and video decoder 30 may apply an *N*-tap intra interpolation filter. The following text describes example changes to JCTVC-N1003 to implement application of a 4-tap intra interpolation filter.

<ins>**x.x.x Intra interpolation filter coefficients initialization process**</ins>

[0216] Output of this process is the array filterCubic[sFrac][pos]. The array index sFrac specifies the fractional position ranging from 0 to 31, pos specifies the filter coefficient for the posth sample. The array filterCubic and filterGaussian is derived as follows:

```
<ins>filterCubic [32][4] = {
 {
  {   0, 256,   0,   0 }, //  0
  {  -3, 252,   8,  -1 }, //  1
  {  -5, 247,  17,  -3 }, //  2
  {  -7, 242,  25,  -4 }, //  3
  {  -9, 236,  34,  -5 }, //  4
  { -10, 230,  43,  -7 }, //  5
```

```
    { -12, 224,  52,  -8 }, //  6
    { -13, 217,  61,  -9 }, //  7
    { -14, 210,  70, -10 }, //  8
    { -15, 203,  79, -11 }, //  9
    { -16, 195,  89, -12 }, // 10
    { -16, 187,  98, -13 }, // 11
    { -16, 179, 107, -14 }, // 12
    { -16, 170, 116, -14 }, // 13
    { -17, 162, 126, -15 }, // 14
    { -16, 153, 135, -16 }, // 15
    { -16, 144, 144, -16 }, // 16
  },
};

sigma = 0.9

for( i=0; i<17; i++ )
{
  for( c=0; c<4; c++ )
  {
     filterGaussian [b][i][c] = ( 256.0 * exp(-((c-delta)/sigma)^2) / sum + 0.5 );
  }
}


for( i=17; i<32; i++ )
{
  for( c=0; c<4; c++ )
  {
     filterCubic [b][i][c] = filterCubic [b][32-i][3-c];
     filterGaussian [b][i][c] = filterGaussian [b][32-i][3-c];
  }
}
</ins>
```

8.4.4.2.6 Specification of intra prediction mode in the range of INTRA ANGULAR2.. INTRA ANGULAR34

[0217] Inputs to this process are:

- the intra prediction mode predModeIntra,
- the neighbouring samples p[x][y], with x = -1, y = -1..nTbS * 2 - 1 and x = 0..nTbS * 2 - 1, y = -1,
- a variable nTbS specifying the transform block size,
- a variable cIdx specifying the colour component of the current block.

[0218] Outputs of this process are the predicted samples predSamples[x][y], with x, y = 0..nTbS - 1.
[0219] The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

- If predModeIntra is equal or greater than 18, the following ordered steps apply:

3. The reference sample array ref[x] is specified as follows:

- The following applies:

$$\text{ref}[\ x\ ] = \text{p}[\ -1 + x\ ][\ -1\ ],\ \text{with}\ x = 0..\text{nTbS} \qquad (8\text{-}47)$$

- If intraPredAngle is less than 0, the main reference sample array is extended as follows:
  - When (nTbS * intraPredAngle) >> 5 is less than -1,

$$\text{ref}[\ x\ ] = \text{p}[\ -1\ ][\ -1 + (\ (\ x * \text{invAngle} + 128\ ) >> 8\ )\ ],\ \text{with}\ x = -1..(\ \text{nTbS} * \text{intraPredAngle}\ ) >> 5 \qquad (8\text{-}48)$$

- Otherwise,

$$\text{ref}[\ x\ ] = \text{p}[\ -1 + x\ ][\ -1\ ],\ \text{with}\ x = \text{nTbS} + 1..2 * \text{nTbS} \qquad (8\text{-}49)$$

4. The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

a. The index variable iIdx and the multiplication factor iFact are derived as follows:

$$\text{iIdx} = (\ (\ y + 1\ ) * \text{intraPredAngle}\ ) >> 5 \qquad (8\text{-}50)$$

$$\text{iFact} = (\ (\ y + 1\ ) * \text{intraPredAngle}\ )\ \&\ 31 \qquad (8\text{-}51)$$

b. Depending on the value of iFact, the following applies:

- If iFact is not equal to 0, the value of the prediction samples predSamples[x][y] is derived as follows:

<dlt>$$\text{predSamples}[\ x\ ][\ y\ ] = (\ (\ 32 - \text{iFact}\ ) * \text{ref}[\ x + \text{iIdx} + 1\ ] + \text{iFact} * \text{ref}[\ x + \text{iIdx} + 2\ ] + 16\ ) >> 5 \qquad (8\text{-}52)$$</dlt>

<ins>For p=0,..,3, pF[p] = (cIdx==0 && nTbS<=8) ? filterCubic[iFact][p] : filterGaussian[iFact][p]

$$P[1] = \text{ref}[\ x + \text{iIdx} + 1\ ]$$

$$P[2] = \text{ref}[\ x + \text{iIdx} + 2\ ]$$

$$P[0] = (\ x{==}0)\ ?\ \text{ref}[\ x + \text{iIdx} + 1\ ]\ :\ \text{ref}[\ x + \text{iIdx}\ ]$$

$$P[3] = (\ x{==}\text{nTbS-1})\ ?\ \text{ref}[\ x + \text{iIdx} + 2\ ]\ :\ \text{ref}[\ x + \text{iIdx} + 3]$$

$$\text{predSamples}[\ x\ ][\ y\ ] = (\ \text{pF}[0] * P[0] + \text{pF}[1] * P[1] + \text{pF}[2] * P[2] + \text{pF}[3] * P[3] + 128) >> 8 \qquad (8\text{-}52)$$</ins>

- Otherwise, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\ x\ ][\ y\ ] = \text{ref}[\ x + \text{iIdx} + 1\ ] \qquad (8\text{-}53)$$

c. When predModeIntra is equal to 26 (vertical), cIdx is equal to 0 and nTbS is less than 32, the following filtering applies with x = 0, y = 0..nTbS - 1:

$$\text{predSamples}[\ x\ ][\ y\ ] = \text{Clip1}_{\text{Y}}(\ p[\ x\ ][\ -1\ ] + (\ (\ p[\ -1\ ][\ y\ ] - p[\ -1\ ][\ -1\ ]\ )\ >>\ 1\ )\ ) \qquad (8\text{-}54)$$

- Otherwise (predModeIntra is less than 18), the following ordered steps apply:

3. The reference sample array ref[x] is specified as follows:

- The following applies:

$$\text{ref}[\ x\ ] = p[\ -1\ ][\ -1 + x\ ],\ \text{with}\ x = 0..\text{nTbS} \qquad (8\text{-}55)$$

- If intraPredAngle is less than 0, the main reference sample array is extended as follows:

  - When (nTbS * intraPredAngle) >> 5 is less than -1,

$$\text{ref}[\ x\ ] = p[\ -1 + (\ (\ x * \text{invAngle} + 128\ )\ >>\ 8\ )\ ][\ -1\ ],\ \text{with}\ x = -1..(\ \text{nTbS} * \text{intraPredAngle}\ )\ >>\ 5 \qquad (8\text{-}56)$$

- Otherwise,

$$\text{ref}[\ x\ ] = p[\ -1\ ][\ -1 + x\ ],\ \text{with}\ x = \text{nTbS} + 1..2 * \text{nTbS} \qquad (8\text{-}57)$$

4. The values of the prediction samples predSamples[x][y], with x, y = 0..nTbS - 1 are derived as follows:

d. The index variable iIdx and the multiplication factor iFact are derived as follows:

$$\text{iIdx} = ((x + 1) * \text{intraPredAngle}) >> 5 \qquad (8\text{-}58)$$

$$\text{iFact} = ((x + 1) * \text{intraPredAngle}) \,\&\, 31 \qquad (8\text{-}59)$$

e. Depending on the value of iFact, the following applies:

- If iFact is not equal to 0, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\,x\,][\,y\,] = ((32 - \text{iFact}) * \text{ref}[\,y + \text{iIdx} + 1\,] + \text{iFact} * \text{ref}[\,y + \text{iIdx} + 2\,] + 16) >> 5 \qquad (8\text{-}60)$$

- Otherwise, the value of the prediction samples predSamples[x][y] is derived as follows:

$$\text{predSamples}[\,x\,][\,y\,] = \text{ref}[\,y + \text{iIdx} + 1 \qquad (8\text{-}61)$$

f. When predModeIntra is equal to 10 (horizontal), cIdx is equal to 0 and nTbS is less than 32, the following filtering applies with x = 0..nTbS - 1, y = 0:

$$\text{predSamples}[\,x\,][\,y\,] = \text{Clip1}_{\text{Y}}(\,p[\,-1\,][\,y\,] + ((p[\,x\,][\,-1\,] - p[\,-1\,][\,-1\,]) >> 1)) \qquad (8\text{-}62)$$

[0220] The methods further described in accordance with techniques of this disclosure are not according to the invention and are present for illustration purposes only.

[0221] FIG. 10 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 10 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder derives, from among a plurality of intra prediction modes, MMPMs for intra prediction of a block of video data (200). In one example, *M* may be greater than 3. The video encoder encodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode (202) of the plurality of intra prediction modes for intra prediction of the block of video data. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Based on the indicated one of the MPM index or the non-MPM index being the MPM index, the video encoder selects, for each of one or more context-modeled bins of the MPM index, based on intra prediction modes used to decode one or more neighboring blocks, a context index for the context-modeled bin (204). The video encoder encodes the block of video data based on the selected intra prediction mode (206).

[0222] FIG. 11 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 11 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder derives, from among a plurality of intra prediction modes, *M* MPMs for intra prediction of a block of video data (220). In one example, *M* may be greater than 3. The video decoder encodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode (222) of the plurality of intra prediction modes for intra prediction of the block of video data. The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the MMPMs is the selected intra prediction mode. Based on the indicated one of the MPM index or the non-MPM index being the MPM index, the video decoder selects, for each of one or more context-modeled bins of the MPM index, based on intra prediction modes used to decode one or more neighboring blocks, a context index for the context-modeled bin (224). The video decoder reconstructs the block of video data based on the selected intra prediction mode (226).

[0223] FIG. 12 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 12 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder derives *M* MPMs (300) for intra prediction of the block of video data from among a plurality of intra prediction modes. In one example, *M* may be greater than 3 and the MPMs may include an MPM for a left neighboring column and an MPM for an above neighboring row. In one example, the *M* MPMs may be derived by at least one of (i) defining a representative intra prediction mode for the left neighboring column and using the representative intra prediction mode for the left neighboring column as the MPM for the left neighboring column (302), or (ii) defining a representative intra prediction mode for the above neighboring row and using the representative intra prediction mode for the above neighboring row as the MPM for the above neighboring row (304).

[0224] The video encoder may encode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (306). Additionally, the video encoder may encode the indicated one of the MPM index or the non-MPM index (308). The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPM is the selected intra prediction mode. The video encoder may encode the block of video data based on the selected intra prediction mode (310).

[0225] FIG. 13 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 13 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder derives *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes (350). In one example, *M* may be greater than 3 and the MPMs may include an MPM for a left neighboring column and an MPM for an above neighboring row. The *M* most probable modes may be derived by at least one of (i) defining a representative intra prediction mode for the left neighboring column and using the representative intra prediction mode for the left neighboring column as the MPM for the left neighboring column (352), or (ii) defining a representative intra prediction mode for the above neighboring row and using the representative intra prediction mode for the above neighboring row as the MPM for the above neighboring row (354).

[0226] The video decoder may decode a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (356). Additionally, the video decoder may decode the indicated one of the MPM index or the non-MPM index (358). The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the MMPM is the selected intra prediction mode. The video decoder may encode the block of video data based on the selected intra prediction mode (360).

[0227] FIG. 14 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 14 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder derives *M* MPMs (400) for intra prediction of the block of video data from among a plurality of intra prediction modes. In one example, *M* may be greater than 3. The video encoder encodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (402). In the example of FIG. 12, the MPM index indicates which of the *M* MPMs is the selected intra prediction mode. Furthermore, the non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Based on the MPM index indicating the selected intra prediction mode, the video encoder encodes the non-MPM index (404). In the example of FIG. 14, the non-MPM index is encoded in the bitstream as a code word shorter than $\lceil \log_2 N \rceil$ bits if the non-MPM index satisfies a criterion and is encoded in the bitstream as a fixed length code with $\lceil \log_2 N \rceil$ bits otherwise. In the example of FIG. 14, there is a total of N available values of the non-MPM index. The video encoder encodes the block of video data based on the selected intra prediction mode (406).

[0228] FIG. 15 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 15 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder derives *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes (450). In one example, *M* may be greater than 3. The video decoder decodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (452). The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. Based on the MPM index indicating the selected intra prediction mode, the video decoder decodes the non-MPM index (454). In the example of FIG. 15, the non-MPM index is encoded in the bitstream as a code word shorter than $\lceil \log_2 N \rceil$ bits if the non-MPM index satisfies a criterion and is encoded in the bitstream as a fixed length code with $\lceil \log_2 N \rceil$ bits otherwise. In the example of FIG. 15, there may be a total of *N* available values of the non-MPM index. Furthermore, in the example of FIG. 15, the video decoder reconstructs the block of video data based on the selected intra prediction mode (456).

[0229] FIG. 16 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 16 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder derives *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes (600). In the example of FIG. 16, *M* is greater than 3. The video encoder encodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (602). The video encoder encodes the indicated one of the MPM index or the non-MPM index (604). The MPM index indicates which of the MMPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MPMs is the selected intra prediction mode. The video encoder encodes the block of video data based on the selected intra prediction mode (606).

[0230] FIG. 17 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 17 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder derives *M* MPMs for intra prediction of the block of video data from among a plurality of intra prediction modes (650). In the example of FIG. 17, *M* is greater than 3. The video decoder decodes a syntax element that indicates whether a MPM index or a non-MPM index is used to indicate a selected intra prediction mode of the plurality of intra prediction modes for intra prediction of the block of video data (652). The video decoder decodes the indicated one of the MPM index or the non-MPM index (654). The MPM index indicates which of the *M* MPMs is the selected intra prediction mode. The non-MPM index indicates which of the plurality of intra prediction modes other than the *M* MMMs is the selected intra prediction mode (654). The video decoder reconstructs the block of video data based on the selected intra prediction mode (656).

[0231] FIG. 18 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 18 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder encodes syntax information that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes (700). In one example, the plurality of intra prediction modes may include greater than 33 angular intra prediction modes. The angular intra prediction modes may be defined such that interpolation is performed in 1/32 pel accuracy. The video encoder encodes the block of video data based on the selected intra prediction mode (702).

[0232] FIG. 19 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 19 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder decodes syntax information that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes (750). In one example, the plurality of intra prediction modes may include greater than 33 angular intra prediction modes, and the angular intra prediction modes may be defined such that interpolation is performed in 1/32 pel accuracy. The video decoder may reconstruct the block of video data based on the selected intra prediction mode (752).

[0233] FIG. 20 is a flowchart illustrating a method of encoding video data in accordance with techniques of this disclosure. FIG. 20 is described with reference to a generic video encoder. The generic video encoder may, for example, correspond to video encoder 20, although the techniques of this disclosure are not limited to any specific type of video encoder. The video encoder encodes syntax information that indicates a selected intra prediction mode for the block of video data from among a plurality of intra prediction modes (800). The video encoder applies an *N*-tap intra interpolation filter to neighboring reconstructed samples of the block of video data according to the selected intra prediction mode (802). In one example, *N* may be greater than 2. The video encoder may encode the block of video data based on the filtered neighboring reconstructed samples according to the selected intra prediction mode (804).

[0234] FIG. 21 is a flowchart illustrating a method of decoding video data in accordance with techniques of this disclosure. FIG. 21 is described with reference to a generic video decoder. The generic video decoder may, for example, correspond to video decoder 30, although the techniques of this disclosure are not limited to any specific type of video decoder. The video decoder decodes syntax information that indicates a selected intra prediction mode for a block of video data from among a plurality of intra prediction modes (850). The video decoder applies an *N*-tap intra interpolation filter to neighboring reconstructed samples of the block of video data according to the selected intra prediction mode (852). In one example, *N* may be greater than 2. The video decoder may reconstruct the block of video data based on the filtered neighboring reconstructed samples according to the selected intra prediction mode (854).

[0235] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0236] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transitory media, but are instead directed to non-transitory, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0237] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0238] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

Claims

1. A method of decoding a block of video data, the method comprising:

> selecting, based on intra prediction modes used to decode one or more neighboring blocks of the block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes;
> decoding (750) syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and
> reconstructing (752) the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy, wherein reconstructing the block of video data comprises:
> for each respective sample of a predictive block of the current PU:
>
> > determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and
> > calculating a value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

2. The method of claim 1, further comprising determining, based on at least one of a coding unit size or a prediction unit size, a total number of intra prediction modes in the plurality of intra prediction modes.

3. The method of claim 1, wherein a total number of intra prediction modes in the plurality of intra prediction modes is signaled in at least one of: a sequence parameter set, picture parameter set, or a slice header.

4. The method of claim 1, wherein the interpolation filter is formulated as:

$$p_{xy} = (1- \alpha)\cdot L + \alpha\cdot R,$$

where p_{xy} is the calculated value of the respective sample, L and R are values of the two reconstructed neighboring samples, and α is the determined fractional position.

5. A method of encoding a block of video data, the method comprising:

selecting, based on intra prediction modes used to encode one or more neighboring blocks of the block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes;
encoding (700) syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and
encoding (702) the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy, wherein encoding the block of video data comprises:
for each respective sample of a predictive block of the current PU:

determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and
calculating a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

6. The method of claim 5, further comprising determining, based on at least one of a coding unit size or a prediction unit size, a total number of intra prediction modes in the plurality of intra prediction modes.

7. The method of claim 5, wherein a total number of intra prediction modes in the plurality of intra prediction modes is signaled in at least one of: a sequence parameter set, picture parameter set, or a slice header.

8. The method of claim 5, wherein the interpolation filter is formulated as:

$$p_{xy} = (1- \alpha)\cdot L + \alpha\cdot R,$$

where p_{xy} is the calculated value of the respective sample, L and R are values of the two reconstructed neighboring samples, and α is the determined fractional position.

9. A device for decoding video data, the device comprising:
a memory configured to store the video data; and
one or more processors configured to:

select, based on intra prediction modes used to decode one or more neighboring blocks of a block of video data in a current picture of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes;
decode (750) syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and
reconstruct (752) the block of video data based on the selected intra prediction mode, the angular intra prediction modes defined such that interpolation is performed in 1/32 pel accuracy, wherein reconstructing the block of video data comprises:

for each respective sample of a predictive block of the current PU:

determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and
calculating a value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

10. A device for encoding video data, the device comprising:
a memory configured to store the video data; and
one or more processors configured to:

select, based on intra prediction modes used to encode one or more neighboring blocks of a block of video data, a subset of 33 angular intra prediction modes from among a plurality of intra prediction modes, wherein the plurality of intra prediction modes includes 65 angular intra prediction modes;
encode (700) syntax information that indicates a selected intra prediction mode for a current prediction unit, PU, of the block of video data from among the subset of 33 angular intra prediction modes; and
encode (702) the block of video data based on the selected intra prediction mode using 1/32 pel interpolation accuracy, wherein encoding the block of video data comprises:
for each respective sample of a predictive block of the current PU:

determining a fractional position between two neighboring reconstructed samples by projecting, along a prediction direction associated with the selected intra prediction mode, a coordinate of the respective sample to a row or column of neighboring reconstructed samples containing the two neighboring reconstructed samples, wherein the two neighboring reconstructed samples neighbor the current PU; and
calculating a prediction value of the respective sample using an interpolation filter that uses values of the two neighboring reconstructed samples to interpolate a value at the determined fractional position, the interpolation being in 1/32 pel accuracy.

11. A computer readable medium that stores instructions that, when executed by one or more processors cause the one or more processors to perform the method of any of claims 1 to 8.

Patentansprüche

1. Verfahren zum Decodieren eines Blocks von Videodaten, wobei das Verfahren Folgendes beinhaltet:

Auswählen, auf der Basis von Intra-Prädiktionsmodi, die zum Decodieren eines oder mehrerer benachbarter Blöcke des Blocks von Videodaten benutzt werden, einer Teilmenge von 33 angularen Intra-Prädiktionsmodi aus mehreren Intra-Prädiktionsmodi, wobei die mehreren Intra-Prädiktionsmodi 65 angulare Intra-Prädiktionsmodi umfassen;
Decodieren (750) von Syntaxinformationen, die einen gewählten Intra-Prädiktionsmodus für eine aktuelle Prädiktionseinheit, PU, des Blocks von Videodaten aus der Teilmenge von 33 angularen Intra-Prädiktionsmodi anzeigen; und
Rekonstruieren (752) des Blocks von Videodaten auf der Basis des gewählten Intra-Prädiktionsmodus mit einer Interpolationsgenauigkeit von 1/32 pel, wobei das Rekonstruieren des Blocks von Videodaten Folgendes beinhaltet:
für jedes jeweilige Sample eines prädiktiven Blocks der aktuellen PU:

Bestimmen einer fraktionalen Position zwischen zwei benachbarten rekonstruierten Samples durch Projizieren, entlang einer mit dem gewählten Intra-Prädiktionsmodus assoziierten Prädiktionsrichtung, einer Koordinate des jeweiligen Sample auf eine Reihe oder Spalte von benachbarten rekonstruierten Samples, die die beiden benachbarten rekonstruierten Samples enthalten, wobei die beiden benachbarten rekonstruierten Samples der aktuellen PU benachbart sind; und
Berechnen eines Wertes des jeweiligen Sample mittels eines Interpolationsfilters, das Werte der beiden benachbarten rekonstruierten Samples zum Interpolieren eines Wertes an der bestimmten fraktionalen

Position benutzt, wobei die Interpolation mit einer Genauigkeit von 1/32 pel erfolgt.

2. Verfahren nach Anspruch 1, das ferner das Bestimmen, auf der Basis einer Codiereinheitsgröße und/oder einer Prädiktionseinheitsgröße, einer Gesamtzahl von Intra-Prädiktionsmodi der mehreren Intra-Prädiktionsmodi beinhaltet.

3. Verfahren nach Anspruch 1, wobei eine Gesamtzahl von Intra-Prädiktionsmodi aus den mehreren Intra-Prädiktionsmodi in mindestens einem aus einem Sequenzparametersatz, einem Bildparametersatz und einem Slice-Header signalisiert wird.

4. Verfahren nach Anspruch 1, wobei das Interpolationsfilter formuliert wird als:

$$p_{xy} = (1-\alpha)\cdot L + \alpha\cdot R,$$

wobei p_{xy} der berechnete Wert des jeweiligen Sample ist, L und R Werte der beiden rekonstruierten benachbarten Samples sind und α die bestimmte fraktionale Position ist.

5. Verfahren zum Encodieren eines Blocks von Videodaten, wobei das Verfahren Folgendes beinhaltet:

   Auswählen, auf der Basis von Intra-Prädiktionsmodi, die zum Encodieren eines oder mehrerer benachbarter Blöcke des Blocks von Videodaten benutzt werden, einer Teilmenge von 33 angularen Intra-Prädiktionsmodi aus mehreren Intra-Prädiktionsmodi, wobei die mehreren Intra-Prädiktionsmodi 65 angulare Intra-Prädiktionsmodi umfassen;
   Encodieren (700) von Syntaxinformationen, die einen gewählten Intra-Prädiktionsmodus für eine aktuelle Prädiktionseinheit, PU, des Blocks von Videodaten aus der Teilmenge von 33 angularen Intra-Prädiktionsmodi anzeigen; und
   Encodieren (702) des Blocks von Videodaten auf der Basis des gewählten Intra-Prädiktionsmodus mit einer Interpolationsgenauigkeit von 1/32 pel, wobei das Encodieren des Blocks von Videodaten Folgendes beinhaltet: für jedes jeweilige Sample eines prädiktiven Blocks der aktuellen PU:

      Bestimmen einer fraktionalen Position zwischen zwei benachbarten rekonstruierten Samples durch Projizieren, entlang einer mit dem gewählten Intra-Prädiktionsmodus assoziierten Prädiktionsrichtung, einer Koordinate des jeweiligen Sample auf eine Reihe oder Spalte benachbarter rekonstruierter Samples, die die beiden benachbarten rekonstruierten Samples enthalten, wobei die beiden benachbarten rekonstruierten Samples der aktuellen PU benachbart sind; und
      Berechnen eines Prädiktionswertes des jeweiligen Sample mit einem Interpolationsfilter, das Werte der beiden benachbarten rekonstruierten Samples zum Interpolieren eines Wertes an der bestimmten fraktionalen Position benutzt, wobei die Interpolation mit einer Genauigkeit von 1/32 pel erfolgt.

6. Verfahren nach Anspruch 5, das ferner das Bestimmen, auf der Basis einer Codiereinheitsgröße und/oder einer Prädiktionseinheitsgröße, einer Gesamtzahl von Intra-Prädiktionsmodi der mehreren Intra-Prädiktionsmodi beinhaltet.

7. Verfahren nach Anspruch 5, wobei eine Gesamtzahl von Intra-Prädiktionsmodi der mehreren Intra-Prädiktionsmodi in mindestens einem aus einem Sequenzparametersatz, einem Bildparametersatz und einem Slice-Header signalisiert wird.

8. Verfahren nach Anspruch 5, wobei das Interpolationsfilter formuliert wird als:

$$p_{xy} = (1-\alpha)\cdot L + \alpha\cdot R,$$

wobei p_{xy} der berechnete Wert des jeweiligen Sample ist, L und R Werte der beiden rekonstruierten benachbarten Samples sind und α die bestimmte fraktionale Position ist.

9. Gerät zum Decodieren von Videodaten, wobei das Gerät Folgendes umfasst:
einen Speicher, der zum Speichern der Videodaten konfiguriert ist; und

einen oder mehrere Prozessoren, konfiguriert zum:

Auswählen, auf der Basis von Intra-Prädiktionsmodi, die zum Decodieren von einem oder mehreren benachbarten Blöcken eines Blocks von Videodaten in einem aktuellen Bild von Videodaten benutzt werden, einer Teilmenge von 33 angularen Intra-Prädiktionsmodi aus mehreren Intra-Prädiktionsmodi, wobei die mehreren Intra-Prädiktionsmodi 65 angulare Intra-Prädiktionsmodi umfassen;
Decodieren (750) von Syntaxinformationen, die einen gewählten Intra-Prädiktionsmodus für eine aktuelle Prädiktionseinheit, PU, des Blocks von Videodaten aus der Teilmenge von 33 angularen Intra-Prädiktionsmodi anzeigen; und
Rekonstruieren (752) des Blocks von Videodaten auf der Basis des gewählten Intra-Prädiktionsmodus, wobei die angularen Intra-Prädiktionsmodi so definiert sind, dass Interpolation mit einer Genauigkeit von 1/32 pel durchgeführt wird, wobei das Rekonstruieren des Blocks von Videodaten Folgendes beinhaltet:
für jedes jeweilige Sample eines prädiktiven Blocks der aktuellen PU:

Bestimmen einer fraktionalen Position zwischen zwei benachbarten rekonstruierten Samples durch Projizieren, entlang einer mit dem gewählten Intra-Prädiktionsmodus assoziierten Prädiktionsrichtung, einer Koordinate des jeweiligen Sample auf eine Reihe oder Spalte von benachbarten rekonstruierten Samples, die die beiden benachbarten rekonstruierten Samples enthalten, wobei die beiden benachbarten rekonstruierten Samples der aktuellen PU benachbart sind; und
Berechnen eines Wertes des jeweiligen Sample mittels eines Interpolationsfilters, das Werte der beiden benachbarten rekonstruierten Samples zum Interpolieren eines Wertes an der bestimmten fraktionalen Position benutzt, wobei die Interpolation mit einer Genauigkeit von 1/32 pel erfolgt.

10. Gerät zum Encodieren von Videodaten, wobei das Gerät Folgendes umfasst:
einen Speicher, der zum Speichern der Videodaten konfiguriert ist; und
einen oder mehrere Prozessoren, konfiguriert zum:

Auswählen, auf der Basis von Intra-Prädiktionsmodi, die zum Encodieren von einem oder mehreren benachbarten Blöcken eines Blocks von Videodaten verwendet werden, einer Teilmenge von 33 angularen Intra-Prädiktionsmodi aus mehreren Intra-Prädiktionsmodi, wobei die mehreren Intra-Prädiktionsmodi 65 angulare Intra-Pädiktionsmodi umfassen;
Encodieren (700) von Syntaxinformationen, die einen gewählten Intra-Prädiktionsmodus für eine aktuelle Prädiktionseinheit, PU, des Blocks von Videodaten aus der Teilmenge von 33 angularen Intra-Prädiktionsmodi anzeigen; und
Encodieren (702) des Blocks von Videodaten auf der Basis des gewählten Intra-Prädiktionsmodus mit einer Interpolationsgenauigkeit von 1/32 pel, wobei das Encodieren des Blocks von Videodaten Folgendes beinhaltet:
für jedes jeweilige Sample eines prädiktiven Blocks der aktuellen PU:

Bestimmen einer fraktionalen Position zwischen zwei benachbarten rekonstruierten Samples durch Projizieren, entlang einer mit dem gewählten Intra-Prädiktionsmodus assoziierten Prädiktionsrichtung, einer Koordinate des jeweiligen Sample auf eine Reihe oder Spalte von benachbarten rekonstruierten Samples, die die beiden benachbarten rekonstruierten Samples enthalten, wobei die beiden benachbarten rekonstruierten Samples der aktuellen PU benachbart sind; und
Berechnen eines Prädiktionswertes des jeweiligen Sample mittels eines Interpolationsfilters, das Werte der beiden benachbarten rekonstruierten Samples zum Interpolieren eines Wertes an der bestimmten fraktionalen Position verwendet, wobei die Interpolation mit einer Genauigkeit von 1/32 pel erfolgt.

11. Computerlesbares Medium, das Befehle speichert, die bei Ausführung durch einen oder mehrere Prozessoren bewirken, dass die ein oder mehreren Prozessoren das Verfahren nach einem der Ansprüche 1 bis 8 durchführen.

Revendications

1. Procédé de décodage d’un bloc de données vidéo, le procédé comprenant :

la sélection, en fonction de modes d’intra-prédiction utilisés pour décoder un ou plusieurs blocs voisins du bloc de données vidéo, d’un sous-ensemble de 33 modes d’intra-prédiction angulaire parmi une pluralité de modes d’intra-prédiction, la pluralité de modes d’intra-prédiction comprenant 65 modes d’intra-prédiction angulaire ;

le décodage (750) d'informations de syntaxe qui indiquent un mode d'intra-prédiction sélectionné pour une unité de prédiction, PU, actuelle, du bloc de données vidéo parmi le sous-ensemble de 33 modes d'intra-prédiction angulaire ; et
la reconstruction (752) du bloc de données vidéo en fonction du mode d'intra-prédiction sélectionné en utilisant une précision d'interpolation de 1/32 pixel, la reconstruction du bloc de données vidéo comprenant :
pour chaque échantillon respectif d'un bloc prédictif de la PU actuelle :

la détermination d'une position fractionnaire entre deux échantillons reconstruits voisins en projetant, le long d'une direction de prédiction associée au mode d'intra-prédiction sélectionné, une coordonnée de l'échantillon respectif vers une rangée ou une colonne d'échantillons reconstruits voisins contenant les deux échantillons reconstruits voisins, les deux échantillons reconstruits voisins étant voisins de la PU actuelle ; et
le calcul d'une valeur de l'échantillon respectif à l'aide d'un filtre d'interpolation qui utilise des valeurs des deux échantillons reconstruits voisins pour interpoler une valeur à la position fractionnaire déterminée, l'interpolation étant d'une précision de 1/32 pixel.

2. Procédé selon la revendication 1, comprenant en outre la détermination, sur la base d'au moins une taille d'unité de codage ou d'une taille d'unité de prédiction, d'un nombre total de modes d'intra-prédiction dans la pluralité des modes d'intra-prédiction.

3. Procédé selon la revendication 1, dans lequel un nombre total de modes d'intra-prédiction dans la pluralité des modes d'intra-prédiction est signalé dans au moins un : d'un ensemble de paramètres de séquence, d'un ensemble de paramètres d'image ou d'un en-tête de tranche.

4. Procédé selon revendication 1, dans lequel le filtre d'interpolation est formulé comme suit

$$p_{xy} = (1 -\alpha).L + \alpha+R,$$

où p_{xy} est la valeur calculée de l'échantillon respectif, L et R sont des valeurs des deux échantillons reconstruits voisins, et α est la position fractionnaire déterminée.

5. Procédé d'encodage d'un bloc de données vidéo, le procédé comprenant :

la sélection, en fonction de modes d'intra-prédiction utilisés pour décoder un ou plusieurs blocs voisins du bloc de données vidéo, d'un sous-ensemble de 33 modes d'intra-prédiction angulaire parmi une pluralité de modes d'intra-prédiction, la pluralité de modes d'intra-prédiction comprenant 65 modes d'intra-prédiction angulaire ;
l'encodage (700) d'informations de syntaxe qui indiquent un mode d'intra-prédiction sélectionné pour une unité de prédiction, PU, actuelle du bloc de données vidéo parmi le sous-ensemble de 33 modes d'intra-prédiction angulaire ; et
l'encodage (702) du bloc de données vidéo en fonction du mode d'intra-prédiction sélectionné en utilisant une précision d'interpolation de 1/32 pixel, l'encodage du bloc de données vidéo comprenant :
pour chaque échantillon respectif d'un bloc prédictif de la PU actuelle :

la détermination d'une position fractionnaire entre deux échantillons reconstruits voisins en projetant, le long d'une direction de prédiction associée au mode d'intra-prédiction sélectionné, une coordonnée de l'échantillon respectif vers une rangée ou une colonne d'échantillons reconstruits voisins contenant les deux échantillons reconstruits voisins, les deux échantillons reconstruits voisins étant voisins de la PU actuelle ; et
le calcul d'une valeur de prédiction de l'échantillon respectif à l'aide d'un filtre d'interpolation qui utilise des valeurs des deux échantillons reconstruits voisins pour interpoler une valeur à la position fractionnaire déterminée, l'interpolation étant d'une précision de 1/32 pixel.

6. Procédé selon la revendication 5, comprenant en outre la détermination, sur la base d'au moins une taille d'unité de codage ou d'une taille d'unité de prédiction, d'un nombre total de modes d'intra-prédiction dans la pluralité de modes d'intra-prédiction.

7. Procédé selon la revendication 5, dans lequel un nombre total de modes d'intra-prédiction dans la pluralité de modes d'intra-prédiction est signalé dans au moins un : d'un ensemble de paramètres de séquence, d'un ensemble de paramètres d'image, ou d'un en-tête de tranche.

8. Procédé selon la revendication 5, dans lequel le filtre d'interpolation est formulé comme suit :

$$p_{xy} = (1 - \alpha).L + \alpha + R,$$

où p_{xy} est la valeur calculée de l'échantillon respectif, L et R sont des valeurs des deux échantillons reconstruits voisins, et α est la position fractionnaire déterminée.

9. Dispositif de décodage de données vidéo, le dispositif comprenant :

une mémoire configurée pour mémoriser les données vidéo ; et
un ou plusieurs processeurs configurés pour :

sélectionner, en fonction de modes d'intra-prédiction utilisés pour décoder un ou plusieurs blocs voisins du bloc de données vidéo, un sous-ensemble de 33 modes d'intra-prédiction angulaire parmi une pluralité de modes d'intra-prédiction, la pluralité de modes d'intra-prédiction comprenant 65 modes d'intra-prédiction angulaire ;
décoder (750) des informations de syntaxe qui indiquent un mode d'intra-prédiction sélectionné pour une unité de prédiction, PU, actuelle, du bloc de données vidéo parmi le sous-ensemble de 33 modes d'intra-prédiction angulaire ; et
reconstruire (752) le bloc de données vidéo en fonction du mode d'intra-prédiction sélectionné en utilisant une précision d'interpolation de 1/32 pixel, la reconstruction du bloc de données vidéo comprenant :
pour chaque échantillon respectif d'un bloc prédictif de la PU actuelle :

la détermination d'une position fractionnaire entre deux échantillons reconstruits voisins en projetant, le long d'une direction de prédiction associée au mode d'intra-prédiction sélectionné, une coordonnée de l'échantillon respectif vers une rangée ou une colonne d'échantillons reconstruits voisins contenant les deux échantillons reconstruits voisins, les deux échantillons reconstruits voisins étant voisins de la PU actuelle ; et
le calcul d'une valeur de l'échantillon respectif à l'aide d'un filtre d'interpolation qui utilise des valeurs des deux échantillons reconstruits voisins pour interpoler une valeur à la position fractionnaire déterminée, l'interpolation étant d'une précision de 1/32 pixel.

10. Dispositif d'encodage de données vidéo, le dispositif comprenant :

une mémoire configurée pour mémoriser les données vidéo ; et
un ou plusieurs processeurs configurés pour :

sélectionner, en fonction de modes d'intra-prédiction utilisés pour décoder un ou plusieurs blocs voisins du bloc de données vidéo, un sous-ensemble de 33 modes d'intra-prédiction angulaire parmi une pluralité de modes d'intra-prédiction, la pluralité de modes d'intra-prédiction comprenant 65 modes d'intra-prédiction angulaire ;
encoder (700) des informations de syntaxe qui indiquent un mode d'intra-prédiction sélectionné pour une unité de prédiction, PU, actuelle du bloc de données vidéo parmi le sous-ensemble de 33 modes d'intra-prédiction angulaire ; et
encoder (702) le bloc de données vidéo en fonction du mode d'intra-prédiction sélectionné en utilisant une précision d'interpolation de 1/32 pixel, l'encodage du bloc de données vidéo comprenant :
pour chaque échantillon respectif d'un bloc prédictif de la PU actuelle :

la détermination d'une position fractionnaire entre deux échantillons reconstruits voisins en projetant, le long d'une direction de prédiction associée au mode d'intra-prédiction sélectionné, une coordonnée de l'échantillon respectif vers une rangée ou une colonne d'échantillons reconstruits voisins contenant les deux échantillons reconstruits voisins, les deux échantillons reconstruits voisins étant voisins de la PU actuelle ; et
le calcul d'une valeur de prédiction de l'échantillon respectif à l'aide d'un filtre d'interpolation qui utilise

des valeurs des deux échantillons reconstruits voisins pour interpoler une valeur à la position fractionnaire déterminée, l'interpolation étant d'une précision de 1/32 pixel.

11. Support lisible par ordinateur mémorisant des instructions qui, à leur exécution par un ou plusieurs processeurs, amènent les un ou plusieurs processeurs à réaliser le procédé selon l'une des revendications 1 à 8.

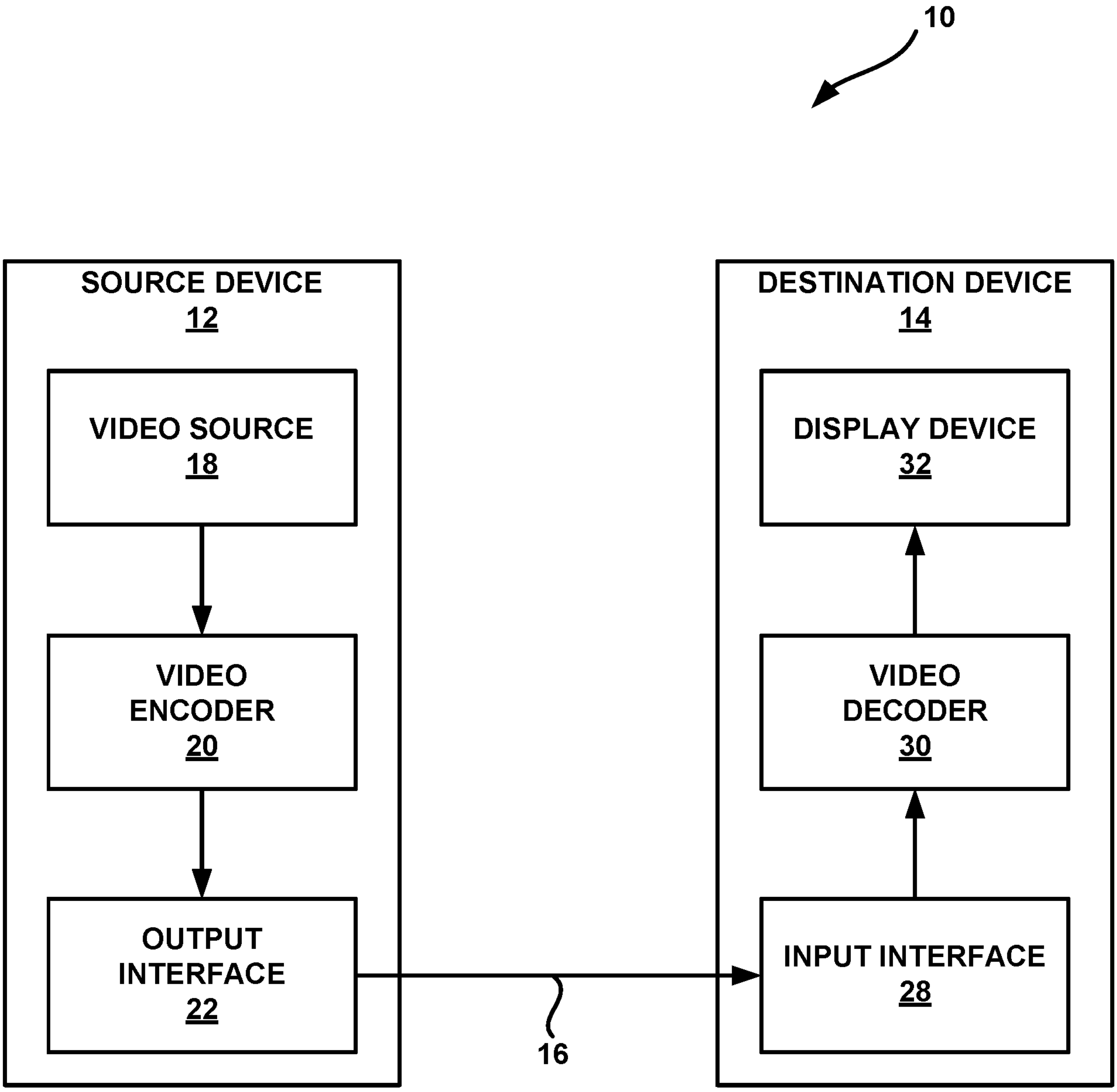
10
SOURCE DEVICE
12
VIDEO SOURCE
18
VIDEO
ENCODER
20
OUTPUT
INTERFACE
22
16
DESTINATION DEVICE
14
DISPLAY DEVICE
32
VIDEO
DECODER
30
INPUT INTERFACE
28

FIG. 1

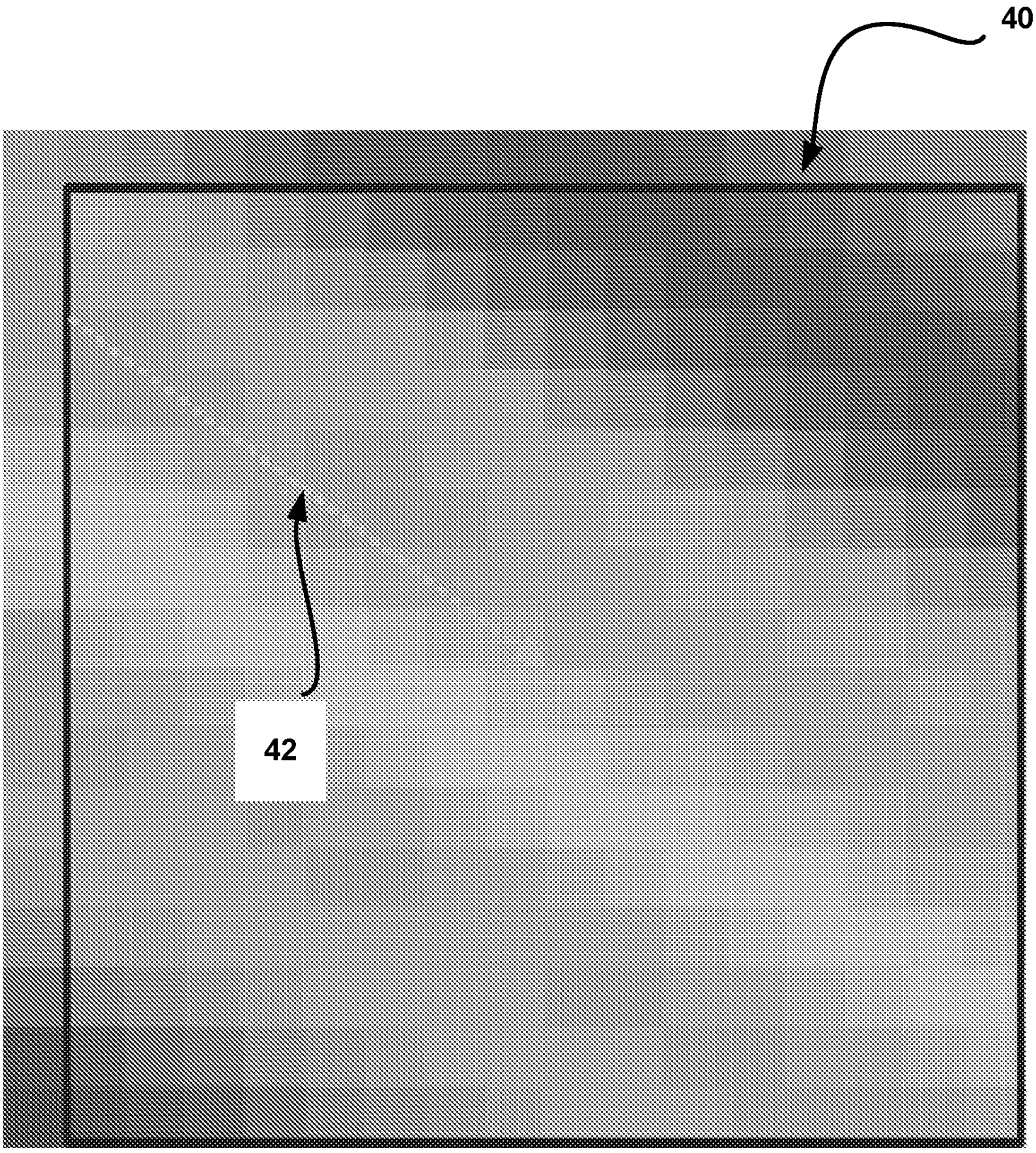

FIG. 2

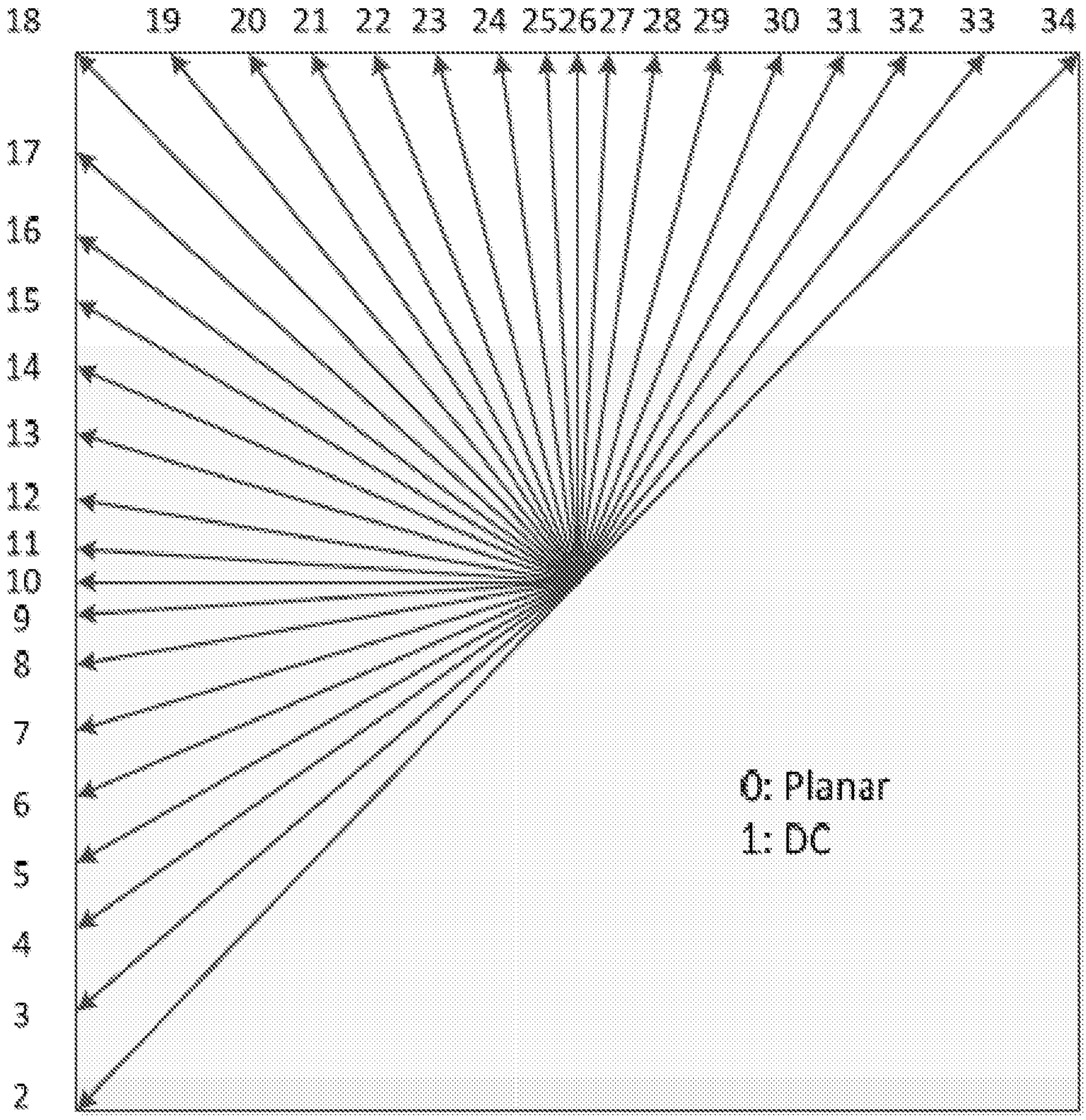
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
0: Planar
1: DC

FIG. 3

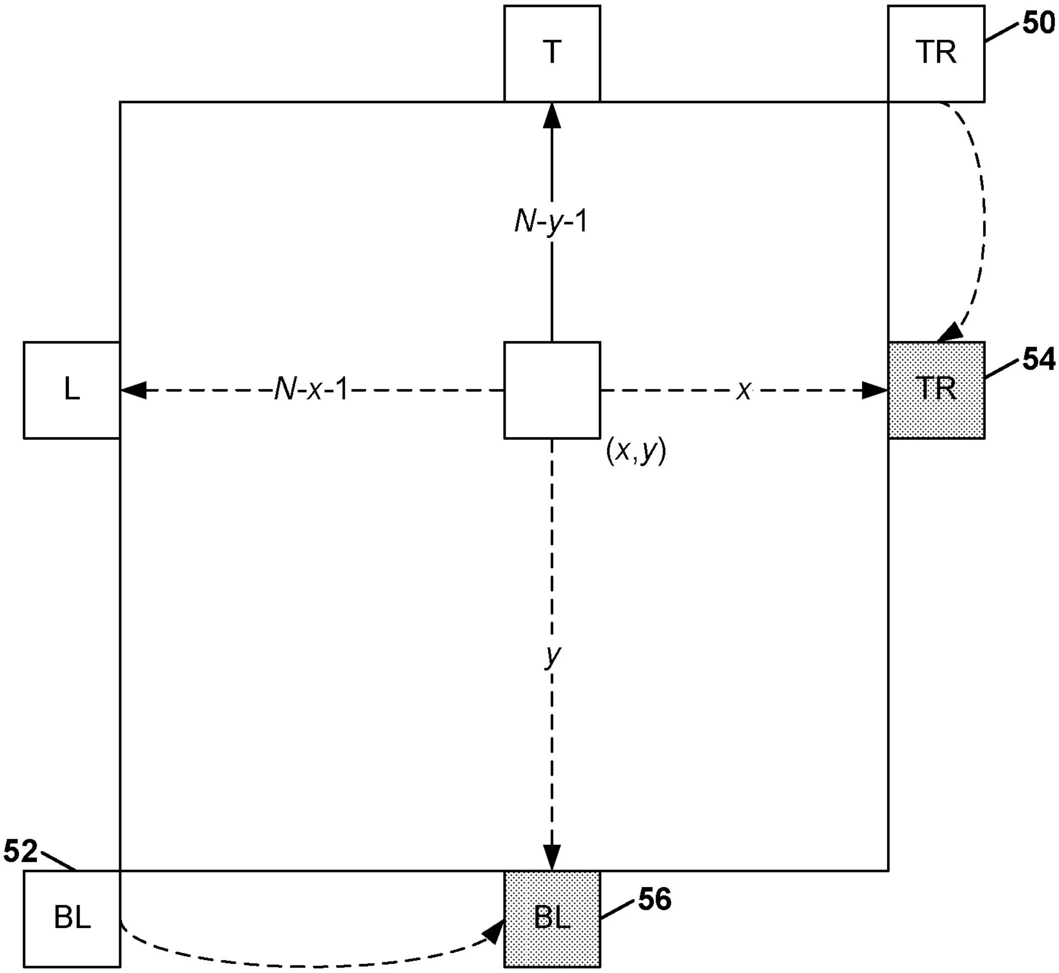

FIG. 4

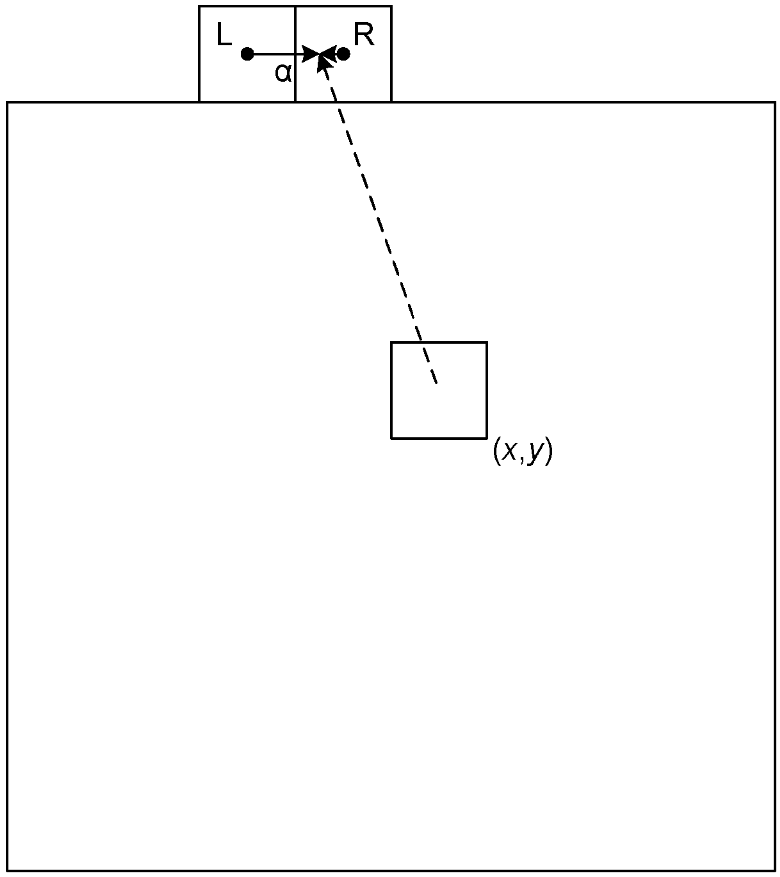

FIG. 5

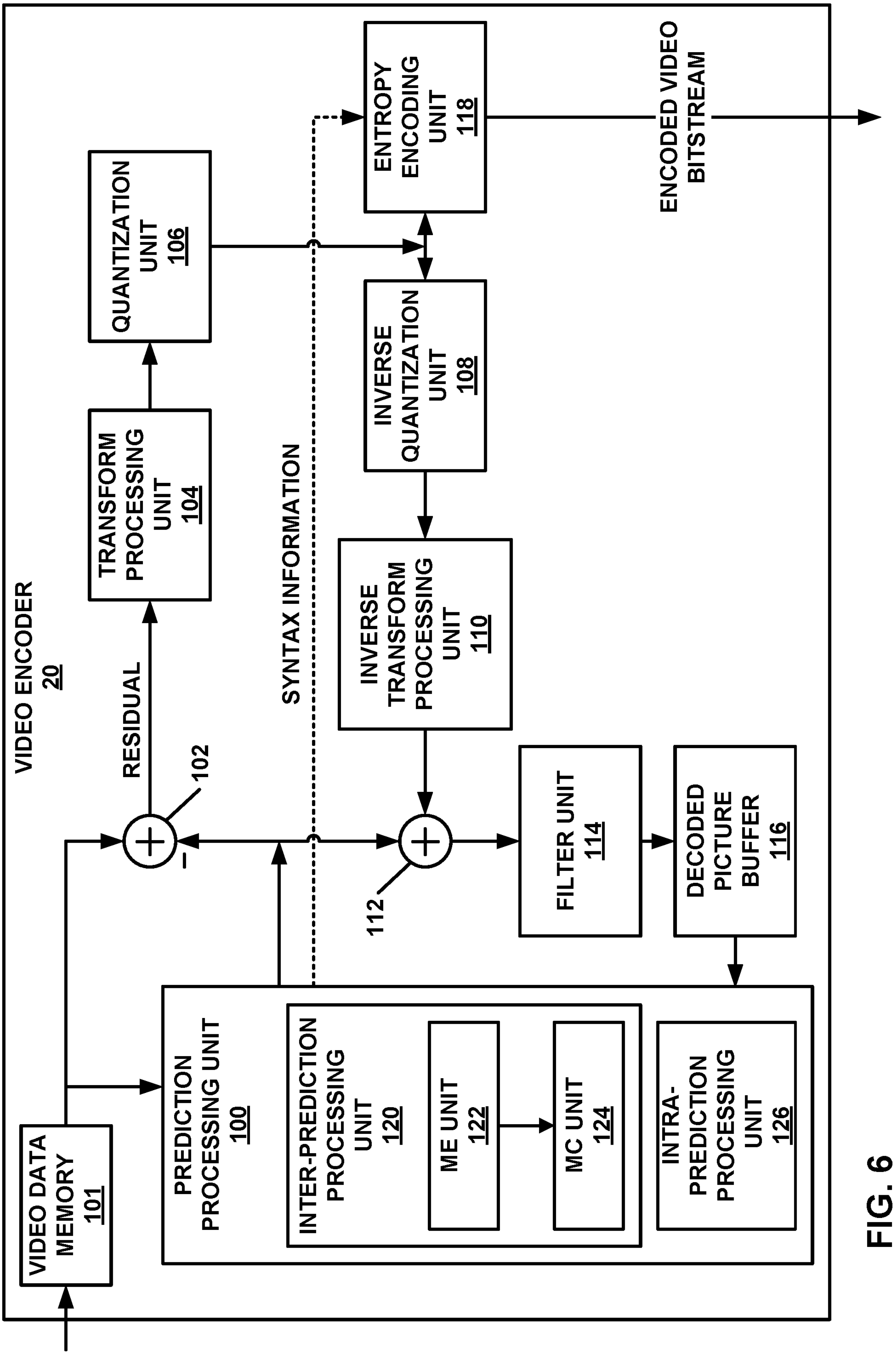
VIDEO ENCODER
20
VIDEO DATA MEMORY
101
PREDICTION PROCESSING UNIT
100
INTER-PREDICTION PROCESSING UNIT
120
ME UNIT
122
MC UNIT
124
INTRA-PREDICTION PROCESSING UNIT
126
RESIDUAL
102
–
TRANSFORM PROCESSING UNIT
104
QUANTIZATION UNIT
106
SYNTAX INFORMATION
INVERSE QUANTIZATION UNIT
108
INVERSE TRANSFORM PROCESSING UNIT
110
112
FILTER UNIT
114
DECODED PICTURE BUFFER
116
ENTROPY ENCODING UNIT
118
ENCODED VIDEO BITSTREAM

FIG. 6

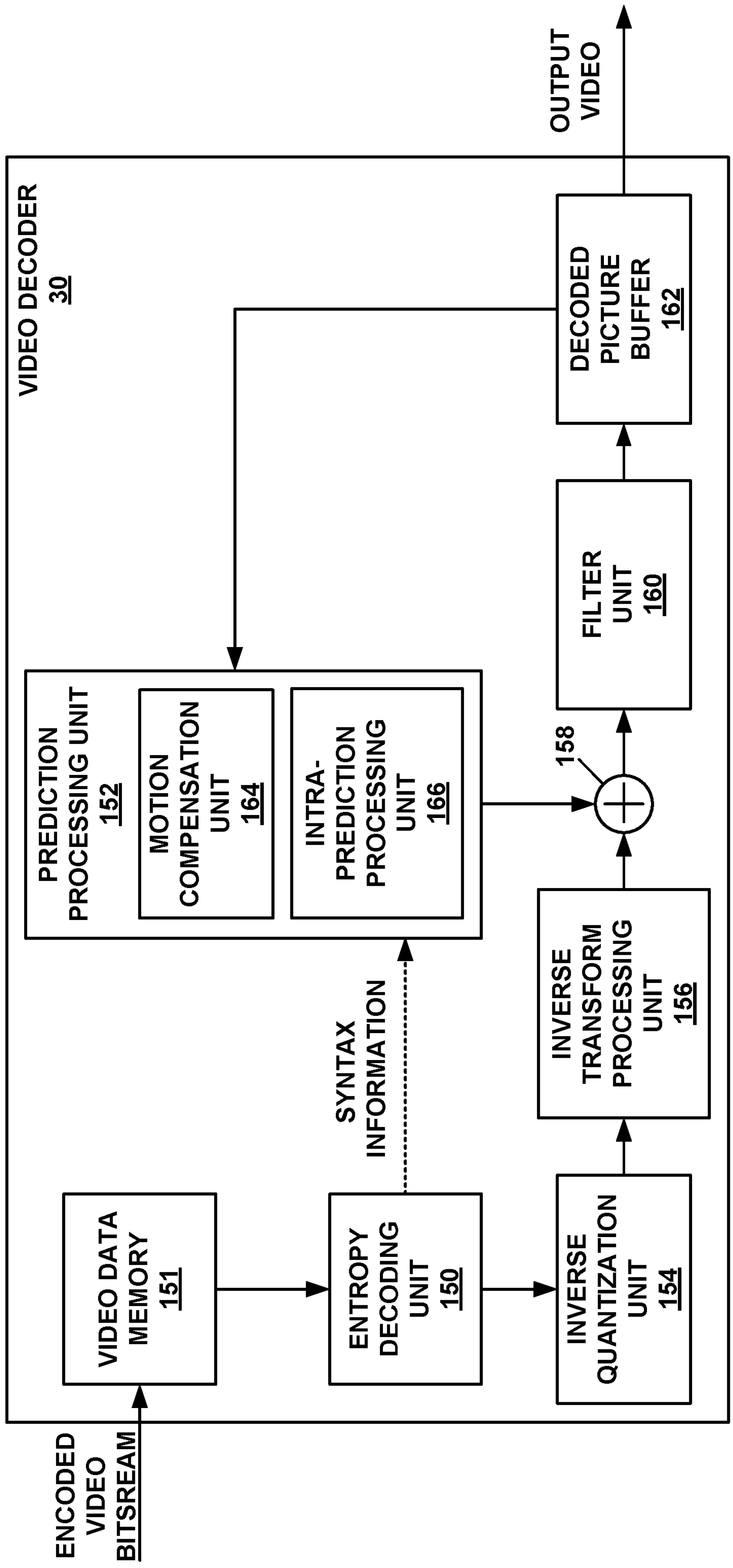
VIDEO DECODER 30
ENCODED VIDEO BITSREAM
VIDEO DATA MEMORY 151
ENTROPY DECODING UNIT 150
SYNTAX INFORMATION
PREDICTION PROCESSING UNIT 152
MOTION COMPENSATION UNIT 164
INTRA-PREDICTION PROCESSING UNIT 166
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM PROCESSING UNIT 156
158
FILTER UNIT 160
DECODED PICTURE BUFFER 162
OUTPUT VIDEO

FIG. 7

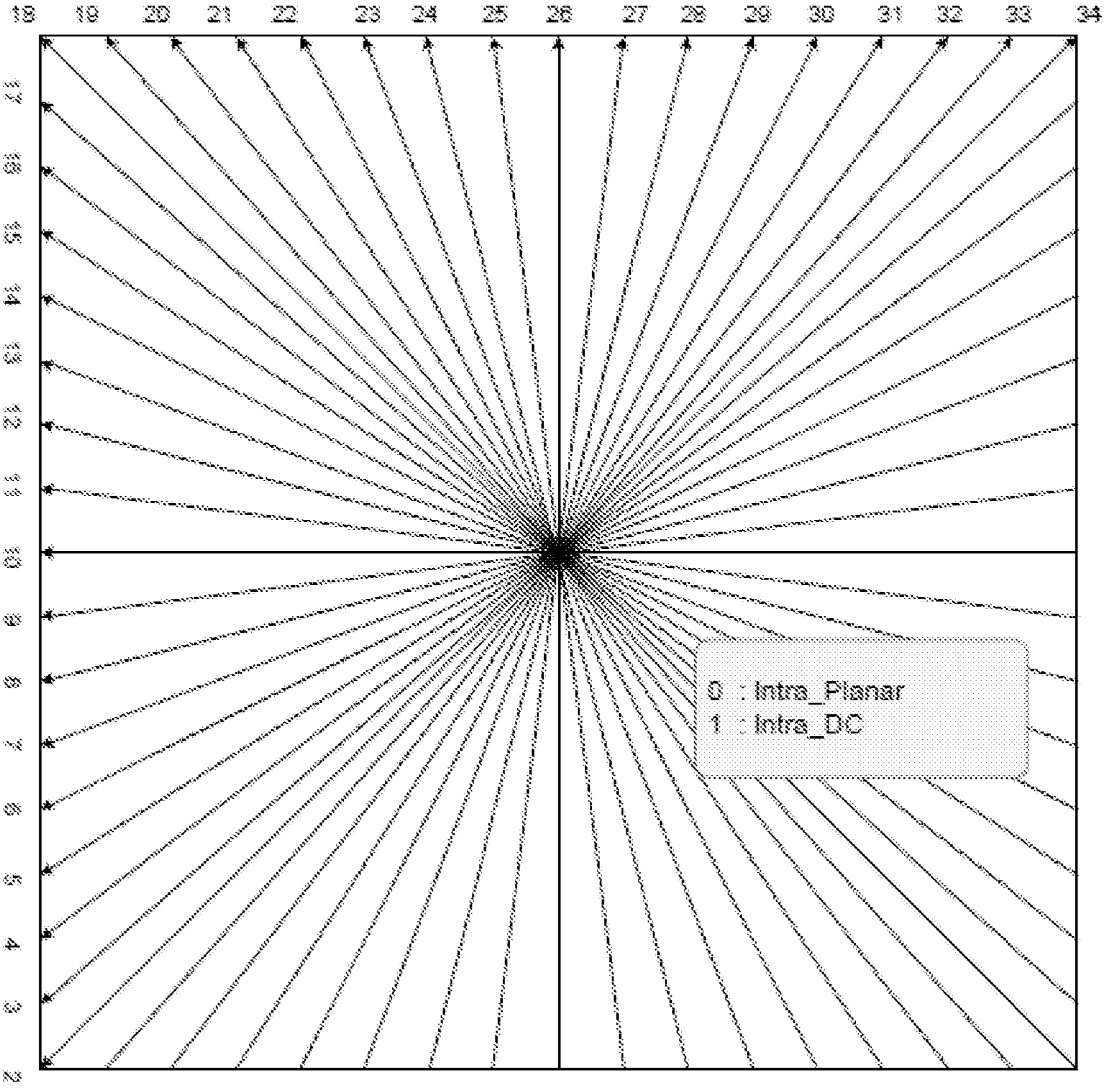
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
0 : Intra_Planar
1 : Intra_DC

FIG. 8

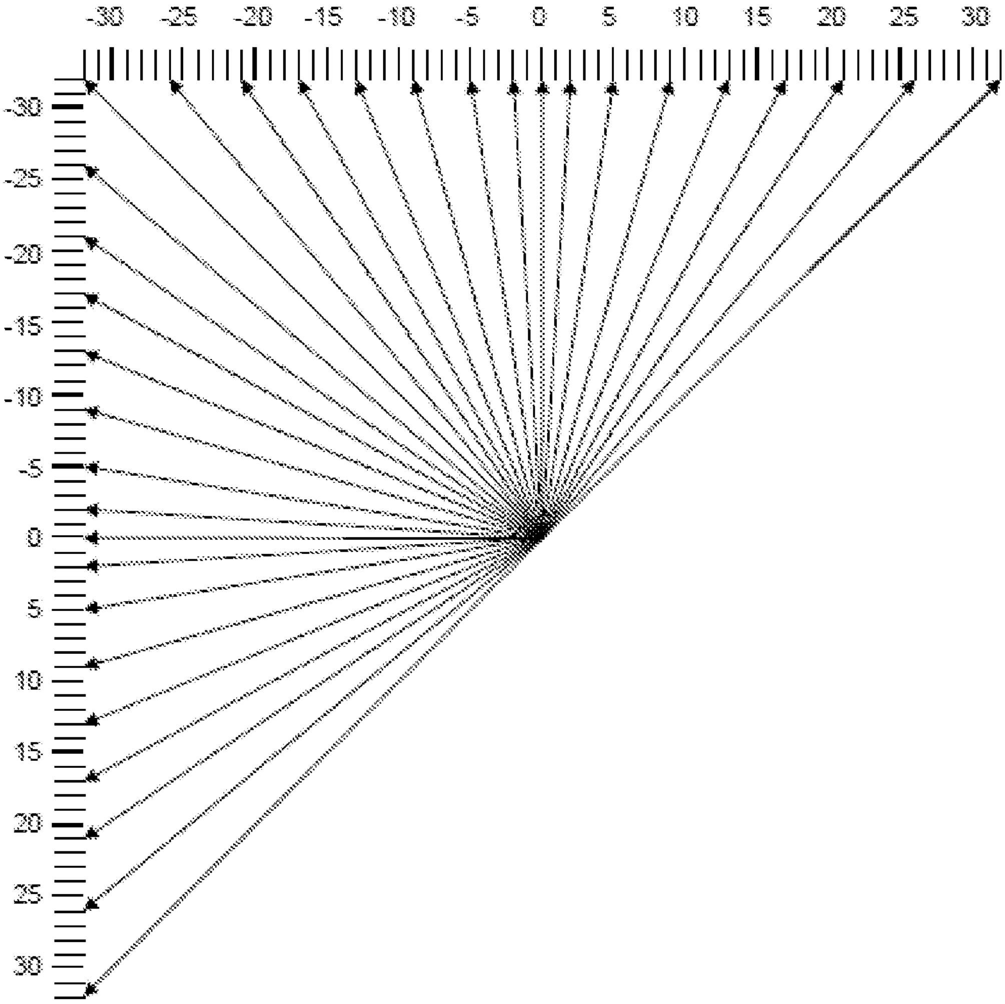

FIG. 9

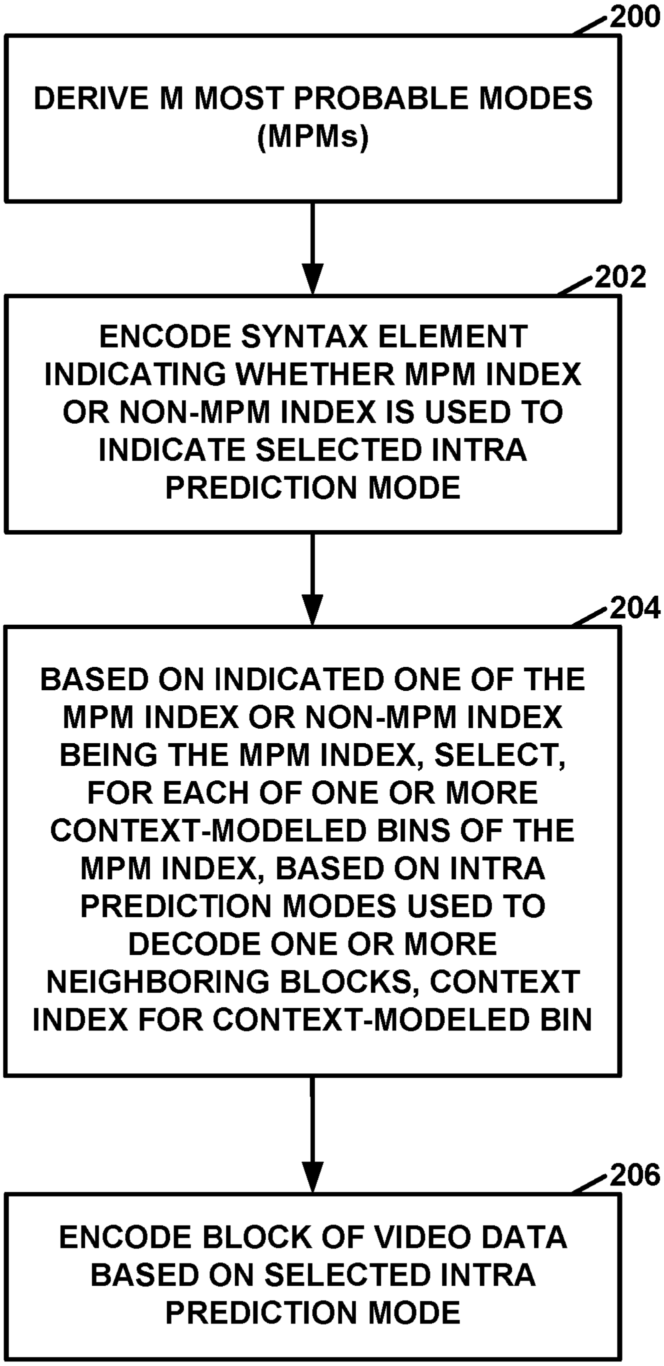
200
DERIVE M MOST PROBABLE MODES (MPMs)
202
ENCODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
204
BASED ON INDICATED ONE OF THE MPM INDEX OR NON-MPM INDEX BEING THE MPM INDEX, SELECT, FOR EACH OF ONE OR MORE CONTEXT-MODELED BINS OF THE MPM INDEX, BASED ON INTRA PREDICTION MODES USED TO DECODE ONE OR MORE NEIGHBORING BLOCKS, CONTEXT INDEX FOR CONTEXT-MODELED BIN
206
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 10

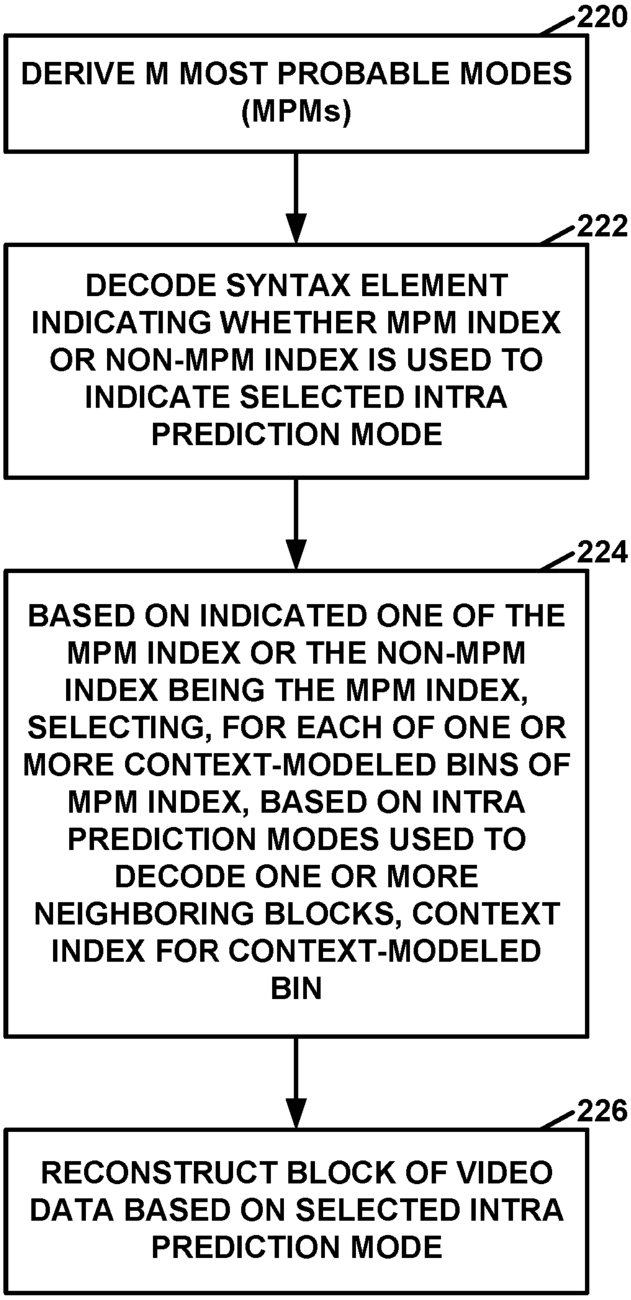
220
DERIVE M MOST PROBABLE MODES (MPMs)
222
DECODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
224
BASED ON INDICATED ONE OF THE MPM INDEX OR THE NON-MPM INDEX BEING THE MPM INDEX, SELECTING, FOR EACH OF ONE OR MORE CONTEXT-MODELED BINS OF MPM INDEX, BASED ON INTRA PREDICTION MODES USED TO DECODE ONE OR MORE NEIGHBORING BLOCKS, CONTEXT INDEX FOR CONTEXT-MODELED BIN
226
RECONSTRUCT BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 11

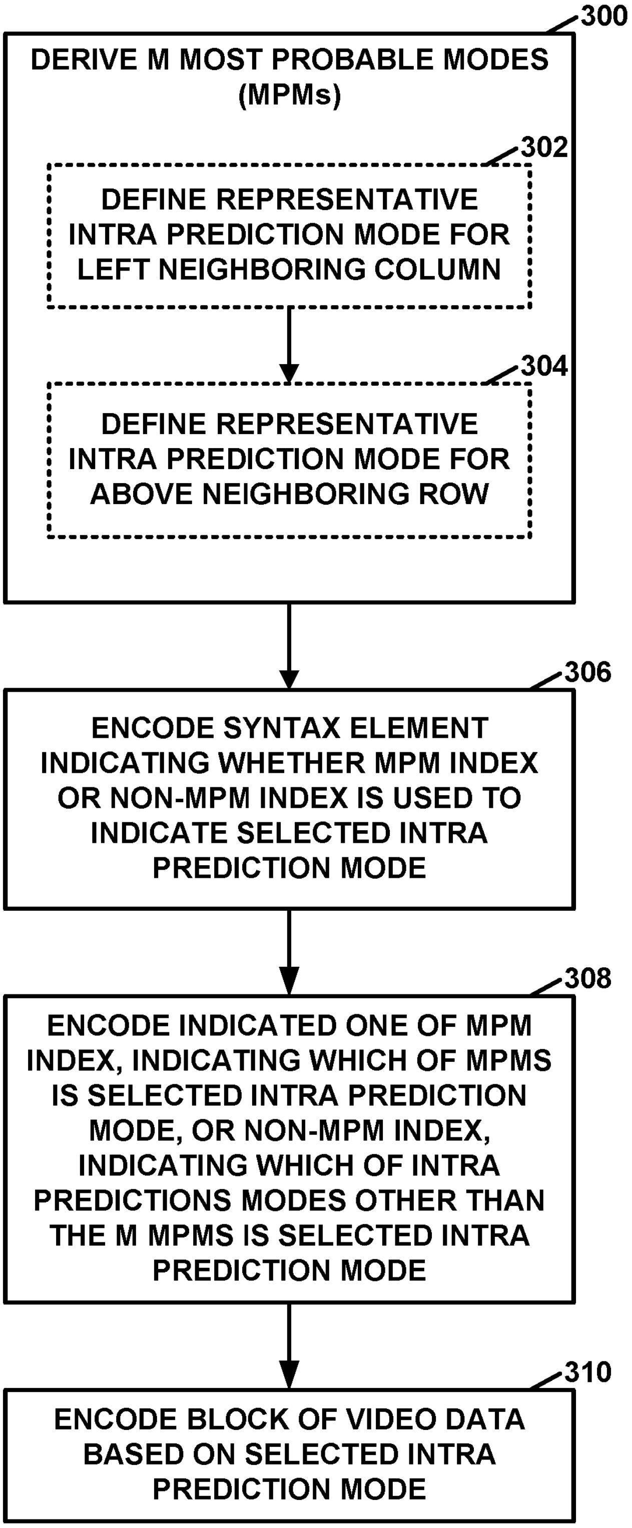
300
DERIVE M MOST PROBABLE MODES (MPMs)
302
DEFINE REPRESENTATIVE INTRA PREDICTION MODE FOR LEFT NEIGHBORING COLUMN
304
DEFINE REPRESENTATIVE INTRA PREDICTION MODE FOR ABOVE NEIGHBORING ROW
306
ENCODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
308
ENCODE INDICATED ONE OF MPM INDEX, INDICATING WHICH OF MPMS IS SELECTED INTRA PREDICTION MODE, OR NON-MPM INDEX, INDICATING WHICH OF INTRA PREDICTIONS MODES OTHER THAN THE M MPMS IS SELECTED INTRA PREDICTION MODE
310
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 12

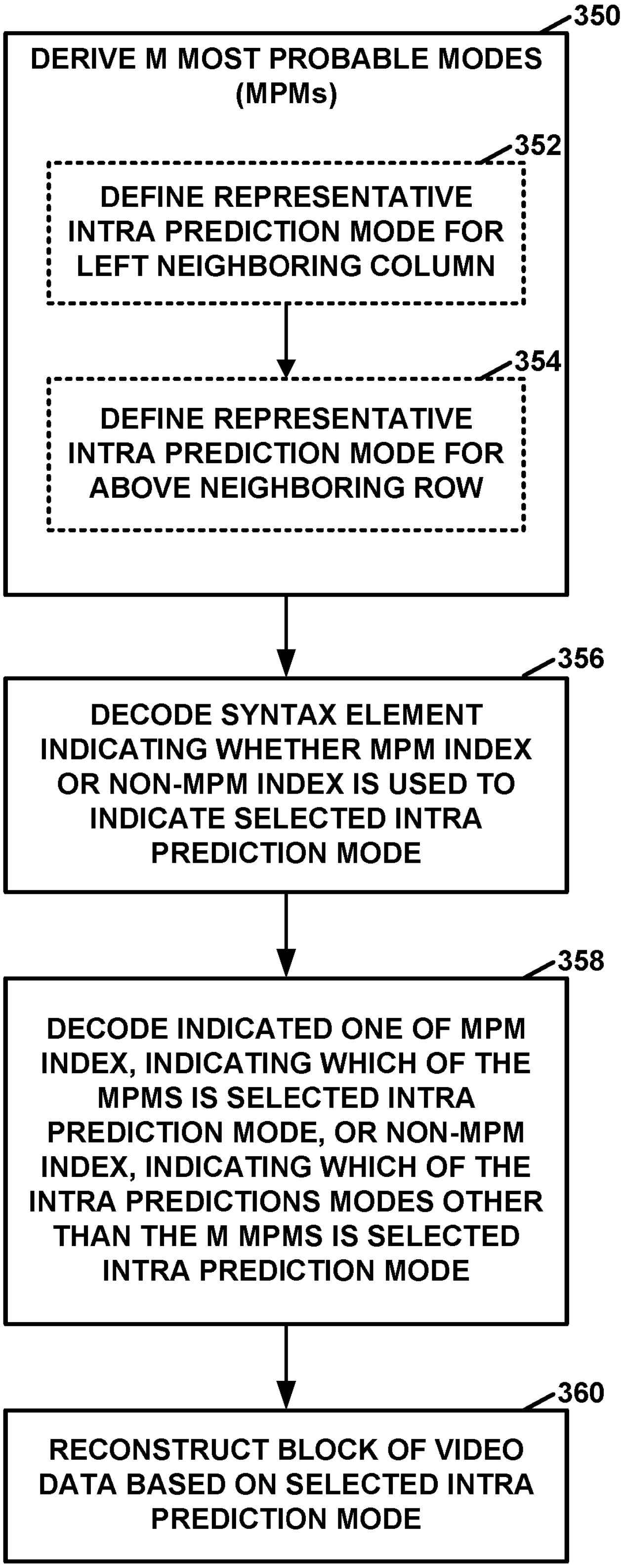
350
DERIVE M MOST PROBABLE MODES (MPMs)
352
DEFINE REPRESENTATIVE INTRA PREDICTION MODE FOR LEFT NEIGHBORING COLUMN
354
DEFINE REPRESENTATIVE INTRA PREDICTION MODE FOR ABOVE NEIGHBORING ROW
356
DECODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
358
DECODE INDICATED ONE OF MPM INDEX, INDICATING WHICH OF THE MPMS IS SELECTED INTRA PREDICTION MODE, OR NON-MPM INDEX, INDICATING WHICH OF THE INTRA PREDICTIONS MODES OTHER THAN THE M MPMS IS SELECTED INTRA PREDICTION MODE
360
RECONSTRUCT BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 13

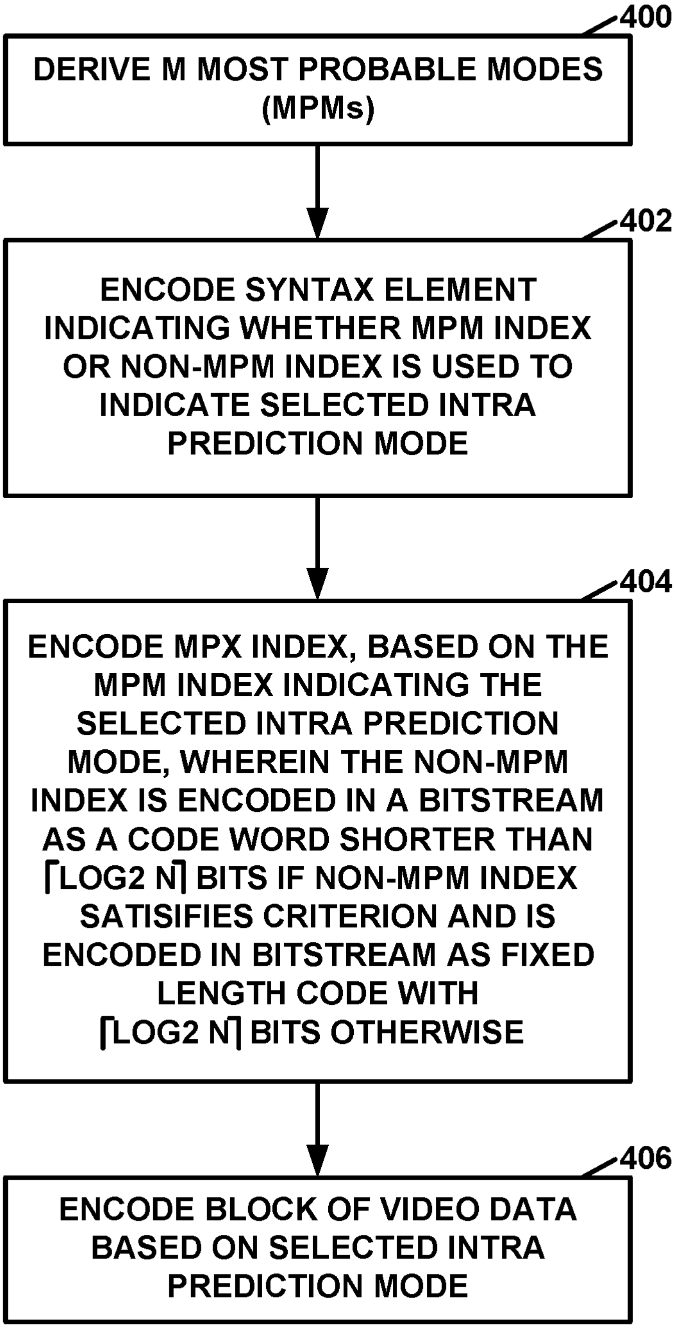
400
DERIVE M MOST PROBABLE MODES (MPMs)
402
ENCODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
404
ENCODE MPX INDEX, BASED ON THE MPM INDEX INDICATING THE SELECTED INTRA PREDICTION MODE, WHEREIN THE NON-MPM INDEX IS ENCODED IN A BITSTREAM AS A CODE WORD SHORTER THAN ⌈LOG2 N⌉ BITS IF NON-MPM INDEX SATISIFIES CRITERION AND IS ENCODED IN BITSTREAM AS FIXED LENGTH CODE WITH ⌈LOG2 N⌉ BITS OTHERWISE
406
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 14

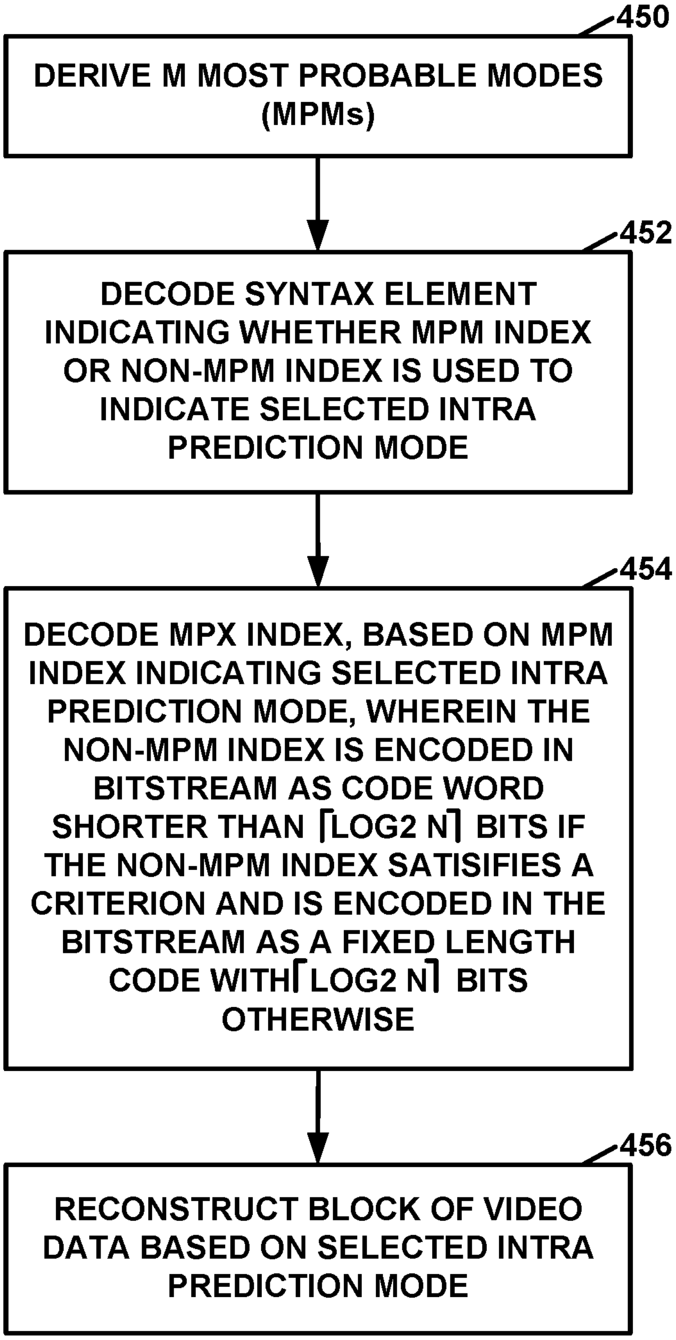
450
DERIVE M MOST PROBABLE MODES (MPMs)
452
DECODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
454
DECODE MPX INDEX, BASED ON MPM INDEX INDICATING SELECTED INTRA PREDICTION MODE, WHEREIN THE NON-MPM INDEX IS ENCODED IN BITSTREAM AS CODE WORD SHORTER THAN ⌈LOG2 N⌉ BITS IF THE NON-MPM INDEX SATISIFIES A CRITERION AND IS ENCODED IN THE BITSTREAM AS A FIXED LENGTH CODE WITH⌈LOG2 N⌉ BITS OTHERWISE
456
RECONSTRUCT BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 15

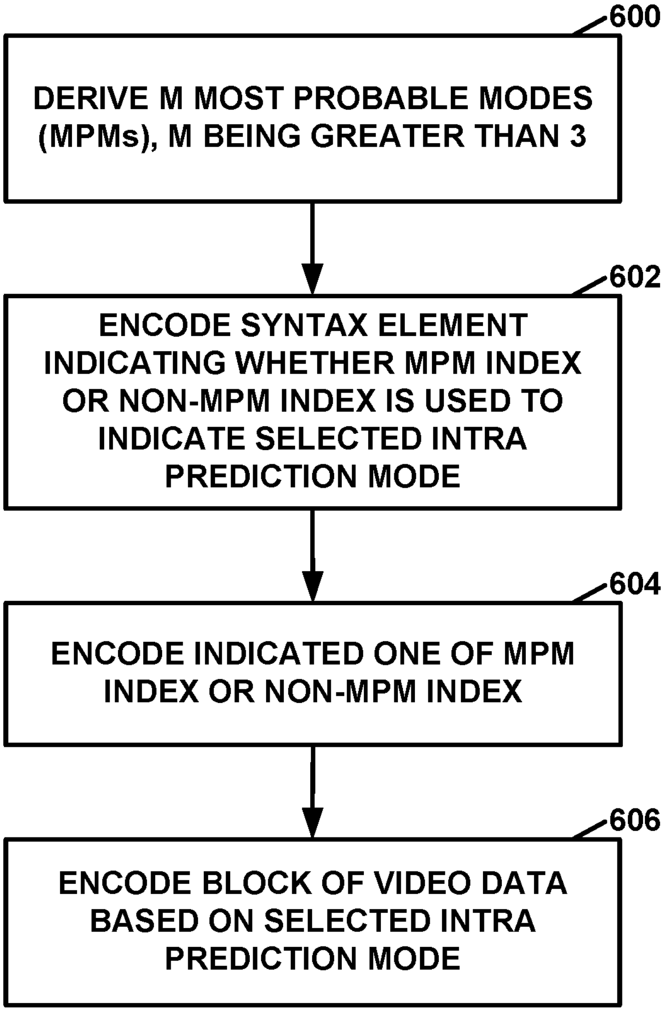
600
DERIVE M MOST PROBABLE MODES (MPMs), M BEING GREATER THAN 3
602
ENCODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
604
ENCODE INDICATED ONE OF MPM INDEX OR NON-MPM INDEX
606
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 16

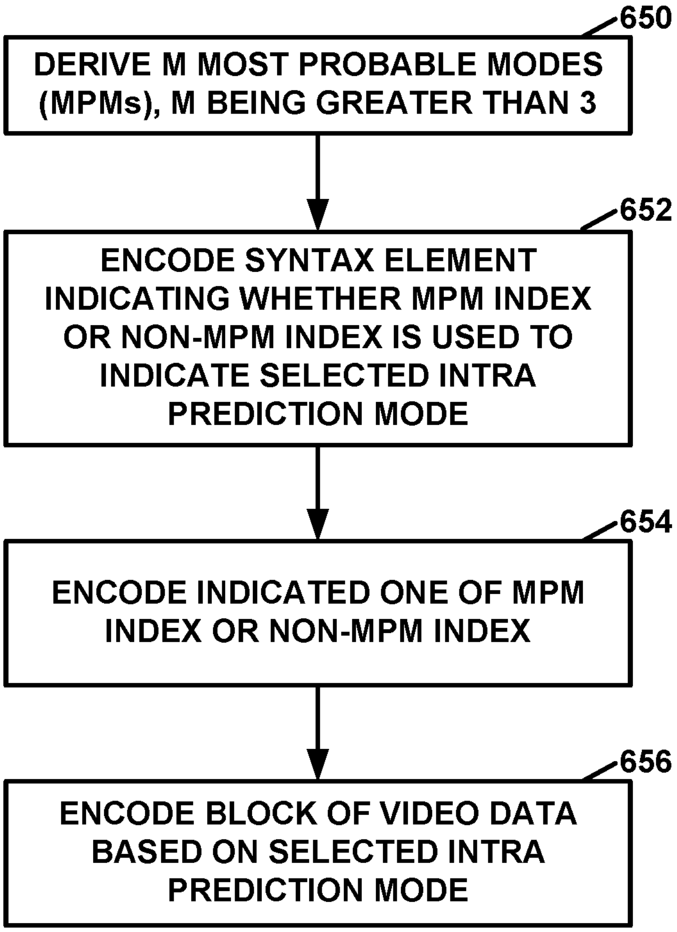
650
DERIVE M MOST PROBABLE MODES (MPMs), M BEING GREATER THAN 3
652
ENCODE SYNTAX ELEMENT INDICATING WHETHER MPM INDEX OR NON-MPM INDEX IS USED TO INDICATE SELECTED INTRA PREDICTION MODE
654
ENCODE INDICATED ONE OF MPM INDEX OR NON-MPM INDEX
656
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 17

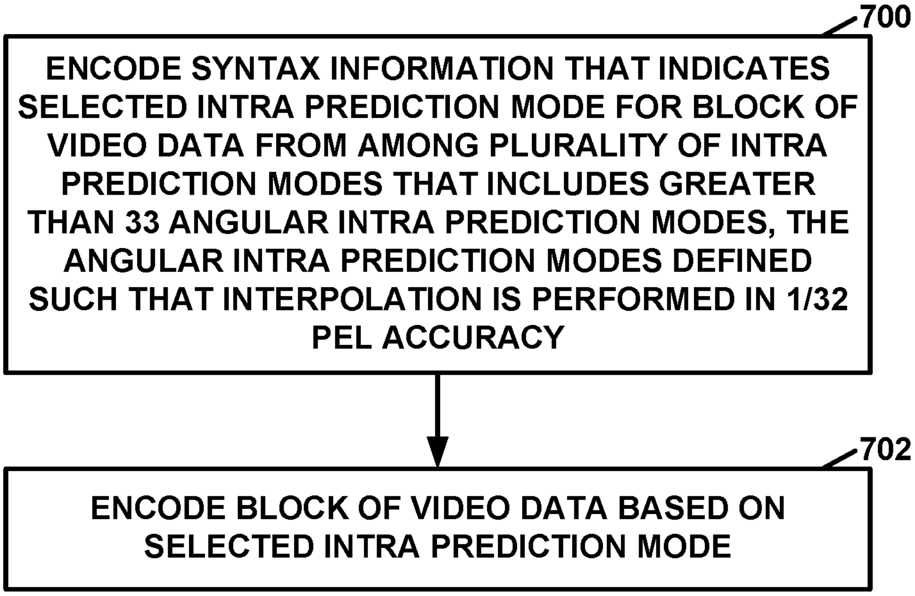
700
ENCODE SYNTAX INFORMATION THAT INDICATES SELECTED INTRA PREDICTION MODE FOR BLOCK OF VIDEO DATA FROM AMONG PLURALITY OF INTRA PREDICTION MODES THAT INCLUDES GREATER THAN 33 ANGULAR INTRA PREDICTION MODES, THE ANGULAR INTRA PREDICTION MODES DEFINED SUCH THAT INTERPOLATION IS PERFORMED IN 1/32 PEL ACCURACY
702
ENCODE BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 18

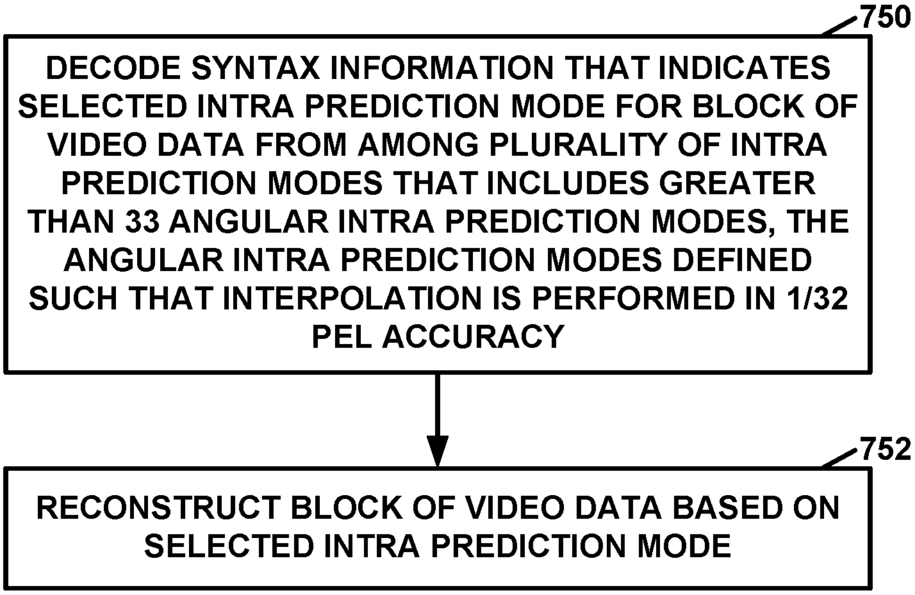
750
DECODE SYNTAX INFORMATION THAT INDICATES SELECTED INTRA PREDICTION MODE FOR BLOCK OF VIDEO DATA FROM AMONG PLURALITY OF INTRA PREDICTION MODES THAT INCLUDES GREATER THAN 33 ANGULAR INTRA PREDICTION MODES, THE ANGULAR INTRA PREDICTION MODES DEFINED SUCH THAT INTERPOLATION IS PERFORMED IN 1/32 PEL ACCURACY
752
RECONSTRUCT BLOCK OF VIDEO DATA BASED ON SELECTED INTRA PREDICTION MODE

FIG. 19

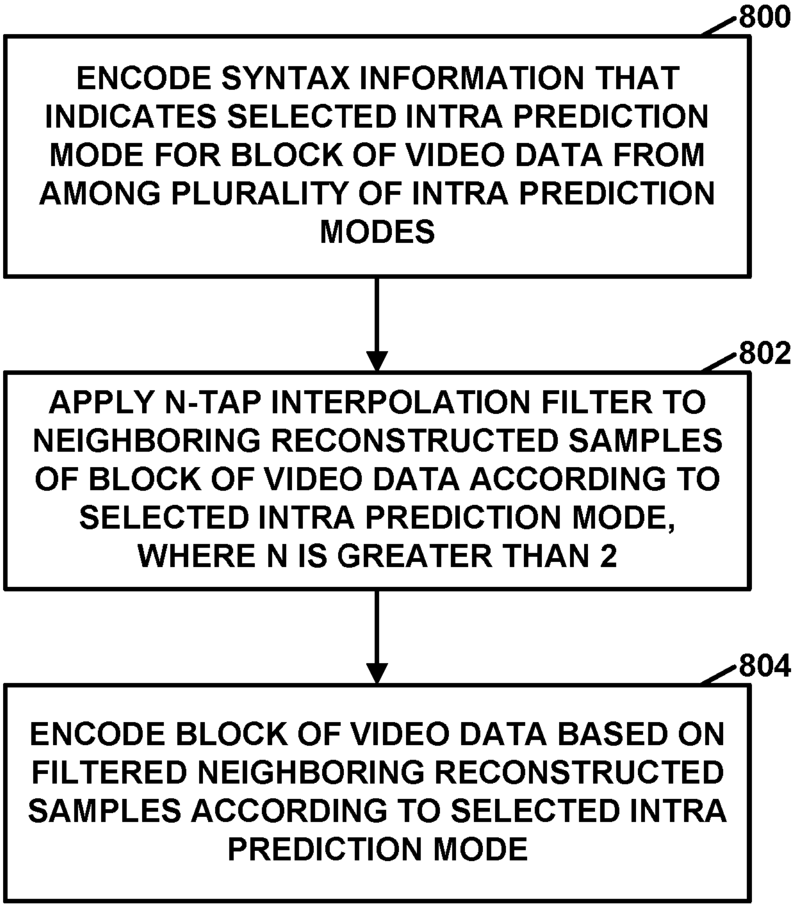
800
ENCODE SYNTAX INFORMATION THAT INDICATES SELECTED INTRA PREDICTION MODE FOR BLOCK OF VIDEO DATA FROM AMONG PLURALITY OF INTRA PREDICTION MODES
802
APPLY N-TAP INTERPOLATION FILTER TO NEIGHBORING RECONSTRUCTED SAMPLES OF BLOCK OF VIDEO DATA ACCORDING TO SELECTED INTRA PREDICTION MODE, WHERE N IS GREATER THAN 2
804
ENCODE BLOCK OF VIDEO DATA BASED ON FILTERED NEIGHBORING RECONSTRUCTED SAMPLES ACCORDING TO SELECTED INTRA PREDICTION MODE

FIG. 20

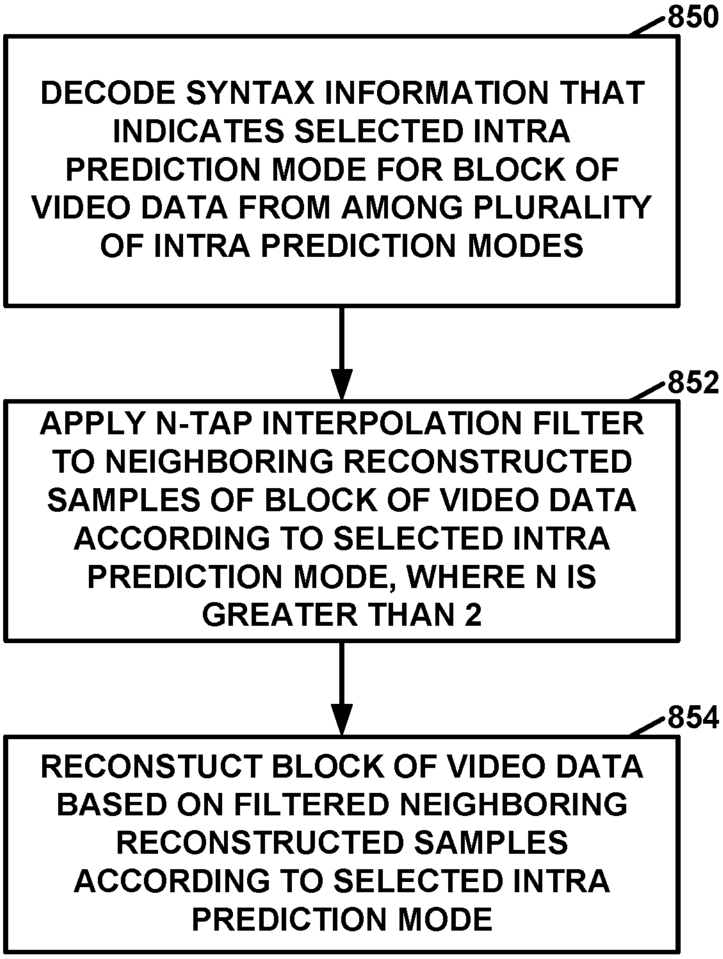
850
DECODE SYNTAX INFORMATION THAT INDICATES SELECTED INTRA PREDICTION MODE FOR BLOCK OF VIDEO DATA FROM AMONG PLURALITY OF INTRA PREDICTION MODES
852
APPLY N-TAP INTERPOLATION FILTER TO NEIGHBORING RECONSTRUCTED SAMPLES OF BLOCK OF VIDEO DATA ACCORDING TO SELECTED INTRA PREDICTION MODE, WHERE N IS GREATER THAN 2
854
RECONSTUCT BLOCK OF VIDEO DATA BASED ON FILTERED NEIGHBORING RECONSTRUCTED SAMPLES ACCORDING TO SELECTED INTRA PREDICTION MODE

FIG. 21

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 31294611, Maani, Ehsan **[0076]**

Non-patent literature cited in the description

- **VIVIENNE SZE.** High Efficiency Video Coding (HEVC), Algorithms and Architectures **[0005]**
- **ZHAO et al.** Fast Mode Decision Algorithm for Intra Prediction in HEVC. *Proceedings the IEEE Conference on Visual Communication and Image Processing (VCIP),* 2011 **[0005]**
- Improved intra angular prediction by DCT-based interpolation filter. **MATSUO et al.** Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European. IEEE, 2012, 1568-1572 **[0076]**
- **M. GUO ; X. GUO ; S. LEI.** Improved Intra Mode Coding. *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th Meeting: Daegu, Korea,* 20 January 2011 **[0076]**