a motion vector of the prediction unit, that is, the updated predicted motion vector is a predicted motion vector of the coding unit, and obtains, based on the updated predicted motion vector and the prediction unit, the picture information corresponding to the coding unit.

[0006] In a process of implementing this application, the inventor finds that a related technology has at least the following disadvantages: A decoder may simultaneously decode a plurality of coding units during decoding. However, when a predicted motion vector of a current coding unit needs to be used for another coding unit, decoding can be performed only after an optimal predicted motion vector of the current coding unit is updated, and consequently a delay is very likely to occur.

SUMMARY

[0007] Embodiments of this application provide a motion vector obtaining method and apparatus, a computer device, and a storage medium, to resolve a decoding delay problem in a related technology.

[0008] According to an aspect, this application provides a motion vector obtaining method, including:

determining a reference block of a to-be-processed picture block, where the reference block and the to-be-processed picture block are located in a same frame of picture;

when the reference block falls within a preset range, obtaining a motion vector of the to-be-processed picture block based on an initial motion vector of the reference block, where the preset range is determined based on a location of the to-be-processed picture block; and

when the reference block falls beyond the preset range, obtaining a motion vector of the to-be-processed picture block based on a final motion vector of the reference block, where the final motion vector is obtained based on the initial motion vector.

[0009] In a possible implementation, that the reference block falls within a preset range includes: a coding tree block CTB in which the reference block is located and a coding tree block in which the to-be-processed picture block is located are located in a same row; and

correspondingly, that the reference block falls beyond the preset range includes: the coding tree block in which the reference block is located and the coding tree block in which the to-be-processed picture block is located are located in different rows.

14. The communications medium may include any wireless or wired communications medium, for example, a radio frequency spectrum or at least one physical transmission lines. The communications medium may constitute a part of a packet-based network (for example, a local area network, a wide area network, or a global network such as the internet). The communications medium may include a router, a switch, a base station, or any other device that may be used to facilitate communication from the source apparatus 12 to the destination apparatus 14.

[0082] Alternatively, the encoded data may be output from an output interface 22 to a storage apparatus. Similarly, the encoded data may be accessed from the storage apparatus through an input interface. The storage apparatus may include any one of a plurality of distributed data storage media or locally accessed data storage media, for example, a hard drive, a Blu-ray, a digital versatile disc (digital versatile disc, DVD), a compact disc read-only memory (compact disc read-only memory, CD-ROM), a flash memory, a volatile or nonvolatile memory, or any other appropriate digital storage medium configured to store the encoded video data. In another feasible implementation, the storage apparatus may correspond to a file server or another intermediate storage apparatus that can keep an encoded video generated by the source apparatus 12. The destination apparatus 14 may access stored video data from the storage apparatus through streaming transmission or downloading. The file server may be any type of server that can store the encoded video data and transmit the encoded video data to the destination apparatus 14. In a feasible implementation, the file server includes a website server, a file transfer protocol server, a network-attached storage apparatus, or a local disk drive. The destination apparatus 14 may access the encoded video data through any standard data connection including an internet connection. The data connection may include a wireless channel (for example, wireless fidelity (wireless-fidelity, Wi-Fi) connection) or a wired connection (for example, a cable modem) that is suitable for accessing the encoded video data stored in the file server, or a combination thereof. Transmission of the encoded video data from the storage apparatus may be streaming transmission, downloading transmission, or a combination thereof.

[0083] The technologies in this application are not necessarily limited to wireless applications or settings. The technologies can be applied to video decoding, to support any one of a plurality of multimedia applications, for example, over-the-air television broadcasting, cable television transmission, satellite television transmission, video streaming transmission (for example, through the internet), encoding of a digital video for storage in a data storage medium, decoding of a digital

to an interface of an external display apparatus. In other feasible implementations, the destination apparatus 14 may be a display apparatus. Generally, the display apparatus 32 displays decoded video data to a user, and may include any one of a plurality of display apparatuses, for example, a liquid crystal display, a plasma display, an organic light-emitting diode display, or a display apparatus of another type.

[0088] The video encoder 20 and the video decoder 30 may operate according to, for example, a next-generation video coding compression standard (H.266) that is currently being developed, and may comply with an H.266 test model.

[0089] Alternatively, the video encoder 20 and the video decoder 30 may operate according to, for example, other dedicated or industrial standards such as the ITU-T H.265 standard or the ITU-T H.264 standard, or extensions of these standards. The ITU-T H.265 standard is also referred to as a high efficiency video decoding standard, and the ITU-T H.264 standard is alternatively referred to as moving picture experts group (moving picture experts group, MPEG-4) Part 10, or advanced video coding (advanced video coding, AVC). However, the technologies in this application are not limited to any specific decoding standard. In other feasible implementations, video compression standards include MPEG-2 and ITU-T H.263.

[0090] Although not shown in FIG. 1, in some aspects, the video encoder 20 and the video decoder 30 may be respectively integrated with an audio encoder and an audio decoder, and may include an appropriate multiplexer-demultiplexer (multiplexer-demultiplexer, MUX-DEMUX) unit or other hardware and software, to encode both an audio and a video in a same data stream or separate data streams. If applicable, in some feasible implementations, the MUX-DEMUX unit may comply with the ITU H.223 multiplexer protocol or other protocols such as the user datagram protocol (user datagram protocol, UDP).

[0091] The video encoder 20 and the video decoder 30 each may be implemented as any one of a plurality of appropriate encoder circuits, for example, one or more microprocessors, digital signal processors (digital signal processor, DSP), application-specific integrated circuits (application specific integrated circuit, ASIC), field programmable gate arrays (field programmable gate array, FPGA), discrete logic, software, hardware, firmware, or any combination thereof. When some of the technologies are implemented as software, an apparatus may store an instruction for the software in an appropriate non-transitory computer-readable medium, and execute the instruction in a form of hardware by using one or more processors, to

implement the technologies in this application. The video encoder 20 and the video decoder 30 each may be included in one or more encoders or decoders. Either of the video encoder 20 and the video decoder 30 may be integrated as a part of a combined encoder/decoder (encoder/decoder, CODEC) in a corresponding apparatus.

[0092] This application may, for example, relate to another apparatus in which the video encoder 20 "signals" specific information to, for example, the video decoder 30. However, it should be understood that the video encoder 20 may associate specific syntax elements with encoded parts of the video data, to signal information. That is, the video encoder 20 may store the specific syntax elements in header information of the encoded parts of the video data, to signal the data. In some applications, these syntax elements may be encoded and stored (for example, stored in a storage system or a file server) before being received and decoded by the video decoder 30. Therefore, the term "signal" may mean, for example, transmission of syntax or transmission of other data used for decoding compressed video data, regardless of whether the transmission is performed in real time, nearly in real time, or within a time span. For example, the transmission may be performed when a syntax element is stored in a medium during encoding, and then the syntax element may be retrieved by a decoding apparatus at any time after being stored in the medium.

[0093] The joint video experts team (joint video team, JVT) developed the H.265 high efficiency video coding (high efficiency video coding, HEVC) standard. HEVC standardization is based on an evolved model of a video decoding apparatus, where the model is referred to as an HEVC test model (HM). A latest H.265 standard document is available at http://www.itu.int/rec/T-REC-H.265. A latest version of the standard document is H.265 (12/16), and the standard document is incorporated herein by reference in its entirety. In the HM, it is assumed that the video decoding apparatus has several additional capabilities relative to an existing algorithm of ITU-TH.264/AVC. For example, H.264 provides nine intra prediction encoding modes, whereas the HM can provide up to 35 intra prediction encoding modes.

[0094] The JVET is committed to developing the H.266 standard. An H.266 standardization process is based on an evolved model of the video decoding apparatus, where the model is referred to as an H.266 test model. H.266 algorithm descriptions are available at http://phenix.int-evry.fr/jvet, and latest algorithm descriptions are included in JVET-G1001-v1. The algorithm description document is incorporated herein by reference in its entirety. In addition, reference software for a

TUs. A size of a TU is usually set based on a size of a PU within a given CU defined for a partitioned LCU. However, a case may not always be like this. The size of the TU is usually the same as or less than that of the PU. In some feasible implementations, a quadtree structure referred to as a "residual quadtree" (residual quadtree, RQT) may be used to split a residual sample corresponding to the CU into smaller units. A leaf node of the RQT may be referred to as a TU. A pixel difference associated with the TU may be transformed to generate a transform coefficient, and the transform coefficient may be quantized.

[0099] Generally, a PU includes data related to a prediction process. For example, when the PU is encoded in an intra mode, the PU may include data describing the intra prediction mode of the PU. In another feasible implementation, when the PU is encoded in an inter mode, the PU may include data defining a motion vector for the PU. For example, the data defining the motion vector for the PU may describe a horizontal component of the motion vector, a vertical component of the motion vector, a resolution (for example, 1/4 pixel precision or 1/8 pixel precision) of the motion vector, a reference picture to which the motion vector points, and/or a reference picture list (for example, a list 0, a list 1, or a list C) of the motion vector.

[00100] Generally, transform and quantization processes are used for a TU. A given CU including one or more PUs may also include one or more TUs. After performing prediction, the video encoder 20 may calculate a residual corresponding to the PU. The residual includes a pixel difference. The pixel difference may be transformed into a transform coefficient, and the transform coefficient is quantized and is scanned by using a TU, to generate a serialized transform coefficient for entropy-decoding. In this application, the term "video block" is usually used to indicate a decoding node of a CU. In some specific applications, in this application, the term "video block" may also be used to indicate a tree block including a decoding node, a PU, and a TU, for example, an LCU or a CU.

[00101] A video sequence usually includes a series of video frames or pictures. For example, a group of pictures (group of pictures, GOP) includes a series of video pictures, or one or more video pictures. The GOP may include syntactic data in header information of the GOP, header information of one or more pictures, or the like, and the syntactic data describes a quantity of pictures included in the GOP. Each slice of a picture may include slice syntactic data describing an encoding mode of the corresponding picture. The video encoder 20 usually performs an operation on a video block in an individual video slice, to encode the video data. The video block

20 may quantize the transform coefficients. Quantization refers to, for example, a process of quantizing the coefficients, to reduce an amount of data used for representing the coefficients and implement further compression. The quantization process can reduce a bit depth associated with some or all of the coefficients. For example, during quantization, an n-bit value may be reduced to an m-bit value through rounding, where n is greater than m.

[00106] The JEM model further improves a video picture encoding structure. Specifically, a block coding structure referred to as a "quadtree plus binary tree" (quad tree plus binary tree, QTBT) structure is introduced. Without using concepts such as a CU, a PU, and a TU in HEVC, the QTBT structure supports more flexible CU split shapes. A CU may be in a square shape or a rectangular shape. Quadtree partitioning is first performed on a CTU, and binary tree partitioning is further performed on a leaf node of a quadtree. In addition, there are two binary tree partitioning modes: symmetric horizontal partitioning and symmetric vertical partitioning. A leaf node of a binary tree is referred to as a CU. The CU in the JEM model cannot be further partitioned during prediction and transform. In other words, the CU, the PU, and the TU in the JEM model have a same block size. In the existing JEM model, a maximum CTU size is 256×256 luma pixels.

[00107] In some feasible implementations, the video encoder 20 may scan the quantized transform coefficient in a predefined scanning order to generate a serialized vector that can be entropy-encoded. In other feasible implementations, the video encoder 20 may perform adaptive scanning. After scanning the quantized transform coefficient to form a one-dimensional vector, the video encoder 20 may entropy-encode the one-dimensional vector based on a context-adaptive variable-length coding (context-adaptive variable-length coding, CAVLC) method, a context-based adaptive binary arithmetic coding (context-based adaptive binary arithmetic coding, CABAC) method, a probability interval partitioning entropy (probability interval partitioning entropy, PIPE) coding method, or another entropy coding method. The video encoder 20 may further entropy-encode the syntax element associated with the encoded video data, so that the video decoder 30 decodes the video data.

[00108] To perform CABAC, the video encoder 20 may assign a context in a context model to a to-be-transmitted symbol. The context may be related to whether a neighboring value of the symbol is non-zero. To perform CAVLC, the video encoder 20 may select a variable-length code of the to-be-transmitted symbol. A codeword in variable-length coding (variable-length coding, VLC) may be constructed, so that a shorter code corresponds to a more probable symbol and a

space to store an MVD, that is, store a difference between a final motion vector and an initial motion vector for a to-be-processed PU. In some embodiments, the reference picture storage 64 may further store a final motion vector for a to-be-processed PU. In this embodiment of this application, the MVD stored in the reference picture storage 64 may be used in an encoding process of another PU. For example, in a decoding process in a merge (merge) mode, an unupdated initial motion vector and an updated final motion vector are separately stored at different times. Therefore, when the to-be-processed PU needs to obtain a predicted motion vector for another PU, under a specific condition, the to-be-processed PU may directly use an unupdated initial motion vector for the another PU as the motion vector for the another PU, and does not need to obtain the motion vector for the another PU after the motion vector for the another PU is updated.

[00126] It should be noted that the target storage space may be provided by a memory other than the reference picture storage 64, for example, a newly added memory. This is not specifically limited in this embodiment of this application.

[00127] It should be noted that, in a related technology, when the video encoder 20 works in the merge mode, content in the reference picture storage 64 is empty, or is a zero vector, that is, when the video encoder 20 works in the merge mode, the target storage space is not used during encoding, and in an encoding process, the predicted motion vector for the another PU that is obtained by the to-be-processed PU is an updated final motion vector for the another PU. Therefore, the to-be-processed PU can obtain the predicted motion vector for the another PU only after the predicted motion vector for the another PU is updated, and consequently an encoding delay occurs during encoding. However, when the video encoder 20 works in a non-merge mode, the reference picture storage 64 may be used. For example, in an AMVP mode, the target storage space is used to store a motion vector residual, and the video encoder 20 encodes the motion vector residual, so that after decoding a current PU, a video decoder 30 may obtain a motion vector residual of a to-be-processed PU, and obtain a final motion vector for the to-be-processed PU based on the motion vector residual of the to-be-processed PU and an initial motion vector for the to-be-processed PU.

[00128] The prediction unit 41 includes a motion estimation unit 42, a motion compensation unit 44, and an intra prediction unit 46. For video block reconstruction, the video encoder 20 further includes an inverse quantization unit 58, an inverse transform unit 60, and a summator 62. The video encoder 20 may further include a deblocking filter (not shown in FIG. 2) to perform filtering on a block boundary, to remove a blocking artifact from a reconstructed video. When

necessary, the deblocking filter usually performs filtering on an output of the summator 62. In addition to the deblocking filter, an additional loop filter (in or after a loop) may also be used.

[00129] As shown in FIG. 2, the video encoder 20 receives video data, and the partitioning unit 35 partitions the data into video blocks. Such partitioning may further include partitioning into slices, picture blocks, or other larger units, and video block partitioning based on quadtree structures of an LCU and a CU. For example, the video encoder 20 is a component for encoding a video block in a to-be-encoded video slice. Usually, one slice may be partitioned into a plurality of video blocks (and may be partitioned into sets of video blocks that are referred to as picture blocks).

[00130] The prediction unit 41 may select one of a plurality of possible decoding modes, for example, one of a plurality of intra decoding modes or one of a plurality of inter decoding modes, for a current video block based on encoding quality and a cost calculation result (for example, a rate-distortion cost (rate distortion cost, RDcost)). The prediction unit 41 may provide an obtained intra-decoded or inter-decoded block to the summator 50 to generate residual block data, and provide the obtained intra-decoded or inter-decoded block to the summator 62 to reconstruct an encoded block and use the reconstructed encoded block as a reference picture.

[00131] The motion estimation unit 42 and the motion compensation unit 44 in the prediction unit 41 perform inter prediction decoding on the current video block relative to one or more prediction blocks of one or more reference pictures, to provide temporal compression. The motion estimation unit 42 may be configured to determine an inter prediction mode for the video slice based on a preset mode of the video sequence. In the preset mode, the video slice in the sequence may be specified as a P slice, a B slice, or a GPB slice. The motion estimation unit 42 and the motion compensation unit 44 may be highly integrated, but are described separately to explain concepts. Motion estimation performed by the motion estimation unit 42 is a process of generating a motion vector for estimating the video block. For example, the motion vector may indicate a displacement of a PU of a video block of a current video frame or picture relative to a prediction block of a reference picture.

[00132] The prediction block is a block of a PU that is found, based on a pixel difference, to be closely matched with a to-be-decoded video block, and the pixel difference may be determined by using a sum of absolute differences (sum of absolute differences, SAD), a sum of squared differences (sum of squared differences, SSD), or another difference metric. In some feasible

the reference picture list 0 and the reference picture list 1. In some other feasible implementations, bidirectional prediction is prediction separately performed based on a reconstructed future frame and a reconstructed past frame that are of a current frame in a display order. When the motion estimation unit 42 performs unidirectional prediction on the PU, the motion estimation unit 42 may search the reference picture in the list 0 or the list 1 for a reference block for the PU. Then, the motion estimation unit 42 may generate a reference index indicating a reference picture including the reference block in the list 0 or the list 1, and a motion vector indicating a spatial displacement between the PU and the reference block. The motion estimation unit 42 may output the reference index, a prediction direction identifier, and the motion vector as motion information for the PU. The prediction direction identifier may indicate that the reference index indicates the reference picture in the list 0 or the list 1. The motion compensation unit 44 may generate a prediction picture block for the PU based on the reference block indicated by the motion information for the PU.

[00137] When the motion estimation unit 42 performs bidirectional prediction on the PU, the motion estimation unit 42 may search the reference picture in the list 0 for a reference block for the PU, and may further search the reference picture in the list 1 for another reference block for the PU. Then, the motion estimation unit 42 may generate reference indexes indicating reference pictures including the reference blocks in the list 0 and the list 1, and motion vectors indicating spatial displacements between the reference blocks and the PU. The motion estimation unit 42 may output the reference indexes and the motion vectors for the PU as motion information for the PU. The motion compensation unit 44 may generate a prediction picture block for the PU based on the reference blocks indicated by the motion information for the PU.

[00138] In some feasible implementations, the motion estimation unit 42 does not output a complete set of the motion information for the PU to the entropy encoding unit 56. Instead, the motion estimation unit 42 may signal the motion information for the PU with reference to motion information for another PU. For example, the motion estimation unit 42 may determine that the motion information for the PU is similar to motion information for a neighboring PU. In this implementation, the motion estimation unit 42 may indicate an indicator value in a syntax structure associated with the PU, and the indicator value indicates, to the video decoder 30, that the motion information for the PU is the same as the motion information for the neighboring PU, or may be derived from the motion information for the neighboring PU. In another implementation, the

motion estimation unit 42 may identify, in the syntax structure associated with the PU, a candidate predicted motion vector and a motion vector difference (motion vector difference, MVD) that are associated with the neighboring PU. The MVD indicates a difference between the motion vector for the PU and the indicated candidate predicted motion vector associated with the neighboring PU. The video decoder 30 may determine the motion vector for the PU by using the indicated candidate predicted motion vector and the MVD.

[00139] As described above, the prediction unit 41 may generate a candidate predicted motion vector list for each PU of a CU. One or more candidate predicted motion vector lists may include one or more original candidate predicted motion vectors and one or more additional candidate predicted motion vectors derived from the one or more original candidate predicted motion vectors.

[00140] It should be noted that, in this embodiment of this application, when establishing the candidate predicted motion vector list in the merge mode, the prediction unit 41 may store a motion vector for a reference PU of the to-be-processed PU in the candidate predicted motion vector list, the motion vector for the reference PU may be an unupdated initial motion vector for the reference PU, or may be an updated final motion vector for the reference PU. The updated final motion vector for the reference PU may be obtained based on a motion vector stored in the target storage space. When the target storage space stores an MVD, the final motion vector for the reference PU may be obtained by adding the initial motion vector for the reference PU and the MVD stored in the target storage space. When the target storage space stores the final motion vector for the reference PU, the final motion vector for the reference PU may be directly obtained from the target storage space. The motion vector for the reference PU may be stored in the candidate predicted motion vector list in the following manner: When the reference PU and the current PU are in a same CTB or CTB row range, the unupdated initial motion vector for the reference unit is stored in the candidate predicted motion vector list; or when the reference PU and the current PU are not in a same CTB or CTB row range, the updated final motion vector for the reference unit is stored in the candidate predicted motion vector list.

[00141] The intra prediction unit 46 in the prediction unit 41 may perform intra prediction decoding on the current video block relative to one or more neighboring blocks that are in a same picture or slice as the to-be-decoded current block, to provide spatial compression. Therefore, as an alternative of inter prediction (as described above) performed by the motion estimation unit 42 and the motion compensation unit 44, the intra prediction unit 46 may perform intra prediction on

than the reference picture storage 92, for example, a newly added memory. This is not specifically limited in this embodiment of this application.

[00150] It should be noted that, in a related technology, when the video decoder 30 works in the merge mode, content in the target storage space is empty, or is a zero vector, that is, when the video decoder 30 works in the merge mode, the target storage space is not used during decoding, and during decoding, the predicted motion vector for the another PU that is obtained by the to-be-processed PU is an updated final motion vector for the another PU. Therefore, the to-be-processed PU can obtain the predicted motion vector for the another PU only after the predicted motion vector for the another PU is updated, and consequently a decoding delay occurs during decoding. However, when the video decoder 30 works in a non-merge mode, the target storage space may be used. For example, in an AMVP mode, the video decoder 30 may obtain an MVD for a to-be-processed PU through decoding, so that the video decoder 30 can obtain a final motion vector for the to-be-processed PU based on the initial motion vector and the MVD for the to-be-processed PU.

[00151] The prediction unit 81 includes a motion compensation unit 82 and an intra prediction unit 84. In some feasible implementations, the video decoder 30 may perform an example decoding process inverse to the encoding process described with respect to the video encoder 20 in FIG. 4.

[00152] During decoding, the video decoder 30 receives, from the video encoder 20, an encoded video bitstream that represents a video block of an encoded video slice and associated syntax elements. The entropy decoding unit 80 of the video decoder 30 entropy decodes the bitstream to generate a quantized coefficient, a motion vector, and other syntax elements. The entropy decoding unit 80 transfers the motion vector and the other syntax elements to the prediction unit 81. The video decoder 30 may receive the syntax elements at a video slice level and/or a video block level.

[00153] When the video slice is decoded into an intra-decoded (I) slice, the intra prediction unit 84 of the prediction unit 81 may generate prediction data of the video block of the current video slice based on a signaled intra prediction mode and data of a previously decoded block of a current frame or picture.

[00154] When the video picture is decoded into an inter-decoded slice (for example, a B slice, a P slice, or a GPB slice), the motion compensation unit 82 of the prediction unit 81 generates a prediction block of the video block of the current video picture based on the motion vector and the other syntax elements that are received from the entropy decoding unit 80. The prediction block

predicted motion vector list, the predicted motion vector for the reference PU may be an unupdated initial motion vector for the reference PU, or may be an updated final motion vector for the reference PU. The updated final motion vector for the reference PU may be obtained based on a motion vector for the reference PU that is stored in the target storage space. For example, when the target storage space stores an MVD, the final motion vector for the reference PU may be obtained by adding the initial motion vector for the reference PU and the MVD stored in the target storage space. When the target storage space stores the final motion vector for the reference PU, the final motion vector for the reference PU may be directly obtained from the target storage space. The predicted motion vector for the reference PU may be stored in the candidate predicted motion vector list in the following manner: When the reference PU and the current PU are in a same CTB or CTB row range, the unupdated initial motion vector for the reference unit is stored in the candidate predicted motion vector list; or when the reference PU and the current PU are not in a same CTB or CTB row range, the updated final motion vector for the reference unit is stored in the candidate predicted motion vector list.

[00159] The inverse quantization unit 86 performs inverse quantization (for example, dequantization) on a quantized transform coefficient that is provided in the bitstream and decoded by the entropy decoding unit 80. The inverse quantization process may include: determining a quantization degree based on a quantization parameter calculated by the video encoder 20 for each video block in the video slice, and similarly determining a to-be-applied inverse quantization degree. The inverse transform unit 88 performs inverse transform (for example, inverse DCT, inverse integer transform, or a conceptually similar inverse transform process) on the transform coefficient to generate a residual block in pixel domain.

[00160] After the motion compensation unit 82 generates the prediction block of the current video block based on the motion vector and the other syntax elements, the video decoder 30 performs a summation operation on the residual block from the inverse transform unit 88 and the corresponding prediction block generated by the motion compensation unit 82, to form the decoded video block. The summator 90 is one or more components that perform the summation operation. When necessary, a deblocking filter may be used to perform filtering on the decoded block to remove a blocking artifact. Another loop filter (in or after a decoding loop) may also be used to smoothen pixels, or video quality may be improved in another manner. Then, a decoded video block in a given frame or picture is stored in the reference picture storage 92. The reference

(temporal motion vector predictor, TMVP).

[00166] After generating the candidate predicted motion vector list, the inter prediction module 121 may select one candidate predicted motion vector from the candidate predicted motion vector list. Then, the inter prediction module 121 may generate a prediction picture block for the PU based on a reference block indicated by the motion information for the PU. In the merge mode, the motion information for the PU may be the same as motion information indicated by the selected candidate predicted motion vector. FIG. 10 described below is a flowchart of an example of the merge mode.

[00167] After generating the prediction picture block for the PU through IME and FME and generating the prediction picture block for the PU through the merge operation, the inter prediction module 121 may select the prediction picture block generated by performing the FME operation or the prediction picture block generated by performing the merge operation. In some feasible implementations, the inter prediction module 121 may select the prediction picture block for the PU by analyzing rate-distortion costs of the prediction picture block generated by performing the FME operation and the prediction picture block generated by performing the merge operation.

[00168] After the inter prediction module 121 selects a prediction picture block for a PU generated by partitioning the current CU according to each partitioning pattern (in some implementations, after a coding tree unit CTU is partitioned into CUs, the CU is not further partitioned into smaller PUs, and in this case, the PU is equivalent to the CU), the inter prediction module 121 may select a partitioning pattern for the current CU. In some implementations, the inter prediction module 121 may select the partitioning pattern for the current CU by analyzing a rate-distortion cost of the selected prediction picture block of the PU generated by partitioning the current CU according to each partitioning pattern. The inter prediction module 121 may output a prediction picture block associated with a PU that belongs to the selected partitioning pattern to a residual generation module 102. The inter prediction module 121 may output a syntax element of motion information for the PU that belongs to the selected partitioning pattern to an entropy encoding module 116.

[00169] In the schematic diagram shown in FIG. 4, the inter prediction module 121 includes IME modules 180A to 180N (collectively referred to as an "IME module 180"), FME modules 182A to 182N (collectively referred to as an "FME module 182"), merge modules 184A to 184N (collectively referred to as a "merge module 184"), PU pattern decision-making modules 186A to

candidate predicted motion vector location 252A may be assigned to an index 0. If the candidate predicted motion vector location 252A is unavailable, the video decoder may not add the candidate predicted motion vector location 252A to the candidate predicted motion vector list. A candidate predicted motion vector location may be unavailable for various reasons. For example, if the candidate predicted motion vector location is not in a current picture, the candidate predicted motion vector location may be unavailable. In another feasible implementation, if the candidate predicted motion vector location undergoes intra prediction, the candidate predicted motion vector location may be unavailable. In another feasible implementation, if the candidate predicted motion vector location is in a slice different from a slice corresponding to the current CU, the candidate predicted motion vector location may be unavailable.

[00180] After considering the candidate predicted motion vector location 252A, the video decoder may consider the candidate predicted motion vector location 252B. If the candidate predicted motion vector location 252B is available and different from the candidate predicted motion vector location 252A, the video decoder may add the candidate predicted motion vector location 252B to the candidate predicted motion vector list. In this specific context, the term "same" or "different" means that candidate predicted motion vector locations are associated with same or different motion information. Therefore, if two candidate predicted motion vector locations have same motion information, the two candidate predicted motion vector locations are considered to be the same; or if two candidate predicted motion vector locations have different motion information, the two candidate predicted motion vector locations are considered to be different. If the candidate predicted motion vector location 252A is unavailable, the video decoder may assign the candidate predicted motion vector location 252B to the index 0. If the candidate predicted motion vector location 252A is available, the video decoder may assign the candidate predicted motion vector location 252B to an index 1. If the candidate predicted motion vector location 252B is unavailable or is the same as the candidate predicted motion vector location 252A, the video decoder skips the candidate predicted motion vector location 252B and does not add the candidate predicted motion vector location 252B to the candidate predicted motion vector list.

[00181] Similarly, the video decoder considers the candidate predicted motion vector location 252C to determine whether to add the candidate predicted motion vector location 252C to the list. If the candidate predicted motion vector location 252C is available and different from the candidate predicted motion vector locations 252B and 252A, the video decoder may assign the candidate

include five candidate predicted motion vectors (the first four spatial candidate predicted motion vectors considered in 602 and the temporal candidate predicted motion vector considered in 604), or may include four candidate predicted motion vectors (the first four spatial candidate predicted motion vectors considered in 602). If the candidate predicted motion vector list includes five candidate predicted motion vectors (yes in 608), that is, includes five candidates, the video decoder completes construction of the list.

[00183] If the candidate predicted motion vector list includes four candidate predicted motion vectors (no in 608), the video decoder may consider a fifth spatial candidate predicted motion vector (610). The fifth spatial candidate predicted motion vector may (for example) correspond to a candidate predicted motion vector location 252E. If the candidate predicted motion vector corresponding to the location 252E is available and different from the candidate predicted motion vectors corresponding to the locations 252A, 252B, 252C, and 252D, the video decoder may add the fifth spatial candidate predicted motion vector to the candidate predicted motion vector list, and assign the fifth spatial candidate predicted motion vector to the index 4. If the candidate predicted motion vector corresponding to the location 252E is unavailable or is the same as one of the candidate predicted motion vectors corresponding to the candidate predicted motion vector locations 252A, 252B, 252C, and 252D, the video decoder may not add the candidate predicted motion vector corresponding to the location 252 to the candidate predicted motion vector list. Therefore, after the fifth spatial candidate predicted motion vector is considered (610), the list may include five candidate predicted motion vectors (the first four spatial candidate predicted motion vectors considered in 602 and the fifth spatial candidate predicted motion vector considered in 610), or may include four candidate predicted motion vectors (the first four spatial candidate predicted motion vectors considered in 602).

[00184] If the candidate predicted motion vector list includes five candidates (yes in 612), that is, includes five candidate predicted motion vectors, the video decoder completes generation of the candidate predicted motion vector list. If the candidate predicted motion vector list includes four candidate predicted motion vectors (no in 612), the video decoder adds a manually generated candidate predicted motion vector (614) until the list includes five candidates (yes in 616), that is, includes five candidate predicted motion vectors.

[00185] If the list includes fewer than four spatial candidate predicted motion vectors (no in 604) after the video decoder considers the first four spatial candidate predicted motion vectors, the

[00202] After selecting the candidate predicted motion vector, the video encoder may output a candidate predicted motion vector index (208). The candidate predicted motion vector index is also a reference frame list index, and the candidate predicted motion vector index may indicate a location of a candidate predicted motion vector selected from the candidate predicted motion vector list. In some feasible implementations, the candidate predicted motion vector index may be represented as "merge_idx".

[00203] FIG. 11 is a flowchart of an example of an advanced motion vector prediction mode according to an embodiment of this application. A video encoder (for example, the video encoder 20) may perform an AMVP operation 210.

[00204] After the video encoder starts the AMVP operation 210, the video encoder may generate one or more motion vectors for a current PU (211). The video encoder may perform integer motion estimation and fraction motion estimation to generate the motion vector for the current PU. As described above, a current picture may be associated with two reference picture lists (a list 0 and a list 1). If uni-prediction is performed on the current PU, the video encoder may generate a list-0 motion vector or a list-1 motion vector for the current PU. The list-0 motion vector may indicate a spatial displacement between a picture block for the current PU and a reference block in a reference picture in the list 0. The list-1 motion vector may indicate a spatial displacement between the picture block for the current PU and a reference block in a reference picture in the list 1. If bi-prediction is performed on the current PU, the video encoder may generate the list-0 motion vector and the list-1 motion vector for the current PU.

[00205] After generating the one or more motion vectors for the current PU, the video encoder may generate a prediction picture block for the current PU (212). The video encoder may generate the prediction picture block for the current PU based on one or more reference blocks indicated by the one or more motion vectors for the current PU.

[00206] In addition, the video encoder may generate a candidate predicted motion vector list for the current PU (213). The video encoder may generate the candidate predicted motion vector list for the current PU in various manners. For example, the video encoder may generate the candidate predicted motion vector list for the current PU according to one or more feasible implementations described below with respect to FIG. 6 to FIG. 9. In some feasible implementations, when the video encoder generates the candidate predicted motion vector list in the AMVP operation 210, the candidate predicted motion vector list may be limited to two

vector of the to-be-processed picture block is used as the initial motion vector of the to-be-processed picture block.

[00221] The predicted motion vector of the to-be-processed picture block may be obtained according to the methods shown in FIG. 6 to FIG. 9 in the embodiments of this application or any one of existing predicted motion vector obtaining methods in the H.265 standard or a JEM reference mode. This is not limited. A motion vector difference may be obtained with reference to the to-be-processed picture block. Motion estimation is performed within a search range determined based on the predicted motion vector of the to-be-processed picture block, and a difference between a motion vector of the to-be-processed picture block that is obtained after the motion estimation and the predicted motion vector of the to-be-processed picture block is used as the motion vector difference.

[00222] During bidirectional prediction, this step specifically includes: obtaining a first initial motion vector of the to-be-processed picture block based on a forward predicted motion vector of the to-be-processed picture block, and obtaining a second initial motion vector of the to-be-processed picture block based on a backward predicted motion vector of the to-be-processed picture block.

[00223] 1302: Obtain a prediction block of the to-be-processed picture block based on the initial motion vector and one or more preset motion vector offsets. Details are as follows:

[00224] S13021: Obtain, from a reference frame of the to-be-processed picture block that is indicated by a reference frame index of the to-be-processed picture block, a picture block indicated by the initial motion vector of the to-be-processed picture block, to serve as a temporary prediction block of the to-be-processed picture block.

[00225] The initial motion vector includes a first initial motion vector and a second initial motion vector. The first initial motion vector indicates a first motion compensation block based on a first reference frame list of the to-be-processed picture block, and the first motion compensation block is a reference block for performing motion compensation on the to-be-processed picture block. The second initial motion vector indicates a second motion compensation block based on a second reference frame list of the to-be-processed picture block, and the second motion compensation block is a reference block for performing motion compensation on the to-be-processed picture block.

[00226] 13022: Add the initial motion vector and the one or more preset motion vector offsets

list index of the to-be-processed picture block, a first picture block indicated by the first initial motion vector of the to-be-processed picture block, where the first picture block is a forward reference block, and correspondingly, the first initial motion vector may be a forward initial motion vector, and obtaining, from a second reference frame list of the to-be-processed picture block that is indicated by a second reference frame list index of the to-be-processed picture block, a second picture block indicated by the second initial motion vector of the to-be-processed picture block, where the second picture block is a backward reference block, and correspondingly, the second initial motion vector may be a backward initial motion vector; performing weighted processing on the first picture block and the second picture block to obtain the temporary prediction block of the to-be-processed picture block; adding the first initial motion vector and the one or more preset motion vector offsets of the to-be-processed picture block to obtain one or more first final motion vectors, and adding the second initial motion vector and the one or more preset motion vector offsets of the to-be-processed picture block to obtain one or more second final motion vectors; obtaining one or more first candidate prediction blocks at one or more search locations indicated by the one or more first final motion vectors, and obtaining one or more second candidate prediction blocks at one or more search locations indicated by the one or more second final motion vectors; selecting, from the one or more first candidate prediction blocks, a candidate prediction block with a minimum pixel difference from the temporary prediction block as the first prediction block of the to-be-processed picture block, and selecting, from the one or more backward candidate prediction blocks, a candidate prediction block with a minimum pixel difference from the temporary prediction block as the second prediction block of the to-be-processed picture block; and performing weighted processing on the first prediction block and the second prediction block to obtain the prediction block of the to-be-processed picture block.

[00233] In some embodiments, another motion vector in a direction indicated by a candidate final motion vector that corresponds to a minimum distortion cost in a plurality of candidate final motion vectors may alternatively be used as a final motion vector of the prediction block. Specifically, the final motion vector of the prediction block may be selected based on a first preset condition.

[00234] In a possible implementation, the plurality of candidate final motion vectors may be a plurality of motion vectors in a plurality of directions. Specific precision in each direction may be first selected. For example, a candidate final motion vector with integer pixel precision is used as

vectors are located in four directions of the prediction block: a top direction, a bottom direction, a left direction, and a right direction. There are two candidate motion vectors in each direction. A candidate motion vector with a maximum length is first selected from each of the four directions to obtain the four first vectors. When the first vector in the top direction corresponds to a minimum distortion cost, the first vector in the top direction is used as the reference vector, that is, the final vector of the prediction block is a motion vector in a positive direction. It is assumed that the first preset condition is that a proportion relationship among the distortion cost A corresponding to the reference vector, the distortion cost C corresponding to the second vector, and the distortion cost B corresponding to the initial motion vector is 1, where the second vector is a candidate motion vector in an opposite direction to the reference vector, and the preset precision is 1/4 precision. In this case, when the distortion cost corresponding to the reference vector is 2, the distortion cost corresponding to the first vector in the bottom direction is 4, and the distortion cost corresponding to the initial motion vector is 3, it is assumed that a proportion relationship among the distortion cost corresponding to the reference vector, the distortion cost corresponding to the first vector in the bottom direction, and the distortion cost corresponding to the initial motion vector is (2×B)/(A+C), and the proportion relationship is 1. In this case, the reference vector is used as the final motion vector of the prediction block. If the distortion cost corresponding to the first vector in the bottom direction is 5, and the proportion relationship (2×B)/(A+C) is not 1, the proportion does not meet the first preset condition. In this case, if A is equal to B, a length of the reference vector is reduced by the preset precision 1/4 to obtain a third vector, and the third vector is used as the final motion vector of the prediction block. If A is not equal to B, when B is equal to C, the length of the reference vector is increased by the preset precision 1/4 to obtain a fourth vector, and the fourth vector is used as the final motion vector of the prediction block. When B is not equal to C, the loop count is set to 3, precision of the offset of the reference vector is first updated to 1/8 in each loop, and it is assumed that the second preset relationship among A, B, and C is K=|(A–C)×16|, and T=((A+C)–2×B)×8. When K is greater than or equal to T, the fourth preset condition is met, and the precision of the offset of the reference vector is updated to 1/16. It is assumed that K and T are updated in each loop, K=K–T, and T=T/2. When the loop count is reduced to 0, the loop ends. After the loop ends, the direction of the offset is determined based on the value relationship between A and C. When A is greater than or equal to C, the direction of the offset is a positive direction. When A is less than C, the direction of the offset is a negative direction. A fifth

vector is obtained based on the reference vector, the offset, and the direction of the offset, and the fifth vector is used as the first target vector of the prediction block. Finally, the first target vector is used as the final motion vector of the prediction block.

[00236] Certainly, in some embodiments, when a third preset relationship between the distortion cost corresponding to the reference vector and a distortion cost corresponding to a third vector in a target direction does not meet a fifth preset condition, the length of the first vector may be increased by the preset precision, until an increased distortion cost corresponding to a second target vector meets a second target condition. When the third preset relationship between the distortion cost corresponding to the reference vector and the distortion cost corresponding to the third vector in the target direction meets the fifth preset condition, the length of the first vector may be reduced by the preset precision, until a reduced distortion cost corresponding to the second target vector meets the target condition. The target direction is any direction other than a direction of the reference vector. The third preset relationship, the fifth preset condition, and the target condition are not specifically limited in this embodiment of the present invention.

[00237] In some embodiments, when a fourth preset relationship among the distortion cost corresponding to the reference vector, the distortion cost corresponding to the third vector in the target direction, and a distortion cost corresponding to an intermediate vector meets a sixth preset condition, a third target vector is determined by increasing a length of the reference vector by the preset precision. When the fourth preset relationship among the distortion cost corresponding to the reference vector, the distortion cost corresponding to the third vector in the target direction, and the distortion cost corresponding to the intermediate vector does not meet the sixth preset condition, the third target vector is determined by reducing the length of the reference vector by the preset precision. The intermediate vector is a motion vector located between the reference vector and the second vector. A process of increasing the reference vector by the preset precision and reducing the reference vector by the preset precision is described above, and details are not described herein again. The fourth preset relationship and the sixth preset condition are not specifically limited in this embodiment of the present invention.

[00238] FIG. 14 is a flowchart of an example of updating a motion vector during video decoding according to an embodiment of this application. A to-be-processed picture block is a to-be-decoded block.

[00239] 1401: Obtain an initial motion vector of the to-be-processed picture block based on a

predicted motion vector of the to-be-processed picture block that is indicated by an index.

[00240] In a feasible implementation, for example, in a merge mode, the predicted motion vector of the to-be-processed picture block is used as the initial motion vector of the to-be-processed picture block.

[00241] The predicted motion vector of the to-be-processed picture block may be obtained according to the methods shown in FIG. 6 to FIG. 9 in the embodiments of this application or any one of existing predicted motion vector obtaining methods in the H.265 standard or a JEM reference mode. This is not limited. The motion vector difference may be obtained by parsing a bitstream.

[00242] During bidirectional prediction, this step specifically includes: obtaining a first initial motion vector of the to-be-processed picture block based on a forward predicted motion vector of the to-be-processed picture block, and obtaining a second initial motion vector of the to-be-processed picture block based on a backward predicted motion vector of the to-be-processed picture block.

[00243] 1402: Obtain a prediction block of the to-be-processed picture block based on the initial motion vector and one or more preset motion vector offsets. Details are as follows:

[00244] S14021: Obtain, from a reference frame of the to-be-processed picture block that is indicated by a reference frame index of the to-be-processed picture block, a picture block indicated by the initial motion vector of the to-be-processed picture block, to serve as a temporary prediction block of the to-be-processed picture block.

[00245] 14022: Add the initial motion vector and the one or more preset motion vector offsets of the to-be-processed picture block to obtain one or more final motion vectors, where each final motion vector indicates a search location.

[00246] 14023: Obtain one or more candidate prediction blocks at one or more search locations indicated by the one or more final motion vectors, where each search location corresponds to one candidate prediction block.

[00247] 14024: Select, from the one or more candidate prediction blocks, a candidate prediction block with a minimum pixel difference from the temporary prediction block as the prediction block of the to-be-processed picture block.

[00248] During bidirectional prediction, this step specifically includes: obtaining, from a first reference frame of the to-be-processed picture block that is indicated by a first reference frame

6), and a picture order count (picture order count, POC) POC of the current decoding block is 4. The POC is used to indicate a picture display order, and POCs of reference pictures indicated by index values of displayed reference pictures of a picture are respectively 2 and 6. In this case, the POC corresponding to the current decoding block is 4, a POC corresponding to a forward prediction reference picture block is 2, and a POC corresponding to a backward prediction reference picture block is 6.

[00251] Forward prediction and backward prediction are separately performed on the current decoding block to obtain an initial forward prediction block (forward prediction block, FPB) and an initial backward prediction block (backward prediction block, BPB) of the current decoding block, and it is assumed that the initial forward decoding prediction block and the initial backward decoding prediction block are respectively FPB1 and BPB1. A first decoding prediction block (decoding prediction block, DPB) of the current decoding block is obtained by performing weighted summation on FPB1 and BPB1, and is assumed to be DPB1.

[00252] (–10, 4) and (5, 6) are used as reference inputs of the forward and backward motion vector predictors, and motion search with first precision is separately performed on the forward prediction reference picture block and the backward prediction reference picture block. In this case, the first precision is 1/2 pixel precision in a 1-pixel range. The first decoding prediction block DPB1 is used as a reference. Corresponding new forward and backward decoding prediction blocks obtained through each motion search are compared with the first decoding prediction block DPB1, to obtain a new decoding prediction block with a minimum difference from DPB1, and forward and backward motion vector predictors corresponding to the new decoding prediction block are used as target motion vector predictors, and are assumed to be (–11, 4) and (6, 6) respectively.

[00253] The target motion vector predictors are updated to (–11, 4) and (6, 6), forward prediction and backward prediction are separately performed on the first decoding block based on the target motion vector predictors, a target decoding prediction block is obtained by performing weighted summation on obtained new forward and backward decoding prediction blocks, and is assumed to be DPB2, and the decoding prediction block of the current decoding block is updated to DPB2.

[00254] It should be noted that, when the motion search with the first precision is performed on the forward prediction reference picture block and the backward prediction reference picture block,

12) and (8, 4), a POC of a picture in which the current coding block is located is 8, and POCs of reference pictures that are indicated by index values of the reference pictures are respectively 4 and 12. In this case, a POC corresponding to the current coding block is 4, a POC corresponding to a forward prediction reference picture block is 4, and a POC corresponding to a backward prediction reference picture block is 12.

[00261] Forward prediction and backward prediction are separately performed on the current coding block to obtain an initial forward encoding prediction block and an initial backward encoding prediction block of the current coding block, and it is assumed that the initial forward encoding prediction block and the initial backward encoding prediction block are respectively FPB1 and BPB1. A first encoding prediction block of the current coding block is obtained by performing weighted summation on FPB1 and BPB1, and is assumed to be DPB1.

[00262] (–6, 12) and (8, 4) are used as reference inputs of the forward and backward motion vector predictors, and motion search with first precision is separately performed on the forward prediction reference picture block and the backward prediction reference picture block. The first encoding prediction block DPB1 is used as a reference. Corresponding new forward and backward encoding prediction blocks obtained in each motion search are compared with the first encoding prediction block DPB1, to obtain a new encoding prediction block with a minimum difference from DPB1, and forward and backward motion vector predictors corresponding to the new encoding prediction block are used as target motion vector predictors, and are assumed to be (–11, 4) and (6, 6) respectively.

[00263] The target motion vector predictors are updated to (–11, 4) and (6, 6), forward prediction and backward prediction are separately performed on the first coding block based on the target motion vector predictors, a target encoding prediction block is obtained by performing weighted summation on obtained new forward and backward encoding prediction blocks, and is assumed to be DPB2, and the encoding prediction block of the current coding block is updated to DPB2.

[00264] Then, (–11, 4) and (6, 6) are used as reference inputs of the forward and backward motion vector predictors, and the motion search with the first precision is separately performed on the forward prediction reference picture block and the backward prediction reference picture block. The encoding prediction block DPB2 of the current coding block is used as a reference. Corresponding new forward and backward encoding prediction blocks obtained in each motion

[00281] It should be noted that, for the motion vector update process on the reference block, refer to the embodiment of this application related to FIG. 13. It should be understood that the reference block in the embodiment related to FIG. 19 is the to-be-processed picture block in the embodiment related to FIG. 13.

[00282] In step 1902, the motion vector of the candidate reference block may be determined based on a preset range of the candidate reference block. For example, when the candidate reference block falls within the preset range, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block, or when the candidate reference block falls beyond the preset range, the final motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Therefore, the one or more motion vectors of the one or more candidate reference blocks can be determined.

[00283] Specifically, the preset range may be a CTB row range. For example, when a CTB in which the candidate reference block is located and a CTB in which the first to-be-processed picture block is located are located in a same row, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Correspondingly, when the CTB in which the candidate reference block is located and the CTB in which the first to-be-processed picture block is located are not located in a same row, the final motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Referring to FIG. 21, it may be understood that, when the coding tree block in which the candidate reference block is located and the coding tree block in which the first to-be-processed picture block is located are located in different rows, and the coding tree block in which the candidate reference block is located and the coding tree block in which the first to-be-processed picture block is located are located in neighboring space of the different rows, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. The neighboring space of the different rows may be top space, top left space, or the like of the different rows. A specific range of the neighboring space of the different rows is not specifically limited in this embodiment of the present invention.

[00284] In a related technology, an encoder may simultaneously encode a plurality of to-be-processed picture blocks. However, when the plurality of to-be-processed picture blocks are located in a same CTB row, the plurality of to-be-processed picture blocks may be candidate reference blocks of each other. Once the plurality of to-be-processed picture blocks are candidate

vector obtained after the initial motion vector is refined, using the final motion vector of the candidate reference block as the motion vector of the candidate reference block can ensure encoding quality.

[00291] 1903: The encoder establishes a candidate predicted motion vector list based on the one or more motion vectors of the one or more candidate reference blocks.

[00292] For a manner of establishing the candidate predicted motion vector list, refer to FIG. 6 to FIG. 9. Details are not described herein again.

[00293] 1904: The encoder determines a reference block of the to-be-processed picture block, where the reference block and the first to-be-processed picture block are located in a same frame of picture.

[00294] A candidate reference block with a minimum rate-distortion cost is selected from the one or more candidate reference blocks of the first to-be-processed picture block as the reference block of the first to-be-processed picture block.

[00295] 1905: The encoder stores a motion vector of the reference block, where the motion vector of the reference block is an initial motion vector of the first to-be-processed picture block.

[00296] The motion vector of the reference block may be obtained from a corresponding location in the established candidate predicted motion vector list.

[00297] Step 1901 to step 1905 are a process in which the encoder obtains an initial motion vector of the first to-be-processed picture block.

[00298] 1906: The encoder determines a prediction block of the first to-be-processed picture block based on the initial motion vector and one or more preset motion vector offsets of the first to-be-processed picture block.

[00299] Step 1906 may be completed by using 13021 to 13024 in FIG. 13. Details are not described herein again.

[00300] The one or more preset motion vector offsets may be set based on a pixel unit. For example, a picture block indicated by the initial motion vector of the first to-be-processed picture block is used as a temporary prediction block of the first to-be-processed picture block, and one or more candidate prediction blocks within one pixel unit or 1/2 pixel unit of the temporary prediction block are searched for by using the temporary prediction block as a center. Certainly, another length or another pixel unit may be used. The one or more preset motion vector offsets are not specifically limited in this embodiment of this application.

[00301] 1907: After determining a motion vector of the prediction block, the encoder uses the motion vector of the prediction block as a final motion vector of the first to-be-processed picture block.

[00302] In step 1907, the motion vector of the prediction block may be determined based on a preset range of the prediction block. For example, when the prediction block falls within the preset range, an initial motion vector of the prediction block is used as the motion vector of the prediction block, or when the prediction block falls beyond the preset range, a final motion vector of the prediction block is used as the motion vector of the prediction block.

[00303] Specifically, the preset range may be a CTB row range. For example, when a CTB in which the prediction block is located and a CTB in which the first to-be-processed picture block is located are located in a same row, the initial motion vector of the prediction block is used as the motion vector of the prediction block. Correspondingly, when the CTB in which the prediction block is located and the CTB in which the first to-be-processed picture block is located are not located in a same row, the initial motion vector of the prediction block is used as the motion vector of the prediction block. Referring to FIG. 21, it may be understood that, when the coding tree block in which the prediction block is located and the coding tree block in which the first to-be-processed picture block is located are located in different rows, and the coding tree block in which the prediction block is located and the coding tree block in which the first to-be-processed picture block is located are located in neighboring space of the different rows, the initial motion vector of the prediction block is used as the motion vector of the prediction block. The neighboring space of the different rows may be top space, top left space, or the like of the different rows. A specific range of the neighboring space of the different rows is not specifically limited in this embodiment of the present invention.

[00304] When the CTB in which the prediction block is located and the CTB in which the first to-be-processed picture block is located are located in a same row, an update process on the prediction block may not be completed. Therefore, the initial motion vector of the prediction block may be directly used as the motion vector of the prediction block, and there is no need to wait until the prediction block is updated, reducing an encoding delay. When the CTB in which the prediction block is located and the CTB in which the first to-be-processed picture block is located are not located in a same row, it indicates that there is enough time to complete an update process on the prediction block. Therefore, the prediction block has the final motion vector, and the final motion

vector of the prediction block may be used as the motion vector of the prediction block. In addition, because the final motion vector is a motion vector obtained after the initial motion vector is refined, using the final motion vector as the motion vector of the prediction block can ensure encoding quality.

[00305] In some embodiments, the preset range may alternatively be a CTB block range. In some other embodiments, the preset range may alternatively be a same CTB block. Determining of the motion vector of the prediction block is similar to determining of the motion vector of the candidate reference block. For this, refer to step 1902. Details are not described herein again.

[00306] 1908: The encoder updates the initial motion vector of the first to-be-processed picture block in the candidate list to the final motion vector of the first to-be-processed picture block.

[00307] 1909: The encoder stores a motion vector residual of the first to-be-processed picture block in target storage space.

[00308] The motion vector residual is a motion vector difference between the final motion vector of the first to-be-processed picture block and the initial motion vector of the first to-be-processed picture block.

[00309] The target storage space is space for storing the motion vector residual by the encoder. For example, the target storage space may be the reference picture storage 64 in FIG. 2.

[00310] It should be noted that, in a merge mode in a related technology, the target storage space stores a zero vector, or the target storage space stores no data. However, in the merge mode in this embodiment of this application, the motion vector residual of the to-be-processed picture block is stored in the target storage space, so that not only the target storage space can be fully used, but also a motion vector of another picture block can be flexibly selected when the motion vector of the another picture block is required for the first to-be-processed picture block. As shown in steps 1902 and 1907, encoding efficiency of the encoder is further improved.

[00311] In addition, in some implementations, in step 1909, the final motion vector of the first to-be-processed picture block may alternatively be used as the motion vector residual of the first to-be-processed picture block and stored in the target storage space, that is, the final motion vector of the first to-be-processed picture block is stored in the target storage space, and the motion vector residual of the first to-be-processed picture block is not stored in the target storage space. In this case, when the final motion vector of the first to-be-processed picture block is required for another block, the final motion vector of the first to-be-processed picture block can be directly obtained

be-processed picture block in the embodiment related to FIG. 14.

[00330] In step 2002, the motion vector of the candidate reference block may be determined based on a preset range of the candidate reference block. For example, when the candidate reference block falls within the preset range, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block, or when the candidate reference block falls beyond the preset range, the final motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Therefore, the one or more motion vectors of the one or more candidate reference blocks can be determined.

[00331] Specifically, the preset range may be a CTB row range. For example, when a CTB in which the candidate reference block is located and a CTB in which the second to-be-processed picture block is located are located in a same row, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Correspondingly, when the CTB in which the candidate reference block is located and the CTB in which the second to-be-processed picture block is located are not located in a same row, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Referring to FIG. 21, it may be understood that, when the coding tree block in which the candidate reference block is located and the coding tree block in which the second to-be-processed picture block is located are located in different rows, and the coding tree block in which the candidate reference block is located and the coding tree block in which the second to-be-processed picture block is located are located in neighboring space of the different rows, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. The neighboring space of the different rows may be top space, top left space, or the like of the different rows. A specific range of the neighboring space of the different rows is not specifically limited in this embodiment of the present invention.

[00332] In a related technology, a decoder may simultaneously decode a plurality of to-be-processed picture blocks. However, when the plurality of to-be-processed picture blocks are located in a same CTB row, the plurality of to-be-processed picture blocks may be candidate reference blocks of each other. Once the plurality of to-be-processed picture blocks are candidate reference blocks of each other, an updated final motion vector of another block can be used as a spatial candidate predicted motion vector of a current to-be-processed picture block only after the another block is updated. Consequently, a decoding delay exists. In addition, because the plurality

of to-be-processed picture blocks are candidate reference blocks of each other, there is no need to wait until the other blocks are updated. Consequently, a decoding delay is increased.

[00333] Therefore, the initial motion vector of the candidate reference block may be directly used as the motion vector of the candidate reference block, and then the motion vector of the candidate reference block is used as the spatial candidate predicted motion vector of the second to-be-processed picture block. For this, refer to step 2003.

[00334] When the CTB in which the candidate reference block is located and the CTB in which the second to-be-processed picture block is located are located in a same row, the update process on the candidate reference block may not be completed. Therefore, the initial motion vector of the candidate reference block may be directly used as the motion vector of the candidate reference block, and there is no need to wait until the candidate reference block is updated, reducing a decoding delay. When the CTB in which the candidate reference block is located and the CTB in which the second to-be-processed picture block is located are not located in a same row, it indicates that there is enough time to complete the update process on the candidate reference block. Therefore, the candidate reference block has the final motion vector, and the final motion vector of the candidate reference block may be used as the motion vector of the candidate reference block. In addition, because the final motion vector is a motion vector obtained after the initial motion vector is refined, using the final motion vector as the motion vector of the candidate reference block can ensure decoding quality.

[00335] The preset range may alternatively be a CTB block range. For example, when the candidate reference block and the second to-be-processed picture block are located in a same CTB, the initial motion vector of the candidate reference block is used as the motion vector of the candidate reference block. Correspondingly, when the candidate reference block and the second to-be-processed picture block are not located in a same CTB, the final motion vector of the candidate reference block is used as the motion vector of the candidate reference block.

[00336] When the candidate reference block and the second to-be-processed picture block are located in a same CTB, it indicates that the update process on the candidate reference block may not be completed, and the initial motion vector of the candidate reference block is directly used as the motion vector of the candidate reference block without a need to wait until the candidate reference block is updated, reducing a decoding delay. In addition, when the candidate reference block and the second to-be-processed picture block are not located in a same CTB, it indicates that

ensure decoding quality.

[00339] 2003: The decoder establishes a candidate predicted motion vector list based on the one or more motion vectors of the one or more candidate reference blocks.

[00340] Step 2003 is the same as step 1903, that is, a manner in which the decoder establishes the candidate predicted motion vector list is consistent with the manner in which the encoder establishes the candidate predicted motion vector list, and the established candidate predicted motion vector lists are the same. The encoder is an encoder that encodes the bitstream received by the decoder.

[00341] 2004: The decoder determines a reference block of the second to-be-processed picture block and a motion vector of the reference block from the candidate predicted motion vector list based on the second identification information.

[00342] A motion vector indicated by an index in the second identification information is selected from the candidate predicted motion vector list based on the index. The motion vector is an initial motion vector of the second to-be-processed picture block, and the motion vector is also the motion vector of the reference block of the second to-be-processed picture block.

[00343] 2005: The decoder stores the motion vector of the reference block as an initial motion vector of the second to-be-processed picture block.

[00344] Steps 2001 to 2005 are a process in which the decoder obtains the initial motion vector of the second to-be-processed picture block.

[00345] 2006: The decoder determines a prediction block of the second to-be-processed picture block based on the initial motion vector and one or more preset motion vector offsets of the second to-be-processed picture block.

[00346] Step 2006 may be completed by using 14021 to 14024 in FIG. 14. Details are not described herein again.

[00347] For the one or more preset motion vector offsets, refer to step 1906. Details are not described herein again.

[00348] It should be noted that there is no specific sequence of performing steps 2005 and 2006, and step 2006 may alternatively be performed before step 2005. A sequence of performing steps 2005 and 2006 is not specifically limited in this embodiment of this application.

[00349] 2007: After determining a motion vector of the prediction block, the decoder uses the motion vector of the prediction block as a final motion vector of the second to-be-processed picture

block.

[00350] In step 2007, the motion vector of the candidate reference block may be determined based on a preset range of the prediction block. For example, when the prediction block falls within the preset range, an initial motion vector of the prediction block is used as the motion vector of the prediction block, or when the prediction block falls beyond the preset range, a final motion vector of the prediction block is used as the motion vector of the prediction block.

[00351] Specifically, the preset range may be a CTB row range. For example, when a CTB in which the prediction block is located and a CTB in which the second to-be-processed picture block is located are located in a same row, the initial motion vector of the prediction block is used as the motion vector of the prediction block. Correspondingly, when the CTB in which the prediction block is located and the CTB in which the second to-be-processed picture block is located are not located in a same row, the initial motion vector of the prediction block is used as the motion vector of the prediction block. Referring to FIG. 21, it may be understood that, when the coding tree block in which the prediction block is located and the coding tree block in which the second to-be-processed picture block is located are located in different rows, and the coding tree block in which the prediction block is located and the coding tree block in which the second to-be-processed picture block is located are located in neighboring space of the different rows, the initial motion vector of the prediction block is used as the motion vector of the prediction block. The neighboring space of the different rows may be top space, top left space, or the like of the different rows. A specific range of the neighboring space of the different rows is not specifically limited in this embodiment of the present invention.

[00352] When the CTB in which the prediction block is located and the CTB in which the second to-be-processed picture block is located are located in a same row, an update process on the prediction block may not be completed. Therefore, the initial motion vector of the prediction block may be directly used as the motion vector of the prediction block, and there is no need to wait until the prediction block is updated, reducing a decoding delay. When the CTB in which the prediction block is located and the CTB in which the second to-be-processed picture block is located are not located in a same row, it indicates that there is enough time to complete an update process on the prediction block. Therefore, the prediction block has the final motion vector, and the final motion vector of the prediction block may be used as the motion vector of the prediction block. In addition, because the final motion vector is a motion vector obtained after the initial

[00360] It should be noted that there is no strict requirement on a sequence between steps 2007 and 2008, and the decoder may alternatively perform step 2007 before step 2008. This is not specifically limited in this embodiment of this application.

[00361] 2009: The decoder obtains, based on the motion vector residual and an initial motion difference of the second to-be-processed picture block, picture information corresponding to the second to-be-processed picture block.

[00362] The picture information includes pixel information that is used to identify original graphics of the second to-be-processed picture block.

[00363] The decoder obtains, based on the motion vector residual and the initial motion difference of the second to-be-processed picture block, the picture information corresponding to the second to-be-processed picture block. For this, refer to FIG. 3. Details are not described herein again.

[00364] Correspondingly, the decoder may alternatively obtain, based on the final motion vector of the second to-be-processed picture block, the picture information corresponding to the second to-be-processed picture block.

[00365] In the foregoing implementation, when decoding the current decoding block, the decoder uses the unupdated initial motion vector of the reference block as the predicted motion vector of the current decoding block instead of the final motion vector of the reference block. In this way, when the motion vector of the reference block is required for the current decoding block, a related step can be performed without a need to wait until the final motion vector of the reference block is updated, ensuring that a decoding delay is reduced while decoding efficiency is improved because the motion vector is updated.

[00366] To further reflect a motion vector storage manner, a specific embodiment is used herein to describe a storage step. Specific descriptions are as follows:

[00367] In the merge mode, there may be two MVs in each prediction direction, which are respectively first MV information and second MV information. For example, the first MV and the second MV may be respectively an initial motion vector of the reference block and a candidate final motion vector of the reference block. An MVD derivation process is introduced into the merge mode. For example, in a forward prediction process, when the first MV is MV0 and the second MV is MV0', a derivation relationship exists: MV0'=MV0+(–)MVD0. By storing two MVs, for example, an index value of an initial motion vector and the foregoing MVD, that is, a difference

between the two MVs, the first MV can be used under a specific condition and the second MV can be used under a specific condition, avoiding a delay.

[00368] For the specific condition, refer to step 1902 or 2002. Details are not described herein again.

[00369] In a feasible implementation, in a non-merge mode, target storage space is used to store a motion vector residual of the reference block, and the motion vector residual of the reference block is a difference between the final motion vector of the reference block and a predicted motion vector of the reference block. In the merge mode, the target storage space is used to store the selected final motion vector, or the target storage space is used to store a difference between the initial motion vector and the selected final motion vector of the reference block.

[00370] Another embodiment of this application is as follows:

[00371] As shown in FIG. 22, a current coding block is a first coding block, predicted motion information of the current coding block is obtained, a reference block is in a merge mode, and a location of motion information of the reference block is not in a same CTB row range as the current coding block. In this case, a forward motion vector predictor of the current coding block (–21, 18) is obtained by adding a forward motion vector (–22, 18) of the reference block and a forward MVD (1, 0), a backward motion vector predictor (1, 12) of the current coding block is obtained by adding a backward motion vector (2, 12) of the reference block and a backward MVD (–1, 0), a POC of a picture in which the current coding block is located is 4, a POC of a forward reference picture that is indicated by an index value of the reference picture is 2, and a POC of a backward reference picture that is indicated by an index value of the reference picture is 6. In this case, a POC corresponding to the current coding block is 4, a POC corresponding to a forward prediction reference picture block is 2, and a POC corresponding to a backward prediction reference picture block is 6.

[00372] (–21, 18) is used as a reference input of the forward motion vector predictor, and motion search with first precision is performed on the forward prediction reference picture block. In this case, the first precision is 1 pixel precision in a 1-pixel range. A first forward prediction block is used as a reference, and a corresponding new forward prediction block is obtained through each motion search. (1, 12) is used as a reference input of the backward motion vector predictor, and motion search with first precision is performed on the backward prediction reference picture block. In this case, the first precision is 1 pixel precision in a 1-pixel range. A first backward prediction

block is used as a reference, and a corresponding new backward prediction block is obtained through each motion search. The new forward prediction block is compared with the new backward prediction block, to obtain a forward prediction block with a minimum difference and a backward prediction block with a minimum difference, and a forward motion vector predictor corresponding to the forward prediction block and a backward motion vector predictor corresponding to the backward prediction block are used as target motion vector predictors, and are assumed to be (–21, 19) and (1, 11) respectively.

[00373] A forward MVD of the current coding block (0, 1) is obtained by subtracting the forward motion vector predictor (–21, 18) from the forward target motion vector predictor (–21, 19), and a backward MVD of the current coding block (0, –1) is obtained by subtracting the backward motion vector predictor (1, 12) from the backward target motion vector predictor (1, 11). MVD calculation method 1 is as follows: MVD0=(–21, 19)–(–21, 18), and MVD1=(1, 11)–(1, 12); or MVD0=(–21, 19)–(–21, 18), and MVD1=–MVD0.

[00374] In addition, forward prediction and backward prediction are separately performed on the first coding block based on the target motion vector predictors, obtained forward and backward prediction blocks are used as target decoding prediction blocks, and the prediction blocks of the current coding block are updated.

[00375] It should be noted that, when the motion search with the first precision is performed on the forward prediction reference picture block and the backward prediction reference picture block, the first precision may be any specified precision, for example, may be integer pixel precision, 1/2 pixel precision, 1/4 pixel precision, or 1/8 pixel precision.

[00376] Another embodiment of this application is as follows:

[00377] As shown in FIG. 22, a current coding block is a first coding block, predicted motion information of the current coding block is obtained, a reference block is in a merge mode, and a location of motion information of the reference block is not in a same CTB range as the current coding block. In this case, a forward motion vector predictor of the current coding block (–21, 18) is obtained by adding a forward motion vector (–22, 18) of the reference block and a forward MVD (1, 0), a backward motion vector predictor (1, 12) of the current coding block is obtained by adding a backward motion vector (2, 12) of the reference block and a backward MVD (–1, 0), a POC of a picture in which the current coding block is located is 4, a POC of a forward reference picture that is indicated by an index value of the reference picture is 2, and a POC of a backward reference

pixel precision, 1/4 pixel precision, or 1/8 pixel precision.

[00382] Another embodiment of this application is as follows:

[00383] As shown in FIG. 22, a current coding block is a first coding block, predicted motion information of the current coding block is obtained, a reference block is in a merge mode, and a location of motion information of the reference block is not in a same CTB line range as the current coding block. In this case, a forward motion vector predictor of the current coding block (–21, 18) is obtained by adding a forward motion vector (–22, 18) of the reference block and a forward MVD (1, 0), a backward motion vector predictor (1, 12) of the current coding block is obtained by adding a backward motion vector (2, 12) of the reference block and a backward MVD (–1, 0), a POC of a picture in which the current coding block is located is 4, a POC of a forward reference picture that is indicated by an index value of the reference picture is 2, and a POC of a backward reference picture that is indicated by an index value of the reference picture is 6. In this case, a POC corresponding to the current coding block is 4, a POC corresponding to a forward prediction reference picture block is 2, and a POC corresponding to a backward prediction reference picture block is 6.

[00384] (–21, 18) is used as a reference input of the forward motion vector predictor, and motion search with first precision is performed on the forward prediction reference picture block. In this case, the first precision is 1 pixel precision in a 1-pixel range. A first forward prediction block is used as a reference, and a corresponding new forward prediction block is obtained through each motion search. (1, 12) is used as a reference input of the backward motion vector predictor, and motion search with first precision is performed on the backward prediction reference picture block. In this case, the first precision is 1 pixel precision in a 1-pixel range. A first backward prediction block is used as a reference, and a corresponding new backward prediction block is obtained through each motion search. The new forward prediction block is compared with the new backward prediction block, to obtain a forward prediction block with a minimum difference and a backward prediction block with a minimum difference, and a forward motion vector predictor corresponding to the forward prediction block and a backward motion vector predictor corresponding to the backward prediction block are used as target motion vector predictors, and are assumed to be (–21, 19) and (1, 11) respectively.

[00385] Then, (–21, 19) is used as a reference input of the forward motion vector predictor, and motion search with first precision is performed on the forward prediction reference picture block.

In this case, the first precision is 1/2 pixel precision. A first forward prediction block is used as a reference, and a corresponding new forward prediction block is obtained through each motion search. (1, 11) is used as a reference input of the backward motion vector predictor, and motion search with first precision is performed on the backward prediction reference picture block. In this case, the first precision is 1/2 pixel precision. A first backward prediction block is used as a reference, and a corresponding new backward prediction block is obtained through each motion search. The new forward prediction block is compared with the new backward prediction block, to obtain a forward prediction block with a minimum difference and a backward prediction block with a minimum difference, and a forward motion vector predictor corresponding to the forward prediction block and a backward motion vector predictor corresponding to the backward prediction block are used as target motion vector predictors.

[00386] A forward MVD of the current coding block (0, 1) is obtained by subtracting the forward motion vector predictor (–21, 18) from the forward target motion vector predictor (–21, 19), and a backward MVD of the current coding block (0, –1) is obtained by subtracting the backward motion vector predictor (1, 12) from the backward target motion vector predictor (1, 11). MVD calculation method 1 is as follows: MVD0=(–21, 19)–(–21, 18), and MVD1=(1, 11)–(1, 12); or MVD0=(–21, 19)–(–21, 18), and MVD1=–MVD0.

[00387] In addition, forward prediction and backward prediction are separately performed on the first coding block based on the target motion vector predictors, obtained forward and backward prediction blocks are used as target decoding prediction blocks, and the prediction blocks of the current coding block are updated.

[00388] It should be noted that, when the motion search with the first precision is performed on the forward prediction reference picture block and the backward prediction reference picture block, the first precision may be any specified precision, for example, may be integer pixel precision, 1/2 pixel precision, 1/4 pixel precision, or 1/8 pixel precision.

[00389] Another embodiment of this application is as follows:

[00390] A current coding block is a first coding block, predicted motion information of the current coding block is obtained. In this case, a forward motion vector predictor of the current coding block (–21, 18) is a forward motion vector (–21, 18) of a reference block, a backward motion vector predictor (1, 12) of the current coding block is a backward motion vector (1, 12) of the reference block, a POC of a picture in which the current coding block is located is 4, a POC of

CLAIMS

What is claimed is:

1. A motion vector residual determining method, comprising:

parsing a bitstream to obtain identification information used to determine an initial motion vector of a picture block to be processed;

separately adding the initial motion vector and a plurality of preset offset vectors to obtain a plurality of candidate final motion vectors;

determining a candidate final motion vector corresponding to a minimum distortion cost in the plurality of candidate final motion vectors as a final motion vector; and

using a difference between the final motion vector and the initial motion vector as a motion vector residual of the picture block, or using the final motion vector as a motion vector residual of the picture block.

2. The method according to claim 1, wherein the method is used for bidirectional inter prediction, the final motion vector comprises a first final motion vector and a second final motion vector, and the initial motion vector comprises a first initial motion vector and a second initial motion vector; the first final motion vector and the first initial motion vector indicate a motion compensation block based on a first reference frame list of the picture block, and the second final motion vector and the second initial motion vector indicate a motion compensation block based on a second reference frame list of the picture block; and

wherein the using a difference between the final motion vector and the initial motion vector as a motion vector residual of the picture block comprises:

using a difference between the first final motion vector and the first initial motion vector as a first motion vector residual of the picture block.

3. The method according to claim 2, further comprising:

using a difference between the second final motion vector and the second initial motion vector as a second motion vector residual of the picture block.

4. The method according to claim 2, further comprising:

using the first final motion vector as the first motion vector residual of the picture block.

5. The method according to claim 4, further comprising:

using the second final motion vector as a second motion vector residual of the picture block.

6. A motion vector residual determining apparatus, comprising:

a parsing module configured to parse a bitstream to obtain identification information used to determine an initial motion vector of a picture block to be processed;

a summation module configured to separately add the initial motion vector and a plurality of preset offset vectors to obtain a plurality of candidate final motion vectors;

a determining module configured to determine a candidate final motion vector corresponding to a minimum distortion cost in the plurality of candidate final motion vectors as the final motion vector; and

an obtaining module configured to: use a difference between the final motion vector and the initial motion vector as a motion vector residual of the picture block, or use the final motion vector as a motion vector residual of the picture block.

7. The apparatus according to claim 6, wherein the apparatus is used for bidirectional inter prediction, the final motion vector comprises a first final motion vector and a second final motion vector, and the initial motion vector comprises a first initial motion vector and a second initial motion vector; the first final motion vector and the first initial motion vector indicate a motion compensation block based on a first reference frame list of the picture block, and the second final motion vector and the second initial motion vector indicate a motion compensation block based on a second reference frame list of the picture block; and wherein

the obtaining module is further configured to:

use a difference between the first final motion vector and the first initial motion vector as a first motion vector residual of the picture block.

8. The apparatus according to claim 7, wherein the obtaining module is further configured to: use a difference between the second final motion vector and the second initial motion vector as a second motion vector residual of the picture block.

9. The apparatus according to claim 7, wherein the obtaining module is further configured to: use the first final motion vector as the first motion vector residual of the picture block.

10. The apparatus according to claim 9, wherein the obtaining module is further configured to: use the second final motion vector as a second motion vector residual of the picture block.

11. A motion vector residual determining apparatus, comprising:

a memory for storing computer executable instructions; and

at least one processor operatively coupled to the memory to execute the computer-executable instructions to:

parse a bitstream to obtain identification information used to determine an initial motion vector of a picture block to be processed;

separately add the initial motion vector and a plurality of preset offset vectors to obtain a plurality of candidate final motion vectors;

determine a candidate final motion vector corresponding to a minimum distortion cost in the plurality of candidate final motion vectors as the final motion vector; and

use a difference between the final motion vector and the initial motion vector as a motion vector residual of the picture block, or use the final motion vector as a motion vector residual of the picture block.

12. The apparatus according to claim 11, wherein the apparatus is used for bidirectional inter prediction, the final motion vector comprises a first final motion vector and a second final motion vector, and the initial motion vector comprises a first initial motion vector and a second initial motion vector; the first final motion vector and the first initial motion vector indicate a motion compensation block based on a first reference frame list of the picture block, and the second final motion vector and the second initial motion vector indicate a motion compensation block based on a second reference frame list of the picture block; and wherein

the at least one processor being further configured to execute the computer-executable instructions to:

use a difference between the first final motion vector and the first initial motion vector as a

first motion vector residual of the picture block.

13. The apparatus according to claim 12, wherein the at least one processor is further configured to execute the computer-executable instructions to: use a difference between the second final motion vector and the second initial motion vector as a second motion vector residual of the picture block.

14. The apparatus according to claim 12, wherein the at least one processor is further configured to execute the computer-executable instructions to: use the first final motion vector as the first motion vector residual of the picture block.

15. The apparatus according to claim 14, wherein the at least one processor is further configured to execute the computer-executable instructions to: use the second final motion vector as a second motion vector residual of the picture block.

16. A non-transitory computer-readable storage medium storing computer instructions, that when executed by one or more processors, cause the one or more processors to:

parse a bitstream to obtain identification information used to determine an initial motion vector of a picture block to be processed;

separately add the initial motion vector and a plurality of preset offset vectors to obtain a plurality of candidate final motion vectors;

determine a candidate final motion vector corresponding to a minimum distortion cost in the plurality of candidate final motion vectors as the final motion vector; and

use a difference between the final motion vector and the initial motion vector as a motion vector residual of the picture block, or use the final motion vector as a motion vector residual of the picture block.

17. The non-transitory computer-readable storage medium according to claim 16, wherein the apparatus is used for bidirectional inter prediction, the final motion vector comprises a first final motion vector and a second final motion vector, and the initial motion vector comprises a first initial motion vector and a second initial motion vector; the first final motion vector and the

first initial motion vector indicate a motion compensation block based on a first reference frame list of the picture block, and the second final motion vector and the second initial motion vector indicate a motion compensation block based on a second reference frame list of the picture block; and wherein

the computer instructions further cause the one or more processors to:

use a difference between the first final motion vector and the first initial motion vector as a first motion vector residual of the picture block.

18. The non-transitory computer-readable storage medium according to claim 17, wherein the computer instructions further cause the one or more processors to: use a difference between the second final motion vector and the second initial motion vector as a second motion vector residual of the picture block.

19. The non-transitory computer-readable storage medium according to claim 17, wherein the computer instructions further cause the one or more processors to: use the first final motion vector as the first motion vector residual of the picture block.

20. The non-transitory computer-readable storage medium according to claim 19, wherein the computer instructions further cause the one or more processors to: use the second final motion vector as a second motion vector residual of the picture block.