

motion model (for example, an affine motion model) is added based on block-based motion compensation (motion compensation, MC) prediction using a translational motion model. Regardless of whether the non-translational motion model is used, a related parameter about the affine motion model needs to be added to a coded video sequence. As a result, a length of the coded video sequence is increased.

SUMMARY

[0006] This application provides a video picture prediction method and apparatus, to resolve a problem in a conventional technology that a length of a coded video sequence is increased.

[0007] According to a first aspect, an embodiment of this application provides a video picture prediction method, including: parsing a bitstream to obtain a first identifier; when the first identifier indicates that a candidate motion model for inter prediction of a picture block in a video picture including a to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine motion model, parsing the bitstream to obtain a second identifier; and determining, based on the second identifier, a prediction mode for inter prediction of the to-be-processed block, where the prediction mode includes an affine motion model-based merge mode, an affine motion model-based AMVP mode, and a non-affine motion model-based prediction mode.

[0008] In the foregoing solution, for example, some video pictures may have some affine features, while some video pictures may have no affine features. In this case, an identifier may be added to a bitstream to indicate whether an affine motion model-based inter prediction mode is enabled for the video picture. If the affine motion model-based inter prediction mode is not enabled for the video picture, a parameter, related to the affine motion model, of a picture block of the video picture may not need to be transmitted. On a decoder side, during decoding of the picture block of the video picture, the parameter related to the affine motion model does not need to be parsed. This can reduce load of a decoder, increase a processing speed, and decrease a processing time.

[0009] For example, terms such as "first" and "second" are merely used for distinction and description, and shall not be understood as an indication or implication of relative importance or an indication or implication of an order. In addition, for a parameter such as an identifier, different

picture prediction method, including: parsing a bitstream to obtain a first identifier; when the first identifier indicates that a candidate motion model for inter prediction of a picture block of a slice including a to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine motion model, parsing the bitstream to obtain a second identifier; and determining, based on the second identifier, a prediction mode for inter prediction of the to-be-processed block, where the prediction mode includes an affine motion model-based merge mode, an affine motion model-based AMVP mode, and a non-affine motion model-based prediction mode.

[0023] In the foregoing solution, for example, some slices of a video picture may have some affine features, while some slices of the video picture may have no affine features. In this case, an identifier may be added to the bitstream to indicate whether an affine motion model-based inter prediction mode is enabled for a slice of the video picture. If the affine motion model-based inter prediction mode is not enabled for the slice, a parameter, related to the affine motion model, of a picture block included in the slice may not need to be transmitted. On a decoder side, during decoding of the picture block in the slice, the parameter related to the affine motion model does not need to be parsed. This can reduce load of a decoder, increase a processing speed, and decrease a processing time.

[0024] It should be noted that the first identifier in the second aspect is referred to as an identifier 2 in a specific embodiment, and the second identifier in the second aspect is referred to as an identifier 22 in a specific embodiment.

[0025] In a possible design of the second aspect, the parsing a bitstream to obtain a first identifier includes: parsing a slice header of the slice in the bitstream to obtain the first identifier.

[0026] When the first identifier is configured in the slice header of the slice, and the first identifier indicates that the affine motion model-based inter prediction mode is not enabled for the slice, each picture-block-level syntax of the slice does not include a syntax element related to the affine motion model. On the decoder side, during decoding of the picture block in the slice, the parameter related to the affine motion model does not need to be parsed. This can reduce load of a decoder, increase a processing speed, and decrease a processing time.

[0027] In a possible design of the second aspect, when the first identifier indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block includes the affine motion model, the method further includes: parsing the

bitstream to obtain a third identifier. When the third identifier is a first value, the affine motion model includes only a 4-parameter affine model; or when the third identifier is a second value, the affine motion model includes a 4-parameter affine model and a 6-parameter affine model. The first value is different from the second value.

5 **[0028]** It should be noted that the third identifier in the second aspect is referred to as an identifier 23 in a specific embodiment.

[0029] In a possible design of the second aspect, the method further includes: when the second identifier indicates that affine motion model-based merge mode is used for inter prediction of the to-be-processed block, and the third identifier is the second value, constructing a first candidate motion vector list, where the first candidate motion vector list includes a first element, and the first element includes motion information of three control points for constructing the 6-parameter affine motion model; or

10

 when the second identifier indicates that affine motion model-based merge mode is used for inter prediction of the to-be-processed block, and the third identifier is the first value, constructing a second candidate motion vector list, where the second candidate motion vector list includes a second element, and the second element includes only motion information of two control points for constructing the 4-parameter affine motion model.

15

[0030] In a possible design of the second aspect, when the second identifier indicates that affine motion model-based merge mode is used for inter prediction of the to-be-processed block, and the third identifier is the second value, the first candidate motion vector list further includes the second element.

20

[0031] In a possible design of the second aspect, the method further includes: when the second identifier indicates that the affine motion model-based AMVP mode is used for inter prediction of the to-be-processed block, and the third identifier is the second value, parsing the bitstream to obtain a fourth identifier.

25

[0032] When the fourth identifier is a third value, the affine motion model is the 6-parameter affine motion model, or when the fourth identifier is a fourth value, the affine motion model is the 4-parameter affine motion model. The third value is different from the fourth value.

[0033] It should be noted that, in this application, the fourth identifier is referred to as an identifier 24 in a specific embodiment.

30

[0034] In the foregoing design, the third identifier indicating whether the affine motion model

may be applied to video picture encoding or decoding. FIG. 1 is a schematic block diagram of a video encoding and decoding system 10 according to an embodiment of this application. As shown in FIG. 1, the system 10 includes a source apparatus 11 and a destination apparatus 12. The source apparatus 11 generates encoded video data and sends the encoded video data to the destination apparatus 12. The destination apparatus 12 is configured to receive the encoded video data, decode the encoded video data, and display the decoded video data. The source apparatus 11 and the destination apparatus 12 may include any one of a wide range of apparatuses, including a desktop computer, a laptop computer, a tablet computer, a set-top box, a mobile phone such as a "smart" phone, a "smart" touch panel, a television set, a camera, a display apparatus, a digital media player, a video gaming console, a video streaming transmission apparatus, and the like.

[00116] A solution for predicting an intra mode of a picture block provided in the embodiments of this application may be applied to video picture encoding or decoding.

[00117] The destination apparatus 12 may receive the to-be-decoded encoded video data through a link 16. The link 16 may include any type of media or apparatuses that can transfer the encoded video data from the source apparatus 11 to the destination apparatus 12. In a possible implementation, the link 16 may include a communications medium that enables the source apparatus 11 to directly transmit the encoded video data to the destination apparatus 12 in real time. The encoded video data may be modulated according to a communications standard (for example, a wireless communications protocol), and modulated encoded video data is transmitted to the destination apparatus 12. The communications medium may include any wireless or wired communications medium, for example, a radio frequency spectrum or one or more physical

25

30

the system 10 may be configured to support unidirectional or bidirectional video transmission, so as to support applications such as streaming video transmission, video playing, video broadcasting, and/or video calling.

[00120] In a possible implementation of FIG. 1, the source apparatus 11 may include a video source 13, a video encoder 20, and an output interface 14. In some applications, the output interface 14 may include a modulator/demodulator (a modem) and/or a transmitter. In the source apparatus 11, the video source 13 may include, for example, the following source devices: a video capture apparatus (for example, a camera), an archive containing a previously captured video, a video feed-in interface for receiving a video from a video content provider, and/or a computer graphics system used to generate computer graphics data as a source video, or a combination thereof. In a possible implementation, if the video source 13 is a camera, the source apparatus 11 and the destination apparatus 12 may constitute a camera phone or a video phone. For example, the technologies described in this application may be applied to, for example, video decoding, and may be applied to wireless and/or wired applications.

[00121] The video encoder 20 may encode, capture, pre-capture, or calculate a generated video. The encoded video data may be directly transmitted to the destination apparatus 12 through the output interface 14 of the source apparatus 11. The encoded video data may also (or alternatively) be stored on the storage apparatus for later access by the destination apparatus 12 or another apparatus for decoding and/or playing.

[00122] The destination apparatus 12 includes the input interface 15, a video decoder 30, and a display apparatus 17. In some applications, the input interface 15 may include a receiver and/or a modem. The input interface 15 of the destination apparatus 12 receives the encoded video data through the link 16. The encoded video data transmitted or provided to the storage apparatus through the link 16 may include a variety of syntax elements generated by the video encoder 20 for the video decoder 30 to decode video data. These syntax elements may be included together with the encoded video data that is transmitted on the communications medium and that is stored on the storage medium or the file server.

[00123] The display apparatus 17 may be integrated with the destination apparatus 12, or may be located outside the destination apparatus 12. In some possible implementations, the destination apparatus 12 may include an integrated display apparatus, and may also be configured to connect to an interface of an external display apparatus. In another possible implementation, the destination

apparatus 12 may be a display apparatus. Generally, the display apparatus 17 displays decoded video data to a user, and may include any of a variety of display apparatuses, for example, a liquid crystal display, a plasma display, an organic light-emitting diode display, or another type of display apparatus.

5 **[00124]** 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 (JEM). 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 thereof, where the
10 ITU-T H.265 standard is also referred to as the high efficiency video coding standard, and the ITU-T H.264 standard is also referred to as 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. Other possible implementations of the video compression standard include MPEG-2 and ITU-T H.263.

15 **[00125]** Although not shown in FIG. 1, in some aspects, the video encoder 20 and the video decoder 30 may be integrated with an audio encoder and an audio decoder, respectively, and may include an appropriate multiplexer-demultiplexer (MUX-DEMUX) unit or other hardware and software to encode both audio and a video in a common data stream or separate data streams. If applicable, in some feasible implementations, the MUX-DEMUX unit may comply with the ITU
20 H.223 multiplexer protocol or other protocols such as a user datagram protocol (UDP).

[00126] The video encoder 20 and the video decoder 30 may be separately implemented as any one of a variety of appropriate encoder circuits, for example, one or more microprocessors, a digital signal processor (digital signal processor, DSP), an application-specific integrated circuit (application-specific integrated circuit, ASIC), a field-programmable gate array (field-programmable gate array, FPGA), discrete logic, software, hardware, firmware, or any
25 combination thereof. When some of the techniques are implemented as software, an apparatus may store an instruction for the software into 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 techniques of this application. Each of the video encoder 20 and the video decoder 30 may be
30 included in one or more encoders or decoders, and either the video encoder 20 or the video decoder 30 may be integrated as a part of a combined encoder/decoder (CODEC) in a corresponding

size. Each CU may include one or more PUs and one or more TUs. For example, syntactic data associated with the CU may describe partitioning of a CU into one or more PUs. A different partitioning mode may be used if the CU is encoded in a skip mode, a direct mode, an intra prediction mode, or an inter prediction mode. The PU obtained through partitioning may be in a non-square shape. For example, the syntactic data associated with the CU may also describe partitioning of a CU into one or more TUs based on the quadtree. The TU may be in a square or non-square shape.

[00131] The HEVC standard allows TU-based transform. Different CUs may include different 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 divide a residual sample corresponding to a 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.

[00132] 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 an 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, 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.

[00133] 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 prediction, the video encoder may calculate a residual value corresponding to a PU. The residual value 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 serialized transform coefficients for entropy decoding. In this application, the term "picture block" is usually used to represent a decoding node of a CU. In some specific applications, in this application, the term "picture block" may also be used to represent a tree block including a decoding node, a PU, and a TU, for example,

(which is not shown in FIG. 2A) to filter a boundary of a picture block, so as to eliminate an artifact from a reconstructed video picture. When necessary, the filter usually filters an output from the summator 28.

[00143] Optionally, the video encoder 20 may further include a partitioning unit (which is not shown in FIG. 2A). The video encoder 20 receives video data, and the partitioning unit partitions the video data into picture blocks. Such partitioning may further include partitioning into slices, picture blocks, or other larger units, and (for example) partitioning into picture blocks that is performed based on quadtree structures of an LCU and a CU. The video encoder 20 describes, for example, a component for encoding a picture block in a to-be-encoded video slice. A slice may usually be split into a plurality of picture blocks (and may be split into a set of picture blocks). Slice types include I (which is mainly used for intra-picture encoding), P (which is used for inter forward reference prediction picture encoding), and B (which is used for inter bidirectional reference prediction picture encoding).

[00144] The prediction module 21 is configured to perform intra or inter prediction on a picture block that needs to be currently processed, to obtain a predictor (which may be referred to as predicted information in this application) of the current block. In this embodiment of this application, the picture block that needs to be currently processed may be a to-be-processed block for short, may be a current picture block for short, or may be the current block for short. Alternatively, the picture block that needs to be currently processed in an encoding phase may be referred to as a current coding block(encoding block) for short, and the picture block that needs to be currently processed in a decoding phase may be referred to as a current coding block(decoding block).

[00145] Specifically, the inter prediction module 211 included in the prediction module 21 performs inter prediction on the current block, to obtain an inter predictor. The intra prediction module 212 performs intra prediction on the current block, to obtain an intra predictor. The inter prediction module 211 searches a reconstructed picture for a matched reference block for a current block in a current picture; uses a sample value of a sample in the reference block as predicted information or a predictor of a sample value of a sample in the current block (the following does not distinguish information from a value), where this process is referred to as motion estimation (Motion estimation, ME) (as shown in FIG. 2B); and transmits motion information of the current block.

[00146] It should be noted that the motion information of the picture block includes indication

motion vector list by using rate-distortion costs, and candidate motion information with a minimum rate-distortion cost is used as a motion vector predictor (motion vector predictor, MVP) of the current block. Both locations of the neighboring blocks and a traversal order the neighboring blocks are predefined. The rate-distortion cost is calculated according to formula (1), where J represents a rate-distortion cost (RD Cost), SAD is a sum of absolute differences (sum of absolute differences, SAD) between an original pixel value and a predicted pixel value obtained through motion estimation by using the candidate motion vector predictor, R represents a bit rate, and λ represents a Lagrange multiplier. An encoder side transfers an index value of the selected motion vector predictor in the candidate motion vector list and a reference picture index value to a decoder side. Further, motion search is performed in a neighborhood with the MVP as a center, to obtain an actual motion vector of the current block. The encoder side transfers a difference (motion vector difference) between the MVP and an actual motion vector to the decoder side.

$$J = SAD + \lambda R \quad (1)$$

[00162] In the merge mode, a candidate motion vector list is first constructed by using motion information of spatial or temporal neighboring encoded block of a current block, then optimal motion information is determined from the candidate motion vector list by calculating a rate-distortion cost and is used as motion information of the current block, and then an index value (which is denoted as a merge index, same hereinafter) of a location of the optimal motion information in the candidate motion vector list is transferred to a decoder side. FIG. 3 shows spatial and temporal candidate motion information of the current block. The spatial candidate motion information comes from five spatial neighboring blocks (A0, A1, B0, B1, and B2). If the neighboring blocks are unavailable (there is no neighboring block, the neighboring blocks are not encoded, or a prediction mode used by the neighboring blocks is not the inter prediction mode), motion information of the neighboring blocks is not added to the candidate motion vector list. The temporal candidate motion information of the current block is obtained by scaling an MV of a block in a corresponding location in a reference frame based on picture order counts (picture order count, POC) of a reference frame and a current frame. Whether a block in a location T in the reference frame is available is first determined. If the block is unavailable, a block in a location C is selected.

[00163] Similar to the AMVP mode, in the merge mode, both locations of the neighboring blocks and a traversal order of the neighboring blocks are also predefined. In addition, the locations

using a 4-tuple including motion information (for example, 4 MVs) of the control points CP1, CP2, CP3, and CP4 is denoted as Bilinear (CP1, CP2, CP3, CP4).

[00211] In this embodiment of this application, for ease of description, a combination of motion information of two control points (or two encoded blocks) is referred to as a 2-tuple for short, a combination of motion information of three control points (or three encoded blocks) is referred to as a 3-tuple for short, and a combination of motion information of four control points (or four encoded blocks) is referred to as a 4-tuple for short.

[00212] These models are traversed in a preset order. If motion information of a specific control point corresponding to a combined model is unavailable, it is considered that the model is unavailable. Otherwise, a reference picture index of the model is determined, and a motion vector of the control point is scaled. If motion information of all control points obtained after scaling is consistent, the model is invalid. If all of motion information of control points controlling the model is available, and the model is valid, the motion information of the control points for constructing the model is added to a motion information candidate list.

[00213] A control point motion vector scaling method is shown in formula (12):

$$MV_s = \frac{CurPoc - DesPoc}{CurPoc - SrcPoc} \times MV(12), \text{ where}$$

CurPoc represents a POC number of a current frame, *DesPoc* represents a POC number of a reference frame of a current block, *SrcPoc* represents a POC number of a reference frame of a control point, MV_s represents a motion vector obtained after scaling, and *MV* represents a motion vector of a control point.

[00214] It should be noted that a combination of different control points may be converted into control points in a same location.

[00215] For example, a 4-parameter affine motion model obtained by using a combination {CP1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, or {CP3, CP4} is converted into {CP1, CP2} or {CP1, CP2, CP3}. A conversion method is: substituting motion vectors of the control points and coordinate information of the control points into formula (2), to obtain a model parameter; and then substituting coordinate information of {CP1, CP2} into formula (3), to obtain motion vectors of CP1 and CP2.

[00216] More directly, conversion may be performed according to formulas (13) to (21). *W* represents the width of the current block. *H* represents the height of the current block. In formulas

allowAffineMerge = 0; or when $cbWidth \geq 8$ and $cbHeight \geq 8$, allowAffineMerge = 1.

[00248] The variable allowAffineInter may be used to indicate whether a condition for allowing an affine motion model-based AMVP mode for the current block is satisfied. For example, when allowAffineInter = 0, it indicates that the condition for allowing the affine motion model-based AMVP mode is not satisfied; or when allowAffineInter = 1, it indicates that the condition for allowing the affine motion model-based AMVP mode is satisfied. The condition for allowing the affine motion model-based AMVP mode may be that both the width and the height of the current block are greater than or equal to 16. In other words, when $cbWidth < 16$ or $cbHeight < 16$, allowAffineInter = 0; or when $cbWidth \geq 16$ and $cbHeight \geq 16$, allowAffineInter = 1.

[00249] The syntax element affine_merge_flag[x0][y0] may be used to indicate whether the affine motion model-based merge mode is used for the current block when the slice including the current block is a P slice or a B slice. For example, when affine_merge_flag[x0][y0] = 1, it indicates that the affine motion model-based merge mode is used for the current block; or when affine_merge_flag[x0][y0] = 0, it indicates that the affine motion model-based merge mode is not used for the current block, and a translational motion model-based merge mode may be used.

[00250] The syntax element affine_inter_flag[x0][y0] may be used to indicate whether the affine motion model-based AMVP mode is used for the current block when the slice including the current block is a P slice or a B slice. For example, when affine_inter_flag[x0][y0] = 1, it indicates that the affine motion model-based AMVP mode is used for the current block; or when affine_inter_flag[x0][y0] = 0, it indicates that the affine motion model-based AMVP mode is not used for the current block, and a translational motion model-based AMVP mode may be used.

[00251] The syntax element affine_type_flag[x0][y0] may be used to indicate whether a 6-parameter affine motion model-based motion compensation is used for the current block when the slice including the current block is a P slice or a B slice. When affine_type_flag[x0][y0] = 0, it indicates that the 6-parameter affine motion model-based motion compensation is not used for the current block, and only a 4-parameter affine motion model-based motion compensation may be used for the current block; or when affine_type_flag[x0][y0] = 1, it indicates that the 6-parameter affine motion model-based motion compensation is used for the current block.

[00252] As shown in Table 2, when MotionModelIdc[x0][y0] = 1, it indicates that the 4-parameter affine motion model is used; when MotionModelIdc[x0][y0] = 2, it indicates that the 6-parameter affine motion model is used; or when MotionModelIdc[x0][y0] = 0, it indicates that a

translational motion model is used.

Table 2

MotionModelIdc[x0][y0]	Motion model for motion compensation (motion model for motion compensation)
0	Translational motion (translational motion)
1	4-parameter affine motion (4-parameter affine motion)
2	6-parameter affine motion (6-parameter affine motion)

[00253] The variable MaxNumMergeCand is used to represent a maximum list length, and indicates a maximum number of candidates of a constructed candidate motion vector list. inter_pred_idc[x0][y0] is used to indicate a prediction direction. PRED_L1 is used to indicate backward prediction. num_ref_idx_l0_active_minus1 indicates a quantity of reference frames in a forward reference frame list(list 0), and ref_idx_l0[x0][y0] indicates a forward (list 0) reference picture index for the current block. mvd_coding(x0, y0, 0, 0) indicates a first motion vector difference. mvp_l0_flag[x0][y0] indicates an MVP candidate index of a forward MVP candidate list (list 0). PRED_L0 indicates forward prediction. num_ref_idx_l1_active_minus1 indicates a quantity of reference frames in a backward reference frame list (list 1). ref_idx_l1[x0][y0] indicates a backward (list 1) reference picture index for the current block, and mvp_l1_flag[x0][y0] indicates an MVP candidate index of a backward MVP candidate list (list 1).

[00254] In Table 1, ae(v) represents a syntax element encoded by using context-based adaptive binary arithmetic coding (context-based adaptive binary arithmetic coding, CABAC).

[00255] The following describes in detail an inter prediction process. Refer to FIG. 6A.

[00256] Step 601: Parse a bitstream based on a syntax structure shown in Table 1, to determine an inter prediction mode of a current block.

[00257] If it is determined that the inter prediction mode of the current block is an affine motion model-based AMVP mode, step 602a is performed.

[00258] That is, when merge_flag = 0 and affine_inter_flag = 1, it indicates that the inter prediction mode of the current block is the affine motion model-based AMVP mode.

[00259] If it is determined that the inter prediction mode of the current block is an affine motion model-based merge (merge) mode, step 602b is performed.

[00260] That is, when `merge_flag = 1` and `affine_merge_flag = 1` in the syntax elements, it indicates that the inter prediction mode of the current block is the affine motion model-based merge mode.

[00261] Step 602a: Construct a candidate motion vector list corresponding to the affine motion model-based AMVP mode, and perform step 603a.

[00262] A candidate control point motion vector of the current block is derived by using a method for predicting an inherited control point motion vector and/or a constructed control point motion vector, and is added to the candidate motion vector list.

[00263] The candidate motion vector list may include a 2-tuple list (a current encoding block is a 4-parameter affine motion model) or a 3-tuple list. The 2-tuple list includes one or more 2-tuples used to construct the 4-parameter affine motion model. The 3-tuple list includes one or more 3-tuples used to construct a 6-parameter affine motion model.

[00264] Optionally, the candidate motion vector 2-tuple/3-tuple list is pruned and sorted according to a specific rule, and may be truncated or padded to a specific quantity.

[00265] A1: Descriptions are provided for a procedure of constructing the candidate motion vector list by using the method for predicting the inherited control point motion vector.

[00266] FIG. 4 is used as an example. For example, blocks in neighboring locations around the current block are traversed in an order A1->B1->B0->A0->B2 in FIG. 4, to find affine coding blocks in which the blocks in the neighboring locations are located and obtain control point motion information of the affine coding blocks. Further, a motion model is constructed by using the control point motion information of the affine coding blocks, and candidate control point motion information of the current block is derived. For details, refer to related descriptions of the method for predicting the inherited control point motion vector in (3). Details are not described herein.

[00267] For example, an affine motion model used for the current block is a 4-parameter affine motion model (that is, `MotionModelIdc = 1`). If a neighboring affine coding block is a 4-parameter affine motion model, motion vectors of two control points of the affine coding block are obtained: a motion vector (`vx4`, `vy4`) of the top-left control point (`x4`, `y4`) and a motion vector (`vx5`, `vy5`) of the top-right control point (`x5`, `y5`). The affine coding block is an affine coding block predicted in an encoding phase by using the affine motion model.

[00268] Motion vectors of the top-left control point and the top-right control point of the current block are derived by using the 4-parameter affine motion model including the two control points

descriptions are provided.

[00318] To resolve the foregoing problem, the embodiments of this application provide, but not limited to, the following several possible implementations.

[00319] In a first possible implementation, for example, some video pictures may have some affine features, while some video pictures may have no affine features. In this case, an identifier may be added to a bitstream to indicate whether an affine motion model-based inter prediction mode is enabled for the video picture.

[00320] In a second possible implementation, for example, not all slices in a video picture have affine features, an identifier may be added to a bitstream to indicate whether an affine motion model-based inter prediction mode is enabled for the slice.

[00321] In a third possible implementation, for example, some video pictures have no affine features, and not all slices of some video pictures have affine features. In this case, two types of identifiers may be added to the bitstream. A first type of identifier is used to indicate whether an affine motion model-based inter prediction mode is enabled for a video picture, and a second type of identifier is used to indicate whether the affine motion model-based inter prediction mode is enabled for a slice in the video picture.

[00322] For ease of subsequent description, the identifier used to indicate whether the affine motion model-based inter prediction mode is enabled for the video picture is referred to as an identifier 1, and the identifier used to indicate whether the affine motion model-based inter prediction mode is enabled for the slice included in the video picture is referred to as an identifier 2.

[00323] In the foregoing three possible implementations, for a picture block included in a video picture or a slice for which an affine motion model does not need to be used, a parameter, related to the affine motion model, of the picture block may not need to be transmitted, for example, affine_merge_flag, affine_inter_flag, or affine_type_flag in Table 1. On a decoder side, during decoding of the picture block, the parameter related to the affine motion model does not need to be parsed. This can reduce load of a decoder, increase a processing speed, and decrease a processing time.

[00324] The following describes in detail the foregoing several possible implementations provided in this application from a perspective of a decoder side with reference to the accompanying drawings. Specifically, the possible implementations may be performed by a video

decoder 30, or may be performed by a motion compensation module in a video decoder, or may be performed by a processor.

[00325] Refer to FIG. 7. A first possible implementation is described.

[00326] Step 701: Parse a bitstream to obtain an identifier 1.

5 **[00327]** The identifier 1 is used to indicate whether a candidate motion model for inter prediction of a video picture including a to-be-processed block includes an affine motion model. In other words, the identifier 1 is used to indicate whether the affine motion model-based motion compensation is enabled for the video picture including the to-be-processed block.

[00328] For example, the identifier 1 may be configured in an SPS of the bitstream. Based on
10 this, the parsing a bitstream to obtain an identifier 1 may be implemented in the following manner: parsing the SPS of the bitstream to obtain the identifier 1.

[00329] Step 702: When the identifier 1 indicates that the candidate motion model for inter prediction of the video picture including the to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine
15 motion model, parse the bitstream to obtain an identifier 12.

[00330] Optionally, the preset condition may include at least one of the following cases:

[00331] The width of the to-be-processed block is greater than or equal to a first preset threshold;
and

the height of the to-be-processed block is greater than or equal to a second preset
20 threshold.

[00332] The first preset threshold may be equal to the second preset threshold. For example, when an inter prediction mode is a merge mode, the preset condition may be that the width of the to-be-processed block is greater than or equal to 8, and that the height of the to-be-processed block is greater than or equal to 8. When an inter prediction mode is an AMVP mode, the preset condition
25 may be that the width of the to-be-processed block is greater than or equal to 16, and that the height of the to-be-processed block is greater than or equal to 16.

[00333] For example, when the inter prediction mode is the merge mode, whether the to-be-processed block meets the preset condition for inter prediction using the affine motion model may be represented by a variable allowAffineMerge. For example, when the inter prediction mode is
30 an AMVP mode, whether the to-be-processed block meets the preset condition for inter prediction using the affine motion model may be represented by a variable allowAffineInter. For related

descriptions of allowAffineMerge and allowAffineInter, refer to Table 1. Details are not described herein again.

[00334] For example, when the identifier 1 indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block does not include an affine motion model, parameter information related to the affine motion model does not need to be parsed when inter prediction is performed on the picture block of the video picture. For example, the bitstream does not need to be parsed to obtain the identifier 12.

[00335] For example, when the identifier 1 is a value A, it indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block includes an affine motion model, or when the identifier 1 is a value B, it indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block does not include an affine motion model. For example, when the identifier 1 is 1, it indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block includes an affine motion model, or when the identifier 1 is 0, it indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block does not include an affine motion model.

[00336] Step 703: Determine, based on the identifier 12, a prediction mode for inter prediction of the to-be-processed block, where the prediction mode includes an affine motion model-based merge mode, an affine motion model-based AMVP mode, and a non-affine motion model-based prediction mode.

[00337] For example, the identifier 1 may be represented by a syntax element `sps_affine_enabled_flag`. `sps_affine_enabled_flag` specifies whether affine model based motion compensation is enabled for inter prediction (`sps_affine_enabled_flag` specifies whether affine model based motion compensation can be used for inter prediction). For example, `sps_affine_enabled_flag` is equal to 0, the syntax elements shall be constrained such that no affine model based motion compensation is used in the CVS, and `affine_merge_flag`, `affine_inter_flag` and `affine_type_flag` are not presented in picture block level syntax of the CVS. `sps_affine_enabled_flag` is equal to 1, affine model based motion compensation may be used in the CVS. (If `sps_affine_enabled_flag` is equal to 0, the syntax elements shall be constrained such that no affine model based motion compensation is used in the CVS, and `affine_merge_flag`, `affine_inter_flag` and `affine_type_flag` are not presented in coding unit level syntax of the CVS. Otherwise (`sps_affine_enabled_flag` is equal to 1), affine model based motion compensation may be used in

the CVS.)

[00338] For example, when the inter prediction mode is the AMVP mode, the identifier 12 may be represented by a syntax element `affine_inter_flag`; or when the inter prediction mode is the merge mode, the identifier 12 may be represented by a syntax element `affine_merge_flag`. For related descriptions of the `affine_inter_flag` and the `affine_merge_flag`, refer to the descriptions of Table 1. Details are not described herein again.

[00339] In the first possible implementation, in a possible example, an identifier 13 may be further configured in the bitstream. The identifier 13 may be configured in a sequence parameter set of the bitstream. If affine motion model-based merge mode is used for inter prediction of the to-be-processed block, when the identifier 1 obtained by parsing the bitstream indicates that the candidate motion model for inter prediction of the picture block in the video picture including the to-be-processed block includes the affine motion model, the bitstream is parsed to obtain the identifier 13. When the identifier 13 is a first value, the affine motion model includes only a 4-parameter affine model; or when the identifier 13 is a second value, the affine motion model includes a 4-parameter affine model and a 6-parameter affine model. The first value is different from the second value. For example, the first value is 1, and the second value is 0; or the first value is 0, and the second value is 1.

[00340] For example, the identifier 13 may be represented by `sps_affine_type_flag`. `sps_affine_type_flag` specifies whether 6-parameter affine model based motion compensation is enabled for inter prediction. For example, `sps_affine_type_flag` is equal to 0, the syntax elements shall be constrained such that no 6-parameter affine model based motion compensation is used in the CVS, and `affine_type_flag` is not presented in picture block level syntax in the CVS. (`sps_affine_type_flag` specifies whether 6-parameter affine model based motion compensation can be used for inter prediction. If `sps_affine_type_flag` is equal to 0, the syntax elements shall be constrained such that no 6-parameter affine model based motion compensation is used in the CVS, and `affine_type_flag` is not presented in cu level syntax in the CVS. Otherwise (`sps_affine_type_flag` is equal to 1), 6-parameter affine model based motion compensation may be used in the CVS. When not present, the value of `sps_affine_type_flag` is inferred to be equal to 0.)

[00341] In the first possible implementation, in a possible example, when the identifier 12 indicates that the affine motion model-based AMVP mode is used for inter prediction of the to-be-processed block, and the identifier 13 is the second value, the bitstream needs to be parsed to

motion information candidate list, the method specifically includes the following steps:

[00365] (1) Obtain motion information of control points of the current block. For details, refer to step 601c. Details are not described herein again.

[00366] (2) Combine the motion information of the control points to obtain constructed control point motion information.

[00367] If $\text{sps_affine_type_flag} = 0$, motion information of two control points is combined to construct a 4-parameter affine motion model. The two control points are combined as follows: {CP1, CP4}, {CP2, CP3}, {CP1, CP2}, {CP2, CP4}, {CP1, CP3}, or {CP3, CP4}. For example, a 4-parameter affine motion model constructed by using a 2-tuple that is based on motion information (for example, 2 MVs) of control points CP1 and CP2 is denoted as Affine (CP1, CP2).

[00368] If $\text{sps_affine_type_flag} = 1$, motion information of three control points is combined to construct a 6-parameter affine motion model. The three control points are combined as follows: {CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, or {CP1, CP3, CP4}. For example, a 6-parameter affine motion model constructed by using a 3-tuple that is based on motion information (for example, 3 MVs) of control points CP1, CP2, and CP3 is denoted as Affine (CP1, CP2, CP3). For example, if $\text{sps_affine_type_flag} = 1$, in addition to combining the motion information of the three control points to construct a 6-parameter affine motion model, motion information of two control points may be further combined to construct a 4-parameter affine motion model.

[00369] It should be noted that a combination of different control points may also be converted into control points in a same location. For a specific conversion method, refer to related descriptions in step 502. Details are not described herein again.

[00370] (3) Add the constructed control point motion information to the candidate motion vector list.

[00371] If a length of the candidate list is less than a maximum list length MaxNumMrgCand at this time, these combinations are traversed in a preset order, to obtain a valid combination as candidate control point motion information. If the candidate motion vector list is empty at this time, the candidate control point motion information is added to the candidate motion vector list. Otherwise, motion information in the candidate motion vector list is sequentially traversed, to check whether motion information that is the same as the candidate control point motion information exists in the candidate motion vector list. If no motion information that is the same as the candidate control point motion information exists in the candidate motion vector list, the

prediction of the picture block of the slice including the to-be-processed block does not include an affine motion model, parameter information related to the affine motion model does not need to be parsed when inter prediction is performed on the picture block of the slice. For example, the bitstream does not need to be parsed to obtain the identifier 12.

5 **[00382]** For example, when the identifier 2 is a value A1, it indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block includes an affine motion model, or when the identifier 2 is a value B1, it indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block includes an affine motion model. For example, when the identifier 2 is equal to 1,
10 it indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block includes an affine motion model, or when the identifier 2 is equal to 0, it indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block does not include an affine motion model.

[00383] Step 903: Determine, based on the identifier 12, a prediction mode for inter prediction
15 of the to-be-processed block, where the prediction mode includes an affine motion model-based merge mode, an affine motion model-based AMVP mode, and a non-affine motion model-based prediction mode.

[00384] For example, the identifier 2 may be represented by a syntax element `slice_affine_enabled_flag`. `slice_affine_enabled_flag` specifies whether affine model based motion
20 compensation is enabled for inter prediction. For example, `slice_affine_enabled_flag` is equal to 0, the syntax elements shall be constrained such that no affine model based motion compensation is used for the slice, and `affine_merge_flag`, `affine_inter_flag` and `affine_type_flag` are not presented in picture block level syntax of the CVS. `slice_affine_enabled_flag` is equal to 1, affine model based motion compensation may be used for the slice. (`slice_affine_enabled_flag` specifies
25 whether affine model based motion compensation can be used for inter prediction. If `slice_affine_enabled_flag` is equal to 0, the syntax elements shall be constrained such that no affine model based motion compensation is used for the slice, and `affine_merge_flag`, `affine_inter_flag` and `affine_type_flag` are not presented in cu level syntax of the CVS. Otherwise (`slice_affine_enabled_flag` is equal to 1), affine model based motion compensation may be used
30 in the slice).

[00385] In the second possible implementation, in a possible example, an identifier 23 may be

further configured in the bitstream. The identifier 23 may be configured in a slice header of a slice of the bitstream. When the identifier 2 obtained by parsing the bitstream indicates that the candidate motion model for inter prediction of the slice including the to-be-processed block includes the affine motion model, the bitstream is further parsed to obtain the identifier 23. When the identifier 23 is a first value, the affine motion model includes only a 4-parameter affine model. When the identifier 23 is a second value, the affine motion model includes a 4-parameter affine model and a 6-parameter affine model. The first value is different from the second value. For example, the first value is 1, and the second value is 0; or the first value is 0, and the second value is 1.

[00386] For example, the identifier 23 may be represented by slice_affine_type_flag. slice_affine_type_flag specifies whether 6-parameter affine model based motion compensation is enabled for inter prediction. For example, slice_affine_type_flag is equal to 0, the syntax elements shall be constrained such that no 6-parameter affine model based motion compensation is used for the slice, and affine_type_flag is not presented in picture block level syntax in the CVS. (slice_affine_type_flag specifies whether 6-parameter affine model based motion compensation can be used for inter prediction. If slice_affine_type_flag is equal to 0, the syntax elements shall be constrained such that no 6-parameter affine model based motion compensation is used in the slice, and affine_type_flag is not presented in cu level syntax in the CVS. Otherwise (slice_affine_type_flag is equal to 1), 6-parameter affine model based motion compensation may be used in the slice. When not present, the value of slice_affine_type_flag is inferred to be equal to 0).

[00387] For example, for syntax structures for parsing the identifier 2 and the identifier 23, refer to Table 5.

Table 5

slice_segment_header() {	Descriptor
...	
slice_affine_enabled_flag	u(1)
if(slice_affine_enabled_flag)	
slice_affine_type_flag	u(1)

the video picture including the to-be-processed block does not include an affine motion model, it indicates that the bitstream does not include the identifier 2.

[00423] Step 1103: When the identifier 2 indicates that a candidate motion model for inter prediction of a picture block in a slice including the to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine motion model, parse the bitstream to obtain an identifier 12.

[00424] For related descriptions of the preset condition, the identifier 1, the identifier 2, and the identifier 12, refer to the embodiments corresponding to FIG. 9 and FIG. 10. Details are not described herein again.

[00425] Step 1104: Determine, based on the identifier 12, a prediction mode for inter prediction of the to-be-processed block, where the prediction mode includes an affine motion model-based merge mode, an affine motion model-based AMVP mode, and a non-affine motion model-based mode.

[00426] In the third possible implementation, in a possible example, an identifier 23 may be further configured in the bitstream. The identifier 23 may be configured in a slice header of a slice of the bitstream. When the identifier 2 obtained by parsing the bitstream indicates that the candidate motion model for inter prediction of the picture block in the slice including the to-be-processed block includes the affine motion model, the bitstream is parsed to obtain the identifier 23. When the identifier 23 is a first value, the affine motion model includes only a 4-parameter affine model. When the identifier 23 is a second value, the affine motion model includes a 4-parameter affine model and a 6-parameter affine model. The first value is different from the second value. For example, the first value is 1, and the second value is 0; or the first value is 0, and the second value is 1.

[00427] For example, for syntax structures for parsing the identifier 1, refer to Table 7.

Table 7

seq_parameter_set_rbsp() {	Descriptor
...	
sps_affine_enabled_flag	u(1)
...	

[00433] For example, if `sps_affine_enabled_flag` = 1, `slice_affine_enabled_flag` = 0, and `merge_flag` = 0 in the syntax elements, it indicates that the inter prediction mode of the current block is a translational motion model-based AMVP mode. In this case, `affine_merge_flag`, `affine_inter_flag`, and `affine_type_flag` may not be included in the bitstream, and do not need to be obtained by parsing the bitstream.

[00434] If `sps_affine_enabled_flag` = 1, `slice_affine_enabled_flag` = 1, `merge_flag` = 1, and `affine_merge_flag` = 1 in the syntax elements, it indicates that the inter prediction mode of the current block is an affine motion model-based merge mode.

[00435] For example, if `sps_affine_enabled_flag` = 0, and `merge_flag` = 1 in the syntax elements, it indicates that the inter prediction mode of the current block is a translational motion model-based merge mode. In this case, `affine_merge_flag`, `affine_inter_flag`, and `affine_type_flag` may not be included in the bitstream, and do not need to be obtained by parsing the bitstream.

[00436] For example, if `sps_affine_enabled_flag` = 1, `slice_affine_enabled_flag` = 0, and `merge_flag` = 1, it indicates that the inter prediction mode of the current block is a translational motion model-based merge mode. In this case, `affine_merge_flag`, `affine_inter_flag`, and `affine_type_flag` may not be included in the bitstream, and do not need to be obtained by parsing the bitstream.

[00437] In the third possible implementation, an identifier 13 may be further configured in the bitstream based on the identifier 1. In the third possible implementation, when the identifier 13 indicates that affine motion model-based merge mode is used for inter prediction of the to-be-processed block, the identifier 13 may further indicate whether the bitstream includes the identifier 23.

[00438] After the parsing a bitstream to obtain an identifier 1 in step 1101 and before the parsing the bitstream to obtain an identifier 2 in step 1102, the method further includes: parsing the bitstream to obtain an identifier 13.

[00439] The identifier 13 may be represented by a syntax element `sps_affine_type_flag`. `sps_affine_type_flag` is used to indicate whether 6-parameter affine motion model-based motion compensation is enabled for inter prediction and whether the bitstream includes `slice_affine_type_flag`. For example, `sps_affine_type_flag` = 0, it indicates that 6-parameter affine motion model-based motion compensation is not enabled for inter prediction. In this case, a CVS does not include a syntax element used to indicate 6-parameter affine motion model-based motion

the identifier 1, the identifier 2, the identifier 12, the identifier 13, the identifier 14, and the identifier 23 respectively parsed in a decoding process). It should be understood that construction of the candidate motion vector list on the encoder side is completely consistent with that on a decoder side. A specific embodiment of the encoder side is not described again. However, it should be understood that the video picture prediction method described in this application is also applicable to an encoding apparatus.

[00451] An embodiment of this application further provides an encoding apparatus. As shown in FIG. 14, the apparatus 1400 may include a communications interface 1410 and a processor 1420. Optionally, the apparatus 1400 may further include a memory 1430. The memory 1430 may be disposed inside or outside the apparatus. The processor 1420 sends or receives a video stream or a bitstream through the communications interface 1410.

[00452] In an aspect, the processor 1420 is configured to: encode an identifier 1 into a bitstream; and when the identifier 1 indicates that a candidate motion model for inter prediction of a picture block in a video picture including a to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine motion model, encode an identifier 12 into the bitstream. The identifier 12 is used to determine a prediction mode for inter prediction of the to-be-processed block. The prediction mode includes an affine motion model-based merge mode, an affine motion model-based advanced motion vector prediction AMVP mode, and a non-affine motion model-based prediction mode.

[00453] In another aspect, the processor 1420 is configured to: encode an identifier 3 into a bitstream; and when the identifier 3 indicates that a candidate motion model for inter prediction of a picture block in a slice including a to-be-processed block includes an affine motion model, and the to-be-processed block meets a preset condition for inter prediction using the affine motion model, encode an identifier 32 into the bitstream. The identifier 32 is used to determine a prediction mode for inter prediction of the to-be-processed block. The prediction mode includes an affine motion model-based merge mode, an affine motion model-based advanced motion vector prediction AMVP mode, and a non-affine motion model-based prediction mode.

[00454] The communications interface 1410 in this embodiment of this application may be a circuit, a bus, a transceiver, or any other apparatus that may be configured to exchange information. For example, the other apparatus may be a device connected to the apparatus 1400. For example, when the apparatus is a video encoder, the other apparatus may be a video decoder.

CLAIMS

What is claimed is:

1. A method comprising:

parsing a bitstream to obtain a first identifier;

5 determining that the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence, wherein the video sequence comprises a picture block to be processed;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence, parsing the bitstream to obtain a third
10 identifier, wherein a value of the third identifier is a true value or a false value, and wherein the true value indicates that the affine motion model comprises a 6-parameter affine motion model and the false value indicates that the affine motion model does not comprise the 6-parameter affine motion model;

determining that the picture block satisfies a condition for inter prediction using the affine
15 motion model;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence and ~~the picture~~ the picture block satisfies the condition, parsing the bitstream to obtain a second identifier;

determining that the second identifier indicates that an affine motion model-based advanced
20 motion vector prediction (AMVP) mode is used for inter prediction of the picture block and the value of the third identifier is the true value;

in response to determining that the second identifier indicates that the AMVP mode is used for inter prediction of the picture block and the value of the third identifier is the true value, parsing the bitstream to obtain a fourth identifier;

25 determining that a value of the fourth identifier is a true value or a false value;

in response to determining that the value of the fourth identifier is the true value:

constructing a 3-tuple list as a candidate motion vector predictor list, wherein the 3-tuple list comprises one or more 3-tuples, and wherein each of the one or more 3-tuples represents motion information of three control points; and

30 determining that the affine motion model is the 6-parameter affine motion model; and

in response to determining that the value of the fourth identifier is the false value:

constructing a 2-tuple list as a candidate motion vector predictor list, wherein the 2-tuple list comprises one or more 2-tuples, and wherein each of the one or more 2-tuples represents motion information of two control points; and

5 determining that the affine motion model is the 4-parameter affine motion model; and
performing inter prediction of the picture block based on the candidate motion vector predictor list.

2. The method according to claim 1, wherein parsing the bitstream to obtain the first identifier
10 comprises:

parsing a sequence parameter set of the bitstream to obtain the first identifier.

3. The method according to claim 2, wherein the first identifier is represented by
sps_affine_enabled_flag indicating whether the affine motion model based motion compensation
15 is enabled for the video sequence, and wherein when a value of the sps_affine_enabled_flag is
equal to 1, it indicates that the affine motion model based motion compensation is enabled for the
video sequence.

4. The method according to claim 1, wherein parsing the bitstream to obtain the third identifier
20 comprises:

parsing a sequence parameter set of the bitstream to obtain the third identifier.

5. The method according to claim 4, wherein the third identifier is represented by a sequence
parameter set (SPS) level flag indicating whether the 6-parameter affine motion model based
25 motion compensation is enabled for the video sequence, wherein the SPS level flag has a value of
1 or 0, wherein the value of 1 of the SPS level flag indicates that the 6-parameter affine motion
model based motion compensation is enabled for the video sequence, and wherein the value of 0
of the SPS level flag indicates the 6-parameter affine motion model based motion compensation
is disabled.

30 6. The method according to claim 1, wherein the second identifier is represented by

affine_inter_flag indicating whether the affine motion model-based AMVP mode is used for the picture block when a slice comprising the picture block is a P slice or a B slice, wherein the affine_inter_flag has a value of 1 or 0, wherein the value of 1 of the affine_inter_flag indicates that the affine motion model-based AMVP mode is used for the picture block and wherein the value of 0 of the affine_inter_flag indicates that the affine motion model-based AMVP mode is not used for the picture block.

7. The method according to claim 1, wherein the fourth identifier is represented by affine_type_flag in a coding unit syntax, wherein the affine_type_flag has a value of 1 or 0, indicating whether a 6-parameter affine motion model-based motion compensation is used for inter prediction of the picture block, wherein the value of 1 of the affine_type_flag indicates that the 6-parameter affine motion model-based motion compensation is used for inter prediction of the picture block, and wherein the value of 0 of the affine_type_flag indicates that the 4-parameter affine motion model-based motion compensation is used for inter prediction of the picture block.

8. The method according to claim 1, wherein the condition comprises that a width of the picture block is greater than or equal to a first predetermined threshold, and a height of the picture block is greater than or equal to a second predetermined threshold.

9. The method according to claim 8, wherein the first predetermined threshold equals the second predetermined threshold.

10. The method according to claim 9, wherein the first predetermined threshold and the second predetermined threshold both equals 16.

11. The method according to claim 1, wherein the true value is indicated by 1 and the false value is indicated by 0.

12. A method comprising:

parsing a bitstream to obtain a first identifier;

determining that the first identifier indicates that an affine motion model based motion

compensation is enabled for a video sequence, wherein the video sequence comprises a picture block to be processed;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence:

5 parsing the bitstream to obtain a third identifier;

determining that a value of the third identifier is a true value or a false value; wherein the true value indicates that the affine motion model comprises a 6-parameter affine motion model and the false value indicates that the affine motion model does not comprise the 6-parameter affine motion model;

10 in response to determining that the value of the third identifier is the true value, constructing a candidate motion vector list that comprises a first element, wherein the first element comprises motion information of three control points for constructing the 6-parameter affine motion model; and

in response to determining that the value of the third identifier is a false value,
15 constructing a candidate motion vector list that comprises a second element, wherein the second element comprises motion information of two control points for constructing a 4-parameter affine motion model; and
performing inter prediction of the picture block based on the candidate motion vector list.

20 13. The method according to claim 12, wherein in response to determining that the value of the third identifier is the true value, constructing the candidate motion vector list comprises constructing the candidate motion vector list that further comprises motion information of two control points for constructing the 4-parameter affine motion model.

25 14. The method according to claim 12, wherein parsing the bitstream to obtain the first identifier comprises:

parsing a sequence parameter set of the bitstream to obtain the first identifier.

15. The method according to claim 14, wherein the first identifier is represented by
30 sps_affine_enabled_flag indicating whether the affine motion model based motion compensation is enabled for the video sequence, and wherein when sps_affine_enabled_flag is equal to 1, it

indicates that the affine motion model based motion compensation is enabled for the video sequence.

16. The method according to claim 12, wherein parsing the bitstream to obtain the third
5 identifier comprises:

parsing a sequence parameter set of the bitstream to obtain the third identifier.

17. The method according to claim 16, wherein the third identifier is represented by a
sequence parameter set (SPS) level flag indicating whether the 6-parameter affine motion model
10 based motion compensation is enabled for the video sequence, wherein the SPS level flag has a
value of 1 or 0, wherein the value of 1 of the SPS level flag indicates that the 6-parameter affine
motion model based motion compensation is enabled for the video sequence, and wherein the
value of 0 of the SPS level flag indicates the 6-parameter affine motion model based motion
compensation is disabled.

18. A video data decoding device, comprising:
a memory storage stores video data in a form of a bitstream[[a]];
a video decoder, configured to:
parse a bitstream to obtain a first identifier;
20 determine that the first identifier indicates that an affine motion model based motion
compensation is enabled for a video sequence, wherein the video sequence comprises a picture
block to be processed;

in response to determining that the first identifier indicates that the affine motion model based
motion compensation is enabled, parse the bitstream to obtain a third identifier, wherein a value
25 of the third identifier is a true value or a false value, and wherein the true value indicates that the
affine motion model comprises a 6-parameter affine motion model and the false value indicates
that the affine motion model does not comprise the 6-parameter affine motion model;

determine that the picture block satisfies a condition for inter prediction using the affine
motion model;

30 in response to determining that the first identifier indicates that the affine motion model
based motion compensation is enabled for the video sequence and ~~the picture~~ the picture block

satisfies the condition, parse the bitstream to obtain a second identifier;

determine that the second identifier indicates that an affine motion model-based advanced motion vector prediction (AMVP) mode is used for inter prediction of the picture block and the value of the third identifier is the true value;

5 in response to determining that the second identifier indicates that the AMVP mode is used for inter prediction of the picture block and the value of the third identifier is the true value, parse the bitstream to obtain a fourth identifier;

 determine that a value of the fourth identifier is a true value or a false value;

 in response to determining that the value of the fourth identifier is the true value:

10 construct a 3-tuple list as a candidate motion vector predictor list, wherein the 3-tuple list comprises one or more 3-tuples, and wherein each of the one or more 3-tuples represents motion information of three control points; and

 determine that the affine motion model is the 6-parameter affine motion model; and

 in response to determining that the value of the fourth identifier is the false value:

15 construct a 2-tuple list as a candidate motion vector predictor list, wherein the 2-tuple list comprises one or more 2-tuples, and wherein each of the one or more 2-tuples represents motion information of two control points; and

 determine that the affine motion model is the 4-parameter affine motion model; and

 perform inter prediction of the picture block based on the candidate motion vector predictor

20 list.

19. The device according to claim 18, wherein parse the bitstream to obtain the first identifier comprises:

 parse a sequence parameter set of the bitstream to obtain the first identifier.

25 20. The device according to claim 19, wherein the first identifier is represented by sps_affine_enabled_flag, indicating whether the affine motion model based motion compensation is enabled for the video sequence, and wherein the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence when a value of the
30 sps_affine_enabled_flag is 1.

21. The device according to claim 18, wherein parse the bitstream to obtain the third identifier comprises:

parse a sequence parameter set of the bitstream to obtain the third identifier.

5 22. The device according to claim 21, wherein the third identifier is represented by a sequence parameter set (SPS) level flag indicating whether the 6-parameter affine motion model based motion compensation is enabled for the video sequence, wherein the SPS level flag has a value of 1 or 0, wherein the value of 1 of the SPS level flag indicates that the 6-parameter affine motion model based motion compensation is enabled for the video sequence, and wherein the value of 0
10 of the SPS level flag indicates the 6-parameter affine motion model based motion compensation is disabled.

23. The device according to claim 18, wherein the second identifier is represented by affine_inter_flag indicating whether the affine motion model-based AMVP mode is used for the
15 picture block when a slice comprising the picture block is a P slice or a B slice, wherein the affine_inter_flag has a value of 1 or 0, wherein the value of 1 of the affine_inter_flag indicates that the affine motion model-based AMVP mode is used for the picture block and wherein the value of 0 of the affine_inter_flag indicates that the affine motion model-based AMVP mode is not used for the picture block.

20 24. The device according to claim 18, wherein the fourth identifier is represented by affine_type_flag in a coding unit syntax, wherein the affine_type_flag has a value of 1 or 0, indicating whether a 6-parameter affine motion model-based motion compensation is used for inter prediction of the picture block, wherein the value of 1 of the affine_type_flag indicates that
25 the 6-parameter affine motion model-based motion compensation is used for inter prediction of the picture block, and wherein the value of 0 of the affine_type_flag indicates that the 4-parameter affine motion model-based motion compensation is used for inter prediction of the picture block.

25 25. The device according to claim 18, wherein the condition comprises that a width of the
30 picture block is greater than or equal to a first predetermined threshold, and a height of the picture block is greater than or equal to a second predetermined threshold.

26. The device according to claim 25, wherein the first predetermined threshold equals the second predetermined threshold.

5 27. The device according to claim 26, wherein the first predetermined threshold and the second predetermined threshold both equals 16.

28. The device according to claim 18, wherein the true value is indicated by 1 and the false value is indicated by 0.

10 29. A video data decoding device, comprising:
a non-transitory memory storage, configured to store video data in a form of a bitstream;
a video decoder, configured to:
parse a bitstream to obtain a first identifier;
15 determine that the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence, wherein the video sequence comprises a picture block to be processed;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence:

20 parse the bitstream to obtain a third identifier;
determine that a value of the third identifier is a true value or a false value; wherein the true value indicates that the affine motion model comprises a 6-parameter affine motion model and the false value indicates that the affine motion model does not comprise the 6-parameter affine motion model;

25 in response to determining that the value of the third identifier is the true value, construct a candidate motion vector list ~~that comprises~~ that comprises a first element, wherein the first element comprises motion information of three control points for constructing the 6-parameter affine motion model; and

in response to determining that the value of the third identifier is a false value,
30 construct a candidate motion vector list that comprises a second element, wherein the second element comprises motion information of two control points for constructing a 4-parameter

affine motion model; and

perform inter prediction of the picture block based on the candidate motion vector list.

5 30. The device according to claim 29, wherein in response to determining that a value of the third identifier is the true value, construct the candidate motion vector list comprises construct the candidate motion vector list that further comprises motion information of two control points for constructing the 4-parameter affine motion model.

10 31. The device according to claim 29, wherein parse the bitstream to obtain the first identifier comprises:

parse a sequence parameter set of the bitstream to obtain the first identifier.

15 32. The device according to claim 31, wherein the first identifier is represented by `sps_affine_enabled_flag`, indicating whether the affine motion model based motion compensation is enabled for the video sequence, and wherein the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence when a value of the `sps_affine_enabled_flag` is 1.

20 33. The device according to claim 29, wherein parse the bitstream to obtain the third identifier comprises:

parse a sequence parameter set of the bitstream to obtain the third identifier.

25 34. The device according to claim 33, wherein the third identifier is represented by a sequence parameter set (SPS) level flag indicating whether the 6-parameter affine motion model based motion compensation is enabled for the video sequence, wherein the SPS level flag has a value of 1 or 0, wherein the value of 1 of the SPS level flag indicates that the 6-parameter affine motion model based motion compensation is enabled for the video sequence, and wherein the value of 0 of the SPS level flag indicates the 6-parameter affine motion model based motion compensation is disabled.

30 35. A computer-readable storage medium, wherein the computer-readable storage medium

stores one or more instructions executable by a computer to perform operations comprising:

parsing a bitstream to obtain a first identifier;

determining that the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence, wherein the video sequence comprises a picture

5 block to be processed;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence, parsing the bitstream to obtain a third identifier, wherein a value of the third identifier is a true value or a false value, and wherein the true value indicates that the affine motion model comprises a 6-parameter affine motion model and
10 the false value indicates that the affine motion model does not comprise the 6-parameter affine motion model;

determining that the picture block satisfies a condition for inter prediction using the affine motion model;

in response to determining that the first identifier indicates that the affine motion model
15 based motion compensation is enabled for the video sequence and ~~the picture~~ the picture block satisfies the condition, parsing the bitstream to obtain a second identifier;

determining that the second identifier indicates that an affine motion model-based advanced motion vector prediction (AMVP) mode is used for inter prediction of the picture block and the value of the third identifier is the true value;

20 in response to determining that the second identifier indicates that the AMVP mode is used for inter prediction of the picture block and the value of the third identifier is the true value, parsing the bitstream to obtain a fourth identifier;

determining that a value of the fourth identifier is a true value or a false value;

in response to determining that the value of the fourth identifier is the true value:

25 constructing a 3-tuple list as a candidate motion vector predictor list, wherein the 3-tuple list comprises one or more 3-tuples, and wherein each of the one or more 3-tuples represents motion information of three control points; and

determining that the affine motion model is the 6-parameter affine motion model; and
in response to determining that the value of the fourth identifier is the false value:

30 constructing a 2-tuple list as a candidate motion vector predictor list, wherein the 2-tuple list comprises one or more 2-tuples, and wherein each of the one or more 2-tuples

represents motion information of two control points; and

determining that the affine motion model is the 4-parameter affine motion model; and

performing inter prediction of the picture block based on the candidate motion vector predictor list.

5

36. A computer-readable storage medium, wherein the computer-readable storage medium stores one or more instructions executable by a computer to perform operations comprising:

parsing a bitstream to obtain a first identifier;

10 determining that the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence, wherein the video sequence comprises a picture block to be processed;

in response to determining that the first identifier indicates that the affine motion model based motion compensation is enabled for the video sequence:

parsing the bitstream to obtain a third identifier;

15 determining that a value of the third identifier is a true value or a false value; wherein the true value indicates that the affine motion model comprises a 6-parameter affine motion model and the false value indicates that the affine motion model does not comprise the 6-parameter affine motion model;

20 in response to determining that the value of the third identifier is the true value, constructing a candidate motion vector list that comprises a first element, wherein the first element comprises motion information of three control points for constructing the 6-parameter affine motion model; and

25 in response to determining that the value of the third identifier is a false value, constructing a candidate motion vector list that comprises a second element, wherein the second element comprises motion information of two control points for constructing a 4-parameter affine motion model; and

performing inter prediction of the picture block based on the candidate motion vector list.

37. A non-transitory storage medium storing an encoded bitstream for video signals, the
30 encoded bitstream comprises a plurality of syntax elements, wherein the plurality of syntax elements comprises a first identifier, and

wherein a third identifier and a second identifier are conditionally signaled at least based on a value of the first identifier, and a fourth identifier is conditionally signaled at least based on a value of the second identifier and the third identifier,

5 wherein the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence including a picture block to be processed,

wherein the second identifier indicates that an affine motion model-based advanced motion vector prediction (AMVP) mode is used for inter prediction of the picture block, and the third identifier has a true value,

10 wherein a false value of the third identifier indicates that a 6-parameter affine motion model based motion compensation is disabled for the video sequence, and the true value of the third identifier indicates that the 6-parameter affine motion model based motion compensation is enabled for the video sequence, and

15 wherein a true value of the fourth identifier indicates that the 6-parameter affine motion model-based motion compensation is used for inter prediction of the picture block, and the false value of the fourth identifier indicates that a 4-parameter affine motion model-based motion compensation is used for inter prediction of the picture block.

38. A non-transitory storage medium storing an encoded bitstream for video signals, the encoded bitstream comprises a plurality of syntax elements, wherein the plurality of syntax
20 elements comprises a first identifier,

wherein a third identifier is conditionally signaled at least based on a value of the first identifier,

wherein the first identifier indicates that an affine motion model based motion compensation is enabled for a video sequence including a picture block to be processed, and

25 wherein a false value of the third identifier indicates that a 6-parameter affine motion model based motion compensation is disabled for the video sequence, and a true value of the third identifier indicates that the 6-parameter affine motion model based motion compensation is enabled for the video sequence.