

## **ENCODING AND DECODING METHOD AND DEVICE, ENCODER SIDE APPARATUS AND DECODER SIDE APPARATUS**

### **TECHNICAL FIELD**

[01] The present application relates to encoding and decoding technologies, in particular to an encoding and decoding method and apparatus, an encoding-side device, and a decoding-side device.

### **BACKGROUND**

[02] In order to save space, video images are transmitted after being encoded, and the complete video encoding method may include the processes of prediction, transformation, quantization, entropy encoding, filtering and the like. The prediction encoding includes inter coding and intra coding, wherein the inter coding uses the pixels of the adjacent encoded image to predict the pixels of the current image according to the time domain correlation of a video, so as to effectively remove the video time domain redundancy.

[03] In the process of inter coding, a Motion Vector (MV) can be used to represent a relative displacement between a current image block of a video image of a current frame and a reference image block of a video image of a reference frame. For example, if there is a strong time domain correlation between a video image A of the current frame and a video image B of the reference frame, when an image block A1 (a current image block) of the video image A needs to be transmitted, a motion search can be performed in the video image B to find an image block B1 (i.e., a reference image block) that best matches the image block A1, and to determine a relative displacement between the image block A1 and the image block B1, the relative displacement is also the motion vector of the image block A1.

[04] The encoding side can send the motion vector to the decoding side, rather than send the image block A1 to the decoding side. The decoding side can obtain the image block A1 based on the motion vector and the image block B1. Obviously, the above approach can save a large number of bits because the number of bits occupied by the motion vector is smaller than the number of bits occupied by the image block A1.

[05] In the conventional method, when a current block is a unidirectional block, after a motion vector of the current block (hereinafter referred to as an original motion vector) is obtained, the original motion vector can be adjusted, and encoding/decoding can be implemented based on the adjusted motion vector, thereby improving encoding performance. However, when the current block is a bi-directional block, there is no reasonable solution at present that how to adjust the first original motion vector and the second original motion vector after obtaining the first and second original motion vectors of the current block. That is, in the case of a bi-directional block, there may be problems of low prediction quality, prediction error, and the like, resulting in poor encoding performance.

## **SUMMARY**

[06] The present application provides a method, an apparatus and a device for encoding and decoding, which can improve encoding performance.

[07] The present application provides an encoding and decoding method, which includes the following steps: if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference

block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector; encoding or decoding the current block according to the first and second target motion vectors.

[08] The present application provides an encoding and decoding apparatus, which includes: a determining module configured for, if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; a processing module configured for adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector; and an encoding and decoding module configured for encoding or decoding the current block according to the first and second target motion vectors.

[09] The present application provides an encoding-side device, which includes a processor and a machine-readable storage medium, wherein the machine-readable storage medium has stored thereon machine-executable instructions that, upon execution by the processor, cause performance of a set of operations: if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first

original motion vector and a second target motion vector corresponding to the second original motion vector; and encoding the current block according to the first and second target motion vectors.

[10] The present application provides a decoding-side device, which includes a processor and a machine-readable storage medium, wherein the machine-readable storage medium has stored thereon machine-executable instructions that, upon execution by the processor, cause performance of a set of operations: if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector; and decoding the current block according to the first and second target motion vectors.

[11] It can be seen from the above technical solutions that in the embodiment of the present application, the first and second target motion vectors of the current block can be determined according to the first and second original motion vectors, and the current block can be encoded or decoded according to the first and second target motion vectors, instead of directly encoding or decoding the current block according to the first and second original motion vectors, so as to solve problems such as low prediction quality, prediction error, and the like, and improve the encoding performance and the encoding efficiency.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[12] In order to more clearly explain the technical solution in the embodiment of this

application or of the prior art, drawings that need to be used in embodiments and the prior art will be briefly described below. Obviously, the drawings provided below are for only some embodiments of the present application; those skilled in the art can also obtain other drawings based on these drawings.

[13] Fig. 1A is a schematic diagram of interpolation according to an embodiment of the present application;

[14] Fig. 1B is a schematic diagram of a video encoding framework according to an embodiment of the present application;

[15] Fig. 2 is a flowchart of an encoding and decoding method according to an embodiment of the present application;

[16] Fig. 3 is a flowchart of an encoding and decoding method according to an embodiment of the present application;

[17] Fig. 4 is a flowchart of an encoding and decoding method according to an embodiment of the present application;

[18] Fig. 5 is a schematic diagram of reference pixels according to an embodiment of the present application;

[19] Fig. 6 is a schematic diagram of motion vector iteration according to an embodiment of the present application;

[20] Figs. 7A-7E are schematic diagrams of the sequence of candidate points according to an embodiment of the present application;

[21] Fig. 8 is a schematic diagram of the spatial domain reference of a current block according to an embodiment of the present application;

[22] Fig. 9 is a structural diagram of an encoding and decoding apparatus according to an embodiment of the present application;

[23] Fig. 10 is a hardware structure diagram of a decoding-side device according to an embodiment of the present application;

[24] Fig. 11 is a hardware structure diagram of an encoding-side device according to an embodiment of the present application.

## **DETAILED DESCRIPTION**

[25] The terms used in the embodiments of the present application are only for the purpose of describing specific embodiments, not intended to limit the present application. In this application, unless otherwise specified and/or unless the particular context clearly dictates otherwise, the terms “a” or “an” mean at least one, and the term “the” means the at least one. It should be understood that the term "and/or" used herein is meant to encompass any and all possible combinations of one or more of the associated listed items.

[26] It should be understood that although the terms first, second, third, etc. may be used herein to describe various elements, the elements should not be limited to these terms. These terms are only used to distinguish one type of elements from another. For example, first element can also be referred to as second element, and similarly, second element may also be referred to as first element, without departing from the scope of the present application. The word "if" used can be interpreted as "at ..." or "when ..." or "in response to determining ", depending on the context.

[27] The embodiment of this application provides an encoding and decoding method, which may relate to the following concepts:

[28] Intra-frame prediction and inter-frame prediction techniques: the intra-frame prediction technique is to use the pixels of the coded block of the current image to predict current pixels according to the spatial correlation of a video, so as to effectively remove the video spatial domain redundancy. In the intra-frame prediction technique, a plurality of prediction modes are defined, each

of which corresponds to one texture direction (except for the DC mode), and the prediction signal of a current block is generated from a boundary reconstructed pixel value of a block adjacent to the current block in the prediction direction. For example, if the texture of the image is horizontally arranged, the image information can be better predicted by selecting the horizontal prediction mode. Since a video sequence usually contains stronger time domain correlation, the inter-frame prediction technique is to use the pixels of the adjacent encoded image to predict the current pixels of the current image according to the time domain correlation of the video, so as to effectively remove the video time domain redundancy. The inter-frame prediction section of the main video encoding standard adopts a block-based motion compensation technology, and the main principle is to find an optimal matching block in a previously encoded image for each pixel block of a current image, which is called Motion Estimation (ME).

[29] Motion Vector (MV): in inter encoding, motion vectors are used to indicate the relative displacement between the current encoding block and the optimal matching block in its reference image. Each divided block has a corresponding motion vector transmitted to a decoding side, and if the motion vector of each block, particularly divided into small-sized blocks, is independently encoded and transmitted, a considerable number of bits are consumed. In order to reduce the number of bits for encoding the motion vector, the spatial correlation between adjacent image blocks is used to predict the motion vector of the current block to be encoded according to the motion vector of the adjacent encoded block, and then the prediction difference is encoded. Thus, the number of bits representing the motion vector can be effectively reduced. In the process of encoding the motion vector of the current block, the motion vector of the current block is predicted by using the motion vector of the adjacent encoded block, and then the Motion Vector Difference (MVD) between the Motion Vector Prediction (MVP) and the real estimated value of the motion vector is encoded, so that the number of encoding bits of the MV is effectively reduced.

[30] Motion Information: since the motion vector indicates the displacement between the current image block and a certain reference image block, in order to accurately acquire information pointing to the image block, index information of the reference frame image is required in addition to the motion vector to indicate which reference frame image is used. In the video encoding technology, for a current frame image, a reference frame image list may be generally established, and index information of the reference frame image indicates which reference frame image in the reference frame image list is adopted by the current image block. In addition, many encoding techniques also support multiple reference image lists, and therefore, an index value, which may be referred to as a reference direction, may also be used to indicate which reference image list is used. In the video encoding technology, information related to motion, such as a motion vector, a reference frame index, and a reference direction, may be collectively referred to as motion information.

[31] Prediction Signal: the prediction signal is a pixel value derived from a pixel which is already encoded and decoded, a residual is obtained through the difference between an original pixel and the predicted pixel, and then the residual is transformed and quantized, and the coefficients are encoded. In particular, the inter-frame prediction signal refers to a pixel value derived from a reference frame (reconstructed pixel frame) of the current block, and the final predicted pixel needs to be obtained through an interpolation due to the discrete pixel position. The closer the predicted pixel is to the original pixel, the smaller the residual energy obtained by subtracting the predicted pixel from the original pixel is, and the higher the encoding compression performance is.

[32] Interpolation: if the current MV is non-integer pixel precision, the existing pixel value cannot be directly copied from the corresponding reference frame, and the required pixel value can only be obtained through interpolation. As shown in Fig. 1A, if a pixel value  $Y_{1/2}$  with an offset of  $1/2$  pixels is to be obtained, it needs to be obtained by interpolating the surrounding existing pixel values  $X$ . If an interpolation filter with the number of taps

being  $N$  is adopted, the interpolation is obtained by interpolating  $N$  surrounding integer-pixels.

[33] Motion compensation: motion compensation is the process of obtaining all the pixel values of the current block by interpolation or copying.

[34] Video encoding framework: as shown in Fig. 1B, the encoding side processing flow according to the embodiment of the present application may be implemented by using the video encoding framework, and in addition, the schematic diagram of the video decoding framework is similar to that of Fig. 1B, which is not described herein either, and the decoding side processing flow according to the embodiment of the present application may be implemented by using a video decoding framework. Specifically, the video encoding framework and the video decoding framework include modules, such as modules of intra-frame prediction, motion estimation/motion compensation, reference image buffer, in-loop filtering, reconstruction, transformation, quantization, inverse transformation, inverse quantization, entropy encoder and the like. At the encoding side, the processing flow at the encoding side can be realized through the cooperation between these modules, and at the decoding side, the processing flow at the decoding side can be realized through the cooperation between these modules.

[35] In the conventional method, when a current block is a bi-directional block, there is no reasonable solution that how to adjust the first and second original motion vectors after obtaining the first and second original motion vectors of the current block. In the embodiment of the application, when a current block is a bi-directional block, firstly whether the characteristic information of the current block meets a specific condition is judged, if yes, a first reference block corresponding to the current block is determined according to a first original motion vector, a second reference block is determined according to a second original motion vector of the current block, and the first and second

original motion vectors are adjusted according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain first and second target motion vectors.

[36] In this way, the first and second target motion vectors are obtained by adjusting the first and second original motion vectors, so that the current block can be encoded or decoded according to the first and second target motion vectors, and the encoding performance and the encoding efficiency can be improved.

[37] The following describes the encoding and decoding method according to the present application in detail with reference to several specific embodiments.

[38] Embodiment 1: referring to Fig. 2, a schematic flow chart of an encoding and decoding method according to an embodiment of the present application is shown, where the method can be applied to a decoding side or an encoding side, and the method can include the following steps:

[39] At step 201, if characteristic information of a current block meets a specific condition, a first reference block corresponding to the current block is determined according to a first original motion vector of the current block, and a second reference block corresponding to the current block is determined according to a second original motion vector of the current block. The characteristic information includes, but is not limited to, one or more of the following: a motion information prediction mode corresponding to the current block; a motion information attribute corresponding to the current block; size information of the current block.

[40] In an example, when the characteristic information is the motion information prediction mode corresponding to the current block, the characteristic information of the current block meets a specific condition, can include but is not limited to: if the motion information prediction mode corresponding to the current block is a regular merge mode,

determining that the motion information prediction mode corresponding to the current block meets the specific condition; or, if the motion information prediction mode corresponding to the current block is a merge mode for generating a new prediction value by combining an inter-frame prediction value and an intra-frame prediction value, determining that the motion information prediction mode corresponding to the current block meets the specific condition.

[41] In an example, when the characteristic information is a motion information attribute corresponding to the current block, the characteristic information of the current block meets a specific condition, can include but is not limited to: if the motion information attribute corresponding to the current block is that the motion information of the current block includes motion information in two different directions, determining that the motion information attribute corresponding to the current block meets the specific condition; or, if the motion information attribute corresponding to the current block is that the motion information of the current block includes motion information in two different directions, and the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are the same, determining that the motion information attribute corresponding to the current block meets the specific condition; or if the motion information attribute corresponding to the current block is that the current block reuses motion information of surrounding blocks, determining that the motion information attribute corresponding to the current block meets the specific condition; or, if the motion information attribute corresponding to the current block is that motion information of each sub-block of the current block is the same, determining that the motion information attribute corresponding to the current block meets the specific condition.

[42] Exemplarily, the motion information of the current block includes motion information in two different directions, which means that: the motion information of the current block

in two different directions respectively corresponds to a first reference frame and a second reference frame, the first reference frame is located in front of the current frame where the current block is located, and the second reference frame is located behind the current frame where the current block is located.

[43] In an example, when the characteristic information is size information of the current block, the size information of the current block can include a width value and a height value of the current block, the characteristic information of the current block meets a specific condition, can include but is not limited to: if the width value of the current block is within a range of a first interval [a first threshold, a second threshold], determining that the size information of the current block meets the specific condition; or, if the height value of the current block is within a range of a second interval [a third threshold, a fourth threshold], determining that the size information of the current block meets the specific condition; or, if the area derived from the width value and the height value of the current block is within a range of a third interval [a fifth threshold, a sixth threshold], determining that the size information of the current block meets the specific condition; or, if the width value is within the range of the first interval [the first threshold, the second threshold], the height value is within the range of the second interval [the third threshold, the fourth threshold], and the area is within the range of the third interval [the fifth threshold, the sixth threshold], determining that the size information of the current block meets the specific condition.

[44] Exemplarily, the first threshold may be smaller than the second threshold, and neither the first threshold nor the second threshold is limited, for example, the first threshold may be 4, and the second threshold may be 128. The third threshold may be smaller than the fourth threshold, and neither the third threshold nor the fourth threshold is limited, for example, the third threshold may be 8, and the fourth threshold may be 128. The fifth threshold may be smaller than the sixth threshold, and neither the fifth threshold nor the

sixth threshold is limited, for example, the fifth threshold may be 64, and the sixth threshold may be 128\*128.

[45] In an example, determining the first reference block corresponding to the current block according to the first original motion vector of the current block, and determining the second reference block corresponding to the current block according to the second original motion vector of the current block can include, but are not limited to: determining the first reference block corresponding to the current block from the first reference frame based on the first original motion vector of the current block; and determining the second reference block corresponding to the current block from the second reference frame based on the second original motion vector of the current block. The first pixel value of each pixel point in the first reference block is obtained by interpolating the pixel values of adjacent pixel points in the first reference block, or is obtained by copying the pixel values of the adjacent pixel points in the first reference block; the second pixel value of each pixel point in the second reference block is obtained by interpolating the pixel values of adjacent pixel points in the second reference block, or is obtained by copying the pixel values of adjacent pixel points in the second reference block. The size of the first reference block is the same as the size of the second reference block, the width value of the first reference block is determined based on the width value of the current block and the search range, and the height value of the first reference block is determined based on the height value of the current block and the search range.

[46] At step 202, the first and second original motion vectors are adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block, to obtain the first target motion vector corresponding to the first original motion vector and the second target motion vector corresponding to the second original motion vector.

[47] In an example, when the current block includes at least one sub-block, for each sub-block of the current block, the first and second original motion vectors are adjusted according to the first and second pixel values to obtain the first and second target motion vectors of the sub-block.

[48] In an example, adjusting the first and second original motion vectors according to the first and second pixel values to obtain the first and second target motion vectors of the sub-block can include: first and second integer-pixel motion vector adjustment values of the sub-block, and/or first and second sub-pixel motion vector adjustment values of the sub-block can be determined according to the first and second pixel values. Then, the first original motion vector can be adjusted according to the first integer-pixel motion vector adjustment value and/or the first sub-pixel motion vector adjustment value, so as to obtain the first target motion vector of the sub-block. And the second original motion vector can be adjusted according to the second integer-pixel motion vector adjustment value and/or the second sub-pixel motion vector adjustment value, so as to obtain the second target motion vector of the sub-block.

[49] In an example, determining the first and second integer-pixel motion vector adjustment values of the sub-block, the first and second sub-pixel motion vector adjustment values of the sub-block according to the first and second pixel values can include: determining the first original motion vector or the second original motion vector as a center motion vector; determining an edge motion vector corresponding to the center motion vector; obtaining a first cost value corresponding to the center motion vector and a second cost value corresponding to the edge motion vector according to the first and second pixel values; then selecting a motion vector from the center motion vector and the edge motion vector as an optimal motion vector according to the first cost value and the second cost value, and judging whether an end condition is met; if not, determining the optimal motion vector as the center motion vector, and returning to execute determining the edge motion

vector corresponding to the center motion vector; if yes, determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector; and determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector.

[50] In an example, determining the edge motion vector corresponding to the center motion vector includes: shifting the center motion vector  $(x, y)$  towards different directions by  $S$ , and sequentially obtaining edge motion vector  $(x, y+S)$ , edge motion vector  $(x, y-S)$ , edge motion vector  $(x+S, y)$ , edge motion vector  $(x-S, y)$  and edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions, or shifting the center motion vector  $(x, y)$  towards different directions by  $S$ , and sequentially obtaining edge motion vector  $(x, y-S)$ , edge motion vector  $(x, y+S)$ , edge motion vector  $(x-S, y)$ , edge motion vector  $(x+S, y)$  and edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions. The default value of the edge motion vector  $(x+\text{right}, y+\text{down})$  is  $(x-S, y-S)$ ; if the cost value of the edge motion vector  $(x+S, y)$  is smaller than the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y+S)$  is less than that of the edge motion vector  $(x, y-S)$ , down is  $S$ ; or if the cost value of the edge motion vector  $(x+S, y)$  is less than or equal to the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y+S)$  is less than or equal to the cost value of the edge motion vector  $(x, y-S)$ , down is  $S$ .

[51] In an example, obtaining the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values can include, but is not limited to: acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values without down-sampling; alternatively, can include performing down-sampling operation on the first and second pixel values; and obtaining the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first

and second pixel values after down-sampling; alternatively, can include performing shifting and down-sampling operation on the first pixel value, and performing shifting and down-sampling operation on the second pixel value; and acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values after shifting and down-sampling.

[52] In an example, determining the first and second integer-pixel motion vector adjustment values of the sub-block, the first and second sub-pixel motion vector adjustment values of the sub-block according to the first and second pixel values can include: taking the first original motion vector or the second original motion vector as a center, selecting some or all of the motion vectors from surrounding motion vectors, and taking the selected motion vectors as candidate motion vectors; acquiring a third cost value corresponding to the first original motion vector or the second original motion vector and a fourth cost value corresponding to each of the candidate motion vectors according to the first and second pixel values; then selecting a motion vector from the first original motion vector or the second original motion vector and the candidate motion vector as an optimal motion vector according to the third cost value and the fourth cost value; determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector; and determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector.

[53] Determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector can include, but is not limited to: determining the first integer-pixel motion vector adjustment value of the sub-block according to the optimal motion vector and the first original motion vector, and determining the second integer-pixel motion vector adjustment value of the sub-block according to the first integer-pixel motion vector adjustment value.

[54] Determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector, can include: determining the first sub-pixel motion vector adjustment value of the sub-block according to the cost value corresponding to the optimal motion vector and the cost value corresponding to the edge motion vector corresponding to the optimal motion vector, and determining the second sub-pixel motion vector adjustment value of the sub-block according to the first sub-pixel motion vector adjustment value.

[55] At step 203, the current block is encoded or decoded according to the first and second target motion vectors. For example, referring to the encoding side, the encoding side can encode the current block according to the first and second target motion vectors; referring to the decoding side, the decoding side can decode the current block according to the first and second target motion vectors.

[56] In an example, for each sub-block of the current block, a third reference block corresponding to the sub-block is determined according to the first target motion vector of the sub-block, and a fourth reference block corresponding to the sub-block is determined according to the second target motion vector of the sub-block; a third pixel value of the third reference block and a fourth pixel value of the fourth reference block are weighted to obtain a prediction value of the sub-block; and the prediction value of the current block is determined according to the prediction value of each sub-block.

[57] In an example, determining the third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining the fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block can include:

[58] implementation 1: determining a fifth reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, and

interpolating pixel values in the fifth reference block to obtain the third reference block. And determining a sixth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, and interpolating pixel values in the sixth reference block to obtain the fourth reference block.

[59] implementation 2: determining a seventh reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, constructing an eighth reference block by using pixel values in the seventh reference block, and interpolating pixel values in the eighth reference block to obtain the third reference block; and determining a ninth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, constructing a tenth reference block by using pixel values in the ninth reference block, and interpolating pixel values in the tenth reference block to obtain the fourth reference block.

[60] In an example, after encoding or decoding the current block according to the first and second target motion vectors, the first and second target motion vectors for the current block can be further stored. In this way, the first and second target motion vectors are used for loop filtering of the current frame, the first and second target motion vectors are used for time domain reference of subsequent frames, and/or the first and second target motion vectors are used for spatial domain reference of the current frame.

[61] As can be seen from the above technical solutions, in the embodiment of the present application, the first and second target motion vectors of the current block can be determined according to the first and second original motion vectors, and the current block can be encoded or decoded according to the first and second target motion vectors, instead of being encoded or decoded directly according to the first and second original motion vectors, so as to solve problems such as low prediction quality, prediction error, and the

like, and improve the encoding performance and the encoding efficiency.

[62] Embodiment 2: referring to Fig. 3, a schematic flow chart of an encoding and decoding method according to an embodiment of the present application is shown, where the encoding and decoding method can be applied to an encoding side, and the encoding and decoding method can include the following steps:

[63] At step 301, the encoding side judges whether the characteristic information of the current block meets a specific condition. If the characteristic information of the current block meets the specific condition, step 302 is executed; and if the characteristic information of the current block does not meet the specific condition, the motion vector adjustment method according to the present application is not required to be adopted, and the processing method is not limited.

[64] In an example, the encoding side can judge whether the characteristic information of the current block meets the specific condition based on the encoded information. If the characteristic information of the current block meets the specific condition, it indicates that the motion information of the current block is not accurate enough, and therefore, the motion vector refinement mode is enabled, and step 302 is performed.

[65] If the characteristic information of the current block does not meet the specific condition, it indicates that the motion information of the current block is accurate enough, the motion vector refinement mode is thus not enabled, and the motion vector refinement mode according to the present application is not needed.

[66] In an example, the characteristic information of the current block includes, but is not limited to, one or more of the following: the motion information prediction mode corresponding to the current block; the motion information attribute corresponding to the current block; size information of the current block.

[67] For the method of judging whether the characteristic information of the current block

meets the specific condition, refer to the following embodiments.

[68] At step 302, the encoding side determines a first reference block corresponding to the current block from a first reference frame based on a first original motion vector of the current block; and determines a second reference block corresponding to the current block from a second reference frame based on the second original motion vector of the current block. The pixel value of each pixel point in the first reference block is referred to as a first pixel value, and the pixel value of each pixel point in the second reference block is referred to as a second pixel value for distinguishing.

[69] In an example, if the current block is a block using bi-directional prediction, then the current block has bi-directional motion information including two reference frames and two original motion vectors. For example, the encoding side can acquire the bi-directional motion information in a conventional method, which method is not limited. The bi-directional motion information includes a first reference frame and a first original motion vector, a second reference frame and a second original motion vector.

[70] The encoding side can determine the first reference block corresponding to the current block from the first reference frame based on the first original motion vector, and the pixel value of each pixel point in the first reference block can be referred to as the first pixel value.

[71] The encoding side can determine the second reference block corresponding to the current block from the second reference frame based on the second original motion vector, and the pixel value of each pixel point in the second reference block can be referred to as the second pixel value.

[72] In an example, the distance between the current frame where the current block is located and the first reference frame, and the distance between the second reference frame and the current frame where the current block is located may be the same. For example, the

first reference frame is the 1st frame, the current frame is the 5th frame, and the second reference frame is the 9th frame. Obviously, the above is only an example, and the two distances can also be different.

[73] The first and second original motion vectors are configured in symmetrically mirrored relation with one another, e.g., the first original motion vector is (4, 4), and the second original motion vector is (-4, -4); the first original motion vector is (2.5, 3.5) and the second original motion vector is (-2.5, -3.5). Obviously, the above is only an example, and there may be no symmetrically mirrored relation between the first and second original motion vectors.

[74] For the method of determining the first and second reference blocks, refer to the following embodiments.

[75] At step 303, for each sub-block of the current block: the encoding side adjusts the first original motion vector according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the encoding side adjusts the second original motion vector according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[76] In an example, if the motion vector refinement mode is enabled for the current block, the encoding side can perform fine adjustment on the first and second original motion vectors by using a local search method based on the first and second pixel values to obtain better first target motion vector and second target motion vector, and then generate a prediction value with less distortion by using the first and second target motion vectors.

[77] In an example, the current block can include at least one sub-block, and if only one sub-block is included, the sub-block is the current block itself. For the sub-block, the sub-block corresponds to the first and second original motion vectors, and after adjustment,

the sub-block corresponds to the first and second target motion vectors.

[78] If the current block includes a sub-block A and a sub-block B, for the sub-block A, the sub-block A corresponds to a first original motion vector A1 and a second original motion vector A2, and after adjustment, the sub-block A corresponds to a first target motion vector A3 and a second target motion vector A4. For the sub-block B, the sub-block B corresponds to a first original motion vector B1 and a second original motion vector B2, and after adjustment, the sub-block B corresponds to a first target motion vector B3 and a second target motion vector B4.

[79] The first original motion vector A1 corresponding to the sub-block A and the first original motion vector B1 corresponding to the sub-block B may be the same, and are both the first original motion vectors of the current block; the second original motion vector A2 corresponding to the sub-block A and the second original motion vector B2 corresponding to the sub-block B may be the same, and are both the second original motion vectors of the current block.

[80] Since the first original motion vectors of each sub-block are adjusted separately, the first target motion vector A3 corresponding to the sub-block A and the first target motion vector B3 corresponding to the sub-block B may be the same or different. Since the second original motion vectors of each sub-block is adjusted separately, the second target motion vector A4 corresponding to the sub-block A and the second target motion vector B4 corresponding to the sub-block B may be the same or different.

[81] For the method of adjusting the original motion vectors, refer to the following embodiments, which are not described herein in detail.

[82] At step 304, the encoding side performs motion compensation on each sub-block of the current block according to the first and second target motion vectors. For example, if the current block includes a sub-block A and a sub-block B, the first and second target

motion vectors of the sub-block A can be used to perform motion compensation on the sub-block A, and the first and second target motion vectors of the sub-block B can be used to perform motion compensation on the sub-block B.

[83] For the motion compensation method, refer to the following embodiments, which are not described herein in detail.

[84] At step 305, the encoding side stores the first and second target motion vectors of each sub-block of the current block, and the first and second target motion vectors are used for encoding reference of a subsequent block.

[85] For the implementation of the encoding reference, refer to the following embodiments, which are not described herein in detail.

[86] Embodiment 3: referring to Fig. 4, a schematic flow chart of an encoding and decoding method according to an embodiment of the present application is shown, where the encoding and decoding method can be applied to a decoding side, and the encoding and decoding method can include the following steps:

[87] at step 401, the decoding side judges whether the characteristic information of the current block meets a specific condition. If the characteristic information of the current block meets the specific condition, step 402 is executed, and if the characteristic information of the current block does not meet the specific condition, the motion vector adjustment method according to the present application is not required to be adopted, and the processing method is not limited.

[88] In an example, the decoding side can receive encoded bitstream sent by the encoding side and determine whether the characteristic information of the current block meets the specific condition based on decoded information. If the characteristic information of the current block meets the specific condition, it indicates that the motion information of the current block is not accurate enough, and therefore, the motion vector refinement mode is

enabled, and step 402 is performed. If the characteristic information of the current block does not meet the specific condition, it indicates that the motion information of the current block is accurate enough, the motion vector refinement mode is thus not enabled, and the motion vector refinement mode according to the present application is not needed.

[89] At step 402, the decoding side determines a first reference block corresponding to the current block from a first reference frame based on a first original motion vector of the current block; and determines a second reference block corresponding to the current block from a second reference frame based on the second original motion vector of the current block. The pixel value of each pixel point in the first reference block is referred to as a first pixel value, and the pixel value of each pixel point in the second reference block is referred to as a second pixel value for distinguishing.

[90] At step 403, for each sub-block of the current block: the decoding side adjusts the first original motion vector according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the decoding side adjusts the second original motion vector according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[91] At step 404, the decoding side performs motion compensation on each sub-block of the current block according to the first and second target motion vectors of the sub-block.

[92] At step 405, the decoding side stores the first and second target motion vectors of each sub-block of the current block, and the first and second target motion vectors are used for encoding reference of a subsequent block.

[93] Embodiment 4: step 301 and step 401 involve judging whether the characteristic information of the current block meets a specific condition. In this embodiment, when the characteristic information meets all of the following conditions, it is determined that the

characteristic information of the current block meets the specific condition, and the motion vector refinement mode is enabled. Otherwise, the motion vector refinement mode is not enabled.

[94] The motion information attribute corresponding to the current block includes: the current block directly multiplexing motion information of surrounding blocks without encoding a motion information difference. The motion information attribute corresponding to the current block includes: the motion information of each sub-block of the current block being the same, i.e., not adopting the sub-block motion information prediction mode. The motion information attribute corresponding to the current block includes: the motion information of the current block including motion information in two different directions. The size information of the current block includes: the size of the current block being within a defined range, which will be described in following embodiments.

[95] Embodiment 5: step 301 and step 401 involve judging whether the characteristic information of the current block meets a specific condition. In this embodiment, when the characteristic information meets all of the following conditions, it is determined that the characteristic information of the current block meets the specific condition, and the motion vector refinement mode is enabled. Otherwise, the motion vector refinement mode is not enabled.

[96] The motion information prediction mode corresponding to the current block adopts a Regular merge mode or a CIIP mode (namely a merge mode for generating a new prediction value by combining the inter-frame prediction value and the intra-frame prediction value); alternatively, the motion information prediction mode corresponding to the current block does not adopt a MMVD mode (i.e., a merge mode for encoding motion information difference values), a SB Merge mode (i.e., a merge mode for using sub-block motion information), or a TPM mode (i.e., a merge mode for triangle prediction).

[97] The motion information attribute corresponding to the current block includes: the motion information of the current block includes motion information in two different directions, and the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are the same. The size information of the current block includes: the size of the current block is within a defined range.

[98] It should be noted that the Merge mode includes, but is not limited to, the following types of Merge modes: a Regular merge mode (i.e., a regular merge mode); a merge mode for triangle prediction (a TPM mode); a merge mode for encoding motion information difference values (a MMVD mode); a merge mode for using sub-block motion information (a SB Merge mode); and a merge mode for generating a new prediction value by combining the inter-frame prediction value and the intra-frame prediction value (a CIIP mode).

[99] In an example, the Regular merge mode includes a regular merge mode where no residuals are encoded (i.e., skip mode) and a regular merge mode where residuals are encoded. The MMVD mode includes a MMVD mode in which a residual is not encoded.

[100] Embodiment 6: step 301 and step 401 involve judging whether the characteristic information of the current block meets a specific condition. In this embodiment, when the characteristic information meets all of the following conditions, it is determined that the characteristic information of the current block meets the specific condition, and the motion vector refinement mode is enabled. Otherwise, the motion vector refinement mode is not enabled.

[101] The motion information prediction mode corresponding to the current block adopts the Regular merge mode (i.e., a regular merge mode), the CIIP mode (namely a merge mode for generating a new prediction value by combining the inter-frame prediction value and the intra-frame prediction value), or the TPM mode (i.e., a merge mode for

triangle prediction); alternatively, the motion information prediction mode corresponding to the current block does not adopt the MMVD mode (i.e., a merge mode for encoding motion information difference values), or the SB Merge mode (i.e., a merge mode for using sub-block motion information).

[102] The motion information attribute corresponding to the current block includes: the motion information of the current block includes motion information in two different directions, and the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are the same. The size information of the current block includes: the size of the current block is within a defined range.

[103] Embodiment 7: step 301 and step 401 involve judging whether the characteristic information of the current block meets a specific condition. In this embodiment, when the characteristic information meets all of the following conditions, it is determined that the characteristic information of the current block meets the specific condition, and the motion vector refinement mode is enabled. Otherwise, the motion vector refinement mode is not enabled.

[104] The motion information prediction mode corresponding to the current block adopts the Regular merge mode (i.e., a regular merge mode), the CIIP mode (namely a merge mode for generating a new prediction value by combining the inter-frame prediction value and the intra-frame prediction value), the TPM mode (i.e., a merge mode for triangle prediction), or the MMVD mode (i.e., a merge mode for encoding motion information difference values); alternatively, the motion information prediction mode corresponding to the current block does not adopt the SB Merge mode (i.e., a merge mode for using sub-block motion information).

[105] The motion information attribute corresponding to the current block includes: the

motion information of the current block includes motion information in two different directions, and the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are the same. The size information of the current block includes: the size of the current block is within a defined range.

[106] Embodiment 8: step 301 and step 401 involve judging whether the characteristic information of the current block meets a specific condition. In this embodiment, when the characteristic information meets all of the following conditions, it is determined that the characteristic information of the current block meets the specific condition, and the motion vector refinement mode is enabled. Otherwise, the motion vector refinement mode is not enabled.

[107] The motion information prediction mode corresponding to the current block adopts the Regular merge mode or the CIIP mode; the motion information prediction mode corresponding to the current block does not adopt the MMVD mode, the SB Merge mode, or the TPM mode; alternatively, the motion information prediction mode corresponding to the current block adopts the Regular merge mode, the CIIP mode, or the TPM mode; the motion information prediction mode corresponding to the current block does not adopt the MMVD mode or the SB Merge mode; alternatively, the motion information prediction mode corresponding to the current block adopts the Regular merge mode, the CIIP mode, or the TPM mode, or the MMVD mode; the motion information prediction mode corresponding to the current block does not adopt the SB Merge mode.

[108] The motion information attribute corresponding to the current block includes: the motion information of the current block includes motion information in two different directions, and it is not limited that the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are

the same, i.e. this limitation condition is removed. The size information of the current block includes: the size of the current block is within a defined range.

[109] Embodiment 9: regarding "the size of the current block is within a defined range" in the above embodiments 4 to 8, if the size of the current block meets the following condition, it is determined that the size of the current block is within the defined range.

[110] The width value of the current block is within a range of [first threshold, second threshold], wherein the first threshold may be 4, and the second threshold may be 128. The height of the current block is within a range of [third threshold, fourth threshold], wherein the third threshold may be 8, and the fourth threshold may be 128. For example, the first threshold, the second threshold, the third threshold, and the fourth threshold can each be  $2^n$ , where  $n$  is an integer greater than or equal to 1. The area of the current block is within a range of [fifth threshold, sixth threshold], wherein the fifth threshold may be 64, and the sixth threshold may be  $128*128=16384$ . For example, the fifth threshold and the sixth threshold can both be  $4^m$ , where  $m$  is an integer greater than or equal to 1. In the above example, the expression  $[a, b]$  can represent a value greater than or equal to  $a$  and less than or equal to  $b$ .

[111] Embodiment 10: at step 302 and step 402, a first reference block corresponding to the current block is determined from a first reference frame based on a first original motion vector, and the pixel value of each pixel point in the first reference block is referred to as a first pixel value; a second reference block corresponding to the current block is determined from a second reference frame based on the second original motion vector, and the pixel value of each pixel point in the second reference block is referred to as a second pixel value, which is described below.

[112] A first pixel value of each pixel point in the first reference block is obtained by interpolating the pixel values of adjacent pixel points in the first reference block, or is

obtained by copying the pixel values of adjacent pixel points in the first reference block; a second pixel value of each pixel point in the second reference block is obtained by interpolating the pixel values of adjacent pixel points in the second reference block, or is obtained by copying the pixel values of adjacent pixel points in the second reference block. The size of the first reference block is the same as the size of the second reference block, the width value of the first/second reference block is determined based on the width value of the current block and the search range, and the height value of the first/second reference block is determined based on the height value of the current block and the search range.

[113] Referring to Fig. 5, assuming that the current block has a width  $W$  and a height  $H$ , the first original motion vector is denoted as  $MV0$ , and the second original motion vector is denoted as  $MV1$ . The integer-pixel block with the area of  $(W+FS-1)*(H+FS-1)$  is obtained from the corresponding position of the first original motion vector  $MV0$  in the first reference frame, and the integer-pixel block is marked as an integer-pixel block A. And the integer-pixel block with the area of  $(W+FS-1)*(H+FS-1)$  is obtained from the corresponding position of the second original motion vector  $MV1$  in the second reference frame, and the integer-pixel block is marked as an integer-pixel block B.

[114] In one example, based on the integer-pixel block A with the area of  $(W+FS-1)*(H+FS-1)$ , an initial reference pixel block with a size of  $(W+2*SR)*(H+2*SR)$  can be obtained by means of bilinear interpolation, and the initial reference pixel block can be denoted as a first reference block. Based on the integer-pixel block B with the area of  $(W+FS-1)*(H+FS-1)$ , an initial reference pixel block with a size of  $(W+2*SR)*(H+2*SR)$  can be obtained by means of bilinear interpolation, and the initial reference pixel block can be denoted as a second reference block.

[115] In another example, based on the integer-pixel block A with the area of  $(W+FS-1)*(H+FS-1)$ , an initial reference pixel block with a size of  $(W+2*SR)*(H+2*SR)$

can be obtained by means of direct copying (without interpolation), and the initial reference pixel block can be denoted as a first reference block. Based on the integer-pixel block B with the area of  $(W+FS-1)*(H+FS-1)$ , an initial reference pixel block with a size of  $(W+2*SR)*(H+2*SR)$  can be obtained by means of direct copying, and the initial reference pixel block can be denoted as a second reference block.

[116] Exemplarily, for the luminance component only (since the subsequent search process calculates a cost value only by using the luminance component to reduce complexity), initial reference pixel blocks with the size of  $(W+2*SR)*(H+2*SR)$  can be obtained based on integer-pixel blocks (e.g., integer-pixel block A and integer-pixel block B) with the area of  $(W+FS-1)*(H+FS-1)$ , i.e., a first reference block (e.g., Pred\_Inter0) and a second reference block (e.g., Pred\_Inter1).

[117] In an example, FS can be a number of taps of the interpolation filter, e.g. FS can be 8, etc.

[118] Obtaining the first/second reference block by means of bilinear interpolation is that: the pixel value of each pixel point in the first/second reference block is obtained by interpolating the pixel values of adjacent pixel points in the first/second reference block, which however is not limited hereto. Obtaining the first/second reference block by copying is that: the pixel value of each pixel point in the first/second reference block is obtained by copying the pixel values of adjacent pixel points in the first/second reference block, which however is not limited hereto.

[119] Referring to the above embodiment, the first reference block has an area of  $(W+2*SR)*(H+2*SR)$ , the second reference block has an area of  $(W+2*SR)*(H+2*SR)$ , that is, the width value of the first/second reference block is  $W+2*SR$ , and the height value of the first/second reference block is  $H+2*SR$ . W is the width of the current block, H is the height of the current block, SR is the search range, i.e. the number of iterations of the

following embodiments, SR is the maximum horizontal/vertical component interpolation of the target motion vector and the original motion vector, e.g. SR can be 2, etc.

[120] For the first and second reference blocks, they are used for the motion vector adjustment in the subsequent process.

[121] Embodiment 11: in step 303 and step 403, for each sub-block of the current block, the first original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the second original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[122] The adjustment process of the original motion vector is described by the processing procedure of one sub-block (e.g. each  $dx*dy$  sized sub-block of the current block, e.g. a  $16*16$  sized sub-block, or a smaller block), for example.

[123] At step a1, a first original motion vector or a second original motion vector is determined as a center motion vector.

[124] For example, assuming that the first original motion vector is (4, 4) and the second original motion vector is (-4, -4), the first original motion vector (4, 4) or the second original motion vector (-4, -4) can be determined as the center motion vector. For convenience of description, taking determining the first original motion vector (4, 4) as the center motion vector for example, the procedure of determining the second original motion vector (-4, -4) as the center motion vector is similar.

[125] At step a2, an edge motion vector corresponding to the center motion vector is determined.

[126] For example, the center motion vector (x, y) is shifted towards different

directions by  $S$ , and an edge motion vector  $(x, y+S)$ , an edge motion vector  $(x, y-S)$ , an edge motion vector  $(x+S, y)$ , an edge motion vector  $(x-S, y)$  and an edge motion vector  $(x+right, y+down)$  in different directions are sequentially obtained, wherein right can be  $S$  or  $-S$ , and down can be  $S$  or  $-S$ , the way how right and down are determined refers to the following embodiments.

[127] Referring to Fig. 6, the center motion vector  $(x, y)$  is taken as the center, that is, the center motion vector is  $(0, 0)$ , and  $S$  is 1, right and down are both 1 for example, then the edge motion vectors corresponding to the center motion vector  $(0, 0)$  include: an edge motion vector  $(0, 1)$ , an edge motion vector  $(0, -1)$ , an edge motion vector  $(1, 0)$ , an edge motion vector  $(-1, 0)$ , and an edge motion vector  $(1, 1)$ .

[128] At step a3, a first cost value corresponding to the center motion vector and a second cost value corresponding to each edge motion vector are obtained according to the first pixel value of the first reference block and the second pixel value of the second reference block.

[129] For example, a sub-reference block A1 corresponding to the center motion vector  $(0, 0)$  is obtained by copying from the first reference block, and the sub-reference block A1 is a sub-reference block of the center motion vector  $(0, 0)$  in the first reference block. The size of the sub-reference block A1 is the size of the current block, and the sub-reference block A1 is the most center  $W*H$  block in the first reference block.

[130] A sub-reference block B1 corresponding to the center motion vector  $(0, 0)$  is obtained by copying from the second reference block, and the sub-reference block B1 is a sub-reference block of the center motion vector  $(0, 0)$  in the second reference block. The size of the sub-reference block B1 is the size of the current block, and the sub-reference block B1 is the most center  $W*H$  block in the second reference block.

[131] Then, a cost value 1 corresponding to the center motion vector  $(0, 0)$  is obtained

according to the first pixel value of the sub-reference block A1 and the second pixel value of the sub-reference block B1, and the method of determining the cost value refers to the following embodiments.

[132] A sub-reference block A2 corresponding to the edge motion vector (0, 1) is obtained by copying from the first reference block, and the sub-reference block A2 is a sub-reference block of the edge motion vector (0, 1) in the first reference block. The size of the sub-reference block A2 is the size of the current block, and the sub-reference block A2 is the most center  $W \times H$  block in the first reference block.

[133] A sub-reference block B2 corresponding to the motion vector (0, -1) symmetrical to the edge motion vector (0, 1) is obtained by copying from the second reference block, and the sub-reference block B2 is a sub-reference block of the symmetric motion vector (0, -1) in the second reference block. The size of the sub-reference block B2 is the size of the current block, and is a block  $W \times H$  in the second reference block.

[134] Then, a cost value 2 corresponding to the edge motion vector (0, 1) is obtained according to the first pixel value of the sub-reference block A2 and the second pixel value of the sub-reference block B2, and the method of determining the cost value refers to the following embodiments.

[135] Based on the method of determining the cost value 2 corresponding to the edge motion vector (0, 1), a cost value 3 corresponding to the edge motion vector (0, -1), a cost value 4 corresponding to the edge motion vector (1, 0), a cost value 5 corresponding to the edge motion vector (-1, 0), and a cost value 6 corresponding to the edge motion vector (1, 1) can be determined.

[136] At step a4: a motion vector is selected from the center motion vector and the edge motion vector as an optimal motion vector according to the first cost value and the second cost value. For example, the motion vector having the smallest cost value can be used as the

optimal motion vector. For example, assuming that the cost value 2 corresponding to the edge motion vector (0, 1) is minimum, the edge motion vector (0, 1) corresponding to the cost value 2 can be used as the optimal motion vector.

[137] At step a5: it is judged whether an end condition is met; if the end condition is not met, the optimal motion vector is determined as the center motion vector, and step a2 is returned to and executed; if the end condition is met, step a6 is executed.

[138] In an example, if the number of iteration time/search range reaches a threshold, an end condition is met; if the iteration time/search range does not reach the threshold, the end condition is not met. For example, assuming that SR is 2, i.e., the threshold is 2, two times of iterations are allowed. If the iteration time/search range has reached 2 times, i.e. step a2 to step a4 have been executed twice, the end condition is met; otherwise, the end condition is not met.

[139] In a further example, after selecting a motion vector from the center motion vector and the edge motion vectors as an optimal motion vector, if the center motion vector is selected as the optimal motion vector, the end condition is met.

[140] At step a6, a first integer-pixel motion vector adjustment value (which is used for adjusting the first original motion vector) and a second integer-pixel motion vector adjustment value (which is used for adjusting the second original motion vector) are determined according to the optimal motion vector.

[141] In an example, the first integer-pixel motion vector adjustment value can be determined according to the optimal motion vector and the first original motion vector, and the second integer-pixel motion vector adjustment value can be determined according to the first integer-pixel motion vector adjustment value, wherein the second integer-pixel motion vector adjustment value is symmetric to the first integer-pixel motion vector adjustment value.

[142] For example, in the first iterative process, the optimal motion vector is the edge motion vector  $(0, 1)$ , the second iteration is performed with the edge motion vector  $(0, 1)$  as the center, and in the second iterative process, the optimal motion vector is the edge motion vector  $(0, 1)$ ; assuming that the iterative process is completed hereto, the first integer-pixel motion vector adjustment value is  $(0, 2)$ , that is, the sum of the edge motion vector  $(0, 1)$  and the edge motion vector  $(0, 1)$ .

[143] Based on the above, assuming that the first original motion vector is  $(4, 4)$ , the optimal motion vector is the edge motion vector  $(0, 1)$  in the first iterative process, i.e. the optimal motion vector can correspond to the optimal motion vector  $(4, 5)$ . The second iterative process is performed by taking the edge motion vector  $(0, 1)$  as a center, wherein the optimal motion vector is the edge motion vector  $(0, 1)$  in the second iterative process, namely the optimal motion vector can correspond to the optimal motion vector  $(4, 6)$ .

[144] In conclusion, a first integer-pixel motion vector adjustment value is determined according to the optimal motion vector  $(4, 6)$  and the first original motion vector  $(4, 4)$ , the first integer-pixel motion vector adjustment value is the difference between the optimal motion vector  $(4, 6)$  and the first original motion vector  $(4, 4)$ , i.e. the first integer-pixel motion vector adjustment value is  $(0, 2)$ .

[145] Then a second integer-pixel motion vector adjustment value is determined according to the first integer-pixel motion vector adjustment value  $(0, 2)$ , which can be  $(0, -2)$ , i.e. a symmetric value of  $(0, 2)$ .

[146] At step a7, a first sub-pixel motion vector adjustment value (which is used for adjusting the first original motion vector) and a second sub-pixel motion vector adjustment value (which is used for adjusting the second original motion vector) are determined according to the optimal motion vector.

[147] In an example, a first sub-pixel motion vector adjustment value can be

determined according to a cost value corresponding to the optimal motion vector and a cost value corresponding to the edge motion vector corresponding to the optimal motion vector, and then a second sub-pixel motion vector adjustment value can be determined according to the first sub-pixel motion vector adjustment value.

[148] For example,  $x_0 = N * (E(-1,0) - E(1,0)) / (E(-1,0) + E(1,0) - 2 * E(0,0))$ ,  $y_0 = N * (E(0,-1) - E(0,1)) / (E(0,-1) + E(0,1) - 2 * E(0,0))$ , and when the motion vector pixel precision is 1/2, 1/4, 1/8, and 1/16,  $N = 1, 2, 4$ , and 8 respectively. Then,  $(x_0, y_0)$  is assigned to  $\Delta MV$ ,  $SPMV = \Delta MV / 2N$ , and if the current motion vector pixel precision is 1/16, the SPMV is  $(x_0/16, y_0/16)$ .

[149] In the above equation, SPMV is the first sub-pixel motion vector adjustment value, and  $N$  is related to motion vector pixel precision, e.g., if the motion vector pixel precision is 1/2,  $N$  is 1; if the motion vector pixel precision is 1/4,  $N$  is 2; if the motion vector pixel precision is 1/8,  $N$  is 4; if the motion vector pixel precision is 1/16,  $N$  is 8.

[150] In the above equation,  $E(0,0)$  represents a cost value of the optimal motion vector;  $E(-1,0)$  is a cost value of the edge motion vector  $(-1,0)$  of an optimal motion vector  $(0,0)$  with the optimal motion vector taken as the center;  $E(1,0)$  is a cost value of the edge motion vector  $(1,0)$  of the optimal motion vector  $(0,0)$  with the optimal motion vector taken as the center;  $E(0, -1)$  is a cost value of an edge motion vector  $(0, -1)$  of the optimal motion vector  $(0,0)$  with the optimal motion vector taken as the center;  $E(0,1)$  is a cost value of the edge motion vector  $(0,1)$  of the optimal motion vector  $(0,0)$  with the optimal motion vector taken as the center. For the cost value of each motion vector, the method of determining the cost value of each motion vector can refer to the above embodiments, which is not described herein in detail.

[151] After determining the first sub-pixel motion vector adjustment value in the above method, the second sub-pixel motion vector adjustment value can be determined according

to the first sub-pixel motion vector adjustment value, where the second sub-pixel motion vector adjustment value is a symmetric value of the first sub-pixel motion vector adjustment value. For example, if the first sub-pixel motion vector adjustment value is (1, 0), the second sub-pixel motion vector adjustment value is (-1, 0), i.e., a symmetric value of (1, 0).

[152] At step a8, the first original motion vector is adjusted according to the first integer-pixel motion vector adjustment value and/or the first sub-pixel motion vector adjustment value to obtain a first target motion vector.

[153] For example, the first target motion vector = the first original motion vector + the first integer-pixel motion vector adjustment value + the first sub-pixel motion vector adjustment value. Of course, the above is merely an example and is not limited thereto.

[154] At step a9, the second original motion vector is adjusted according to the second integer-pixel motion vector adjustment value and/or the second sub-pixel motion vector adjustment value to obtain a second target motion vector.

[155] For example, the second target motion vector = the second original motion vector + the second integer-pixel motion vector adjustment value + the second sub-pixel motion vector adjustment value. Of course, the above is merely an example and is not limited thereto.

[156] Embodiment 12: at step 303 and step 403, for each sub-block of the current block, the first original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the second original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[157] The adjustment process of the original motion vector is described by the

processing procedure of one sub-block (e.g. each  $dx*dy$  sized sub-block of the current block, e.g. a  $16*16$  sized sub-block, or a smaller block), for example. The first original motion vector is denoted as  $Org\_MV0$ , the second original motion vector is denoted as  $Org\_MV1$ , the first target motion vector obtained after the adjustment of the first original motion vector  $Org\_MV0$  is denoted as  $Refined\_MV0$ , and the second target motion vector obtained after the adjustment of the second original motion vector  $Org\_MV1$  is denoted as  $Refined\_MV1$ .

[158] At step b1, SR times of iterations are performed to obtain the optimal integer-pixel offset of the integer-pixel MV point, which is denoted as  $IntegerDeltaMV$ , which is the first integer-pixel motion vector adjustment value in the above embodiments. For example,  $IntegerDeltaMV$  is initialized to  $(0, 0)$ , and the subsequent processing for each iteration is performed as:

[159] At step b11,  $deltaMV$  is set to  $(0, 0)$ . If the first iteration is performed, a prediction value block A1 (i.e., the  $W*H$  block in the center of the first reference block) is copied based on the reference pixel of the first original motion vector in the first reference block; a prediction value block B1 (i.e. the  $W*H$  block in the center of the second reference block) is copied based on the reference pixel of the second original motion vector in the second reference block. An initial cost value  $cost$  (initial cost value is SAD (sum of absolute differences) based on the prediction value block A1 and the prediction value block B1) is obtained based on the prediction value block A1 and the prediction value block B1, and the method of determining the initial cost value is described in the following embodiments. If the initial cost value  $cost$  is less than  $4*dx*dy/2$ , and  $dx$  and  $dy$  are the width and height of the current sub-block, the subsequent search process is directly skipped, step b2 is executed, and  $notZeroCost$  is set as false.

[160] Step b12, as shown in Fig. 6, five offset MVs (which are all referred to as

MVOffset) are sequentially obtained according to  $\{Mv(0,1), Mv(0,-1), Mv(1,0), Mv(-1,0), Mv(right, down)\}$ , with the above initial point taken as the center, and the calculation and comparison process of the cost values of these five offset MVs is performed.

[161] For example, based on a certain MVOffset (e.g.,  $Mv(0, 1)$ , etc.), in the first and second reference blocks, two prediction value blocks (e.g., a  $W \times H$  block with center position offset MVOffset in the first reference block, and a  $W \times H$  block with center position offset -MVOffset (opposite to MVOffset) in the second reference block) are obtained from this MVOffset, and the down-sampling SAD of the two prediction value blocks are calculated as the cost value of MVOffset.

[162] Then, the MVOffset with the minimum cost value is reserved, this MVOffset is updated to the value of deltaMV, and is used as a new center offset point of the next iteration.

[163] The value of IntegerDeltaMV is updated based on the deltaMV, wherein the updated  $IntegerDeltaMV = IntegerDeltaMV \text{ before updating} + deltaMV$ , i.e., adding deltaMV on the basis of the current IntegerDeltaMV.

[164] At step b13, after iteration, if the optimal MV is still the initial MV (i.e. not MVOffset) or the minimum cost value is 0, the next iterative search process is not performed, step b2 is executed, and notZeroCost is set as false. Otherwise, if the iteration times reach SR, step b2 is executed, and if the iteration times do not reach SR, the optimal MV is taken as the center, and the next iterative search process is performed, that is, step b11 is returned to.

[165] After the iterative search process is finished, the value of IntegerDeltaMV is obtained, namely the final value of IntegerDeltaMV, namely the first integer-pixel motion vector adjustment value, which is subsequently denoted as IntegerDeltaMV.

[166] At step b2, the optimal integer-pixel MV point in step b1 is taken as the center,

then the optimal sub-pixel offset MV is obtained, which is denoted as SPMV, and the SPMV is the first sub-pixel motion vector adjustment value in the above embodiments. For example, the SPMV can be initialized to (0, 0) and then the subsequent processing is performed as follows.

[167] At step b21, subsequent processing (namely obtaining the SPMV) is only performed when notZeroCost is not false and deltaMV is (0, 0), otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV instead of using IntegerDeltaMV and SPMV.

[168] At step b22,  $E(x, y)$  is represented as the cost value corresponding to the MV of the optimal MV point offset (x, y) obtained in step b1 (the cost value calculated in step b1). Based on the  $E(x, y)$  of the center, top, bottom, left and right five points, the offset  $(x_0, y_0)$  of the point with the smallest  $E(x, y)$  can be obtained as follows:  $x_0 = N * (E(-1, 0) - E(1, 0)) / (E(-1, 0) + E(1, 0) - 2 * E(0, 0))$ ,  $y_0 = N * (E(0, -1) - E(0, 1)) / (E(0, -1) + E(0, 1) - 2 * E(0, 0))$ .

[169] In an example, when the motion vector pixel precision is 1/2, 1/4, 1/8, and 1/16,  $N = 1, 2, 4$ , and  $8$  respectively. Then,  $(x_0, y_0)$  is assigned to  $\text{deltaMv}$ ,  $\text{SPMV} = \text{deltaMv} / 2N$ , and if the current motion vector pixel precision is 1/16, the SPMV is  $(x_0/16, y_0/16)$ .

[170] If  $E(-1, 0) = E(0, 0)$ , a shift is performed by half a pixel horizontally to the left ( $\text{deltaMv}[0] = -N$ ).

[171] If  $E(1, 0) = E(0, 0)$ , a shift is performed by half a pixel horizontally to the right ( $\text{deltaMv}[0] = N$ ).

[172] If  $E(0, -1) = E(0, 0)$ , a shift is performed by half a pixel vertically upward ( $\text{deltaMv}[1] = -N$ ).

[173] If  $E(0, 1) = E(0, 0)$ , a shift is performed by half a pixel vertically downward ( $\text{deltaMv}[1] = N$ ).

[174] Based on the above processing, the value of the SPMV, i.e., the first sub-pixel motion vector adjustment value, can be obtained.

[175] At step b3, the optimal offset MV is obtained based on the integer-pixel offset IntegerDeltaMV in step b1 and the sub-pixel offset SPMV in step b2, and the optimal offset MV is denoted as BestMVoffset. Also,  $\text{BestMVoffset} = \text{IntegerDeltaMV} + \text{SPMV}$ .

[176] Based on BestMVoffset, the target motion vectors in two directions can be obtained:  
$$\text{Refined\_MV0} = \text{Org\_MV0} + \text{BestMVoffset};$$
$$\text{Refined\_MV1} = \text{Org\_MV1} - \text{BestMVoffset}.$$

[177] Notably,  $\text{BestMVoffset} = \text{IntegerDeltaMV} + \text{SPMV}$ , i.e. the sum of the first integer-pixel motion vector adjustment value and the first sub-pixel motion vector adjustment value. Furthermore,  $-\text{IntegerDeltaMV}$  is the symmetric value of IntegerDeltaMV, i.e. the second integer-pixel motion vector adjustment value,  $-\text{SPMV}$  is the symmetric value of SPMV, i.e. the second sub-pixel motion vector adjustment value, so that  $-\text{BestMVoffset} = (-\text{IntegerDeltaMV}) + (-\text{SPMV})$ , i.e. the sum of the second integer-pixel motion vector adjustment value and the second sub-pixel motion vector adjustment value.

[178] Embodiment 13: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[179] "if the initial cost value cost is less than  $4 * dx * dy / 2$ , ..., the subsequent search process is directly skipped " in step b11 is removed, that is, even if the initial cost value cost is less than  $4 * dx * dy / 2$ , it will not "skip the subsequent search process directly", but continue the subsequent search process, that is, step b12 needs to be executed.

[180] Embodiment 14: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first

and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[181] "if the initial cost value cost is less than  $4*dx*dy/2$ , then directly skip the subsequent search process" in step b11 is removed, that is, even if the initial cost value cost is less than  $4*dx*dy/2$ , it will not "skip the subsequent search process directly", but continue the subsequent search process, that is, step b12 needs to be executed.

[182] "if the optimal MV is still the initial MV (i.e. not MVOffset) or the minimum cost value is 0, the next iterative search process is not performed " in step b13 is removed, that is, even if the optimal MV is still the initial MV or the minimum cost value is 0, the next iterative search process can be performed.

[183] Embodiment 15: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[184] the correlation process of "notZeroCost" is removed, that is, the value of notZeroCost is not set and saved in step b11 and step b13. In step b21, the sub-pixel offset calculation process (i.e., step b22) can be performed when the deltaMV is (0, 0), instead of only when notZeroCost is not false and deltaMV is (0, 0).

[185] Embodiment 16: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[186] "subsequent processing is only performed when notZeroCost is not false and deltaMV is (0, 0), otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV" in step b21 is modified to "subsequent processing is only

performed when notZeroCost is not false and the cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixel are obtained by calculation in the step b1, otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV”.

[187] In an example, "subsequent processing" refers to the sub-pixel offset calculation process in step b22.

[188] In an example, the sub-pixel offset calculation process in step b22 needs to use cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixel, so it is a necessary condition that "cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixe" have been calculated in step b1.

[189] Embodiment 17: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[190] “subsequent processing is only performed when notZeroCost is not false and deltaMV is (0, 0), otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV” in step b21 is modified to “subsequent processing (i.e., the sub-pixel offset calculation process) is only performed when cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixel are obtained by step b1, otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV”.

[191] Embodiment 18: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first

and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiment 12, except that:

[192] “subsequent processing is only performed when notZeroCost is not false and deltaMV is (0, 0), otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV” in step b21 is modified to “subsequent processing (i.e., the sub-pixel offset calculation process in the step b22) is performed when cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixel are obtained by step b1, otherwise, step b23 is executed.

[193] At step b23, the current optimal integer-pixel point MV\_inter\_org is set to the nearest integer-pixel point MV\_inter\_nearest, the cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the nearest integer-pixel point have been obtained in step b1. Then, the sub-pixel offset calculation process of step b22 is performed centering on the MV\_inter\_nearest, that is, obtaining the SPMV centering on the MV\_inter\_nearest.

[194] For example, if the cost values of four points which are respectively at the top, bottom, left and right of and have a distance of one pixel from the current optimal integer-pixel point are not all obtained by step b1, one integer-pixel point MV\_inter\_nearest is selected from the periphery of the optimal integer-pixel point MV\_inter\_org, and the cost values of four points which are respectively at the top, bottom, left and right of the integer-pixel point MV\_inter\_nearest are obtained by step b1.

[195] Then, the integer-pixel point MV\_inter\_nearest is set as the current optimal integer-pixel point, and the SPMV is obtained centering on the integer-pixel point MV\_inter\_nearest, and the specific process refers to step b22.

[196] When obtaining the SPMV centering on the integer-pixel point MV\_inter\_nearest,

referring to step b22,  $x_0$  and  $y_0$  can be limited to the range of  $[-2N, 2N]$  when calculating  $x_0$  and  $y_0$ . If  $x_0/y_0$  is greater than  $2N$ ,  $x_0/y_0$  can be assigned with  $2N$ ; if  $x_0/y_0$  is less than  $-2N$ ,  $x_0/y_0$  can be assigned with  $-2N$ . When the motion vector pixel precision is  $1/2$ ,  $1/4$ ,  $1/8$ , and  $1/16$ ,  $N=1, 2, 4$ , and  $8$  respectively.

[197] Embodiment 19: in the above embodiments, it is necessary to determine edge motion vectors corresponding to the center motion vector, such as 5 edge motion vectors. For example, the center motion vector  $(x, y)$  can be shifted by  $S$  in different directions, and edge motion vector  $(x, y+S)$ , edge motion vector  $(x, y-S)$ , edge motion vector  $(x+S, y)$ , edge motion vector  $(x-S, y)$ , and edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions can be obtained sequentially. Alternatively, the center motion vector  $(x, y)$  can be shifted by  $S$  in different directions, and edge motion vector  $(x, y-S)$ , edge motion vector  $(x, y+S)$ , edge motion vector  $(x-S, y)$ , edge motion vector  $(x+S, y)$ , and edge motion vector  $(x + \text{right}, y + \text{down})$  in different directions can be obtained sequentially.

[198] For example, assuming that  $(x, y)$  is  $(0, 0)$  and  $S$  is 1, then 5 edge motion vectors can be obtained sequentially according to  $(0, 1)$ ,  $(0, -1)$ ,  $(1, 0)$ ,  $(-1, 0)$ , and  $(\text{right}, \text{down})$ . Alternatively, 5 edge motion vectors can be obtained sequentially according to  $(0, -1)$ ,  $(0, 1)$ ,  $(-1, 0)$ ,  $(1, 0)$ , and  $(\text{right}, \text{down})$ .

[199] Embodiment 20: in the above embodiments, the default value of the edge motion vector  $(x+\text{right}, y+\text{down})$  is  $(x-S, y-S)$ . If the cost value of the edge motion vector  $(x+S, y)$  is smaller than that of the edge motion vector  $(x-S, y)$ , right is  $S$  ( $-S$  is modified to  $S$ ); if the cost value of the edge motion vector  $(x, y+S)$  is smaller than that of the edge motion vector  $(x, y-S)$ , down is  $S$  ( $-S$  is modified to  $S$ ). Alternatively, if the cost value of the edge motion vector  $(x+S, y)$  is less than or equal to the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$  ( $-S$  is modified to  $S$ ); if the cost value of the edge motion vector  $(x, y+S)$  is less than or equal to the cost value of the edge motion vector  $(x, y-S)$ , down is  $S$  ( $-S$  is modified

to S).

[200] For example, 5 edge motion vectors are obtained sequentially according to (0, 1), (0, -1), (1, 0), (-1, 0), (right, down), and the default value of (right, down) is (-1, -1). If the cost value of the edge motion vector (1, 0) is less than the cost value of the edge motion vector (-1, 0), right is 1; if the cost value of the edge motion vector (0, 1) is smaller than the cost value of the edge motion vector (0, -1), down is 1. Alternatively, if the cost value of the edge motion vector (1, 0) is less than or equal to the cost value of the edge motion vector (-1, 0), right is 1; if the cost value of the edge motion vector (0, 1) is less than or equal to the cost value of the edge motion vector (0, -1), down is 1.

[201] For example, 5 edge motion vectors are obtained sequentially according to (0, -1), (0, 1), (-1, 0), (1, 0), (right, down), and the default value of (right, down) is (-1, -1). If the cost value of the edge motion vector (1, 0) is less than the cost value of the edge motion vector (-1, 0), right is 1; if the cost value of the edge motion vector (0, 1) is smaller than the cost value of the edge motion vector (0, -1), down is 1. Alternatively, if the cost value of the edge motion vector (1, 0) is less than or equal to the cost value of the edge motion vector (-1, 0), right is 1; if the cost value of the edge motion vector (0, 1) is less than or equal to the cost value of the edge motion vector (0, -1), down is 1.

[202] Embodiment 21: in steps 303 and 403, for each sub-block of the current block, the first original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the second original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[203] The adjustment process of the original motion vector is described by the processing procedure of one sub-block (e.g. each  $dx*dy$  sized sub-block of the current

block, e.g. a 16\*16 sized sub-block, or a smaller block), for example.

[204] At step c1, the first original motion vector or the second original motion vector is taken as a center, some or all of the motion vectors are selected from surrounding motion vectors, and the selected motion vectors are determined as candidate motion vectors.

[205] For example, some or all of the motion vectors can be selected from the surrounding motion vectors as candidate motion vectors, with the first original motion vector taken as the center. Refer to the following embodiments for details.

[206] In the following embodiments, the first original motion vector is taken as the center motion vector for description.

[207] At step c2, a third cost value corresponding to the first original motion vector and a fourth cost value corresponding to each candidate motion vector are acquired according to the first pixel value of the first reference block and the second pixel value of the second reference block.

[208] For example, a sub-reference block A1 corresponding to the first original motion vector is obtained by copying from the first reference block, the sub-reference block A1 is a sub-reference block of the first original motion vector in the first reference block, and the size of the sub-reference block A1 is the size of the current block. A sub-reference block B1 corresponding to the second original motion vector is obtained by copying from the second reference block, the sub-reference block B1 is a sub-reference block of the second original motion vector in the second reference block, and the size of the sub-reference block B1 is the size of the current block. Then, a third cost value corresponding to the first original motion vector is obtained according to the first pixel value of the sub-reference block A1 and the second pixel value of the sub-reference block B1.

[209] For each candidate motion vector, a sub-reference block A2 corresponding to the candidate motion vector is obtained by copying from the first reference block, the

sub-reference block A2 is a sub-reference block of the candidate motion vector in the first reference block, and the size of the sub-reference block A2 is the size of the current block. A sub-reference block B2 corresponding to the motion vector symmetrical to the candidate motion vector is obtained by copying from the second reference block, and the size of the sub-reference block B2 is the size of the current block. And acquiring a fourth cost value corresponding to the candidate motion vector according to the first pixel value of the sub-reference block sA2 and the second pixel value of the sub-reference block B2.

[210] At step c3: a motion vector is selected from the first original motion vector and the candidate motion vectors as an optimal motion vector according to the third cost value and the fourth cost value. For example, the motion vector with the smallest cost value (e.g., the first original motion vector, or any one of the candidate motion vectors) is used as the optimal motion vector.

[211] At step c4, a first integer-pixel motion vector adjustment value (which is used for adjusting the first original motion vector) and a second integer-pixel motion vector adjustment value (which is used for adjusting the second original motion vector) are determined according to the optimal motion vector. In an example, the first integer-pixel motion vector adjustment value can be determined according to the optimal motion vector and the first original motion vector, and the second integer-pixel motion vector adjustment value can be determined according to the first integer-pixel motion vector adjustment value, wherein the second integer-pixel motion vector adjustment value is symmetric to the first integer-pixel motion vector adjustment value.

[212] For example, assuming that the optimal motion vector is (4, 6) and the first original motion vector is (4, 4), the first integer-pixel motion vector adjustment value is determined from the optimal motion vector (4, 6) and the first original motion vector (4, 4), the first integer-pixel motion vector adjustment value is the difference between the optimal

motion vector (4, 6) and the first original motion vector (4, 4), i.e. the first integer-pixel motion vector adjustment value is (0, 2).

[213] Then a second integer-pixel motion vector adjustment value is determined according to the first integer-pixel motion vector adjustment value (0, 2), the second integer-pixel motion vector adjustment value can be (0, -2), i.e. a symmetric value of (0, 2).

[214] At step c5, a first sub-pixel motion vector adjustment value (which is used for adjusting the first original motion vector) and a second sub-pixel motion vector adjustment value (which is used for adjusting the second original motion vector) are determined according to the optimal motion vector.

[215] In an example, a first sub-pixel motion vector adjustment value can be determined according to a cost value corresponding to the optimal motion vector and a cost value corresponding to the edge motion vector corresponding to the optimal motion vector, and then a second sub-pixel motion vector adjustment value can be determined according to the first sub-pixel motion vector adjustment value.

[216] For example,  $x_0 = N * (E(-1,0) - E(1,0)) / (E(-1,0) + E(1,0) - 2 * E(0,0))$ ,  $y_0 = N * (E(0,-1) - E(0,1)) / (E(0,-1) + E(0,1) - 2 * E(0,0))$ , and when the motion vector pixel precision is 1/2, 1/4, 1/8, and 1/16,  $N = 1, 2, 4$ , and 8 respectively. Then,  $(x_0, y_0)$  is assigned to  $\Delta Mv$ ,  $SPMV = \Delta Mv / 2N$ , and if the current motion vector pixel precision is 1/16, the SPMV is  $(x_0/16, y_0/16)$ .

[217] SPMV is the first sub-pixel motion vector adjustment value.  $E(0,0)$  represents a cost value of the optimal motion vector;  $E(-1,0)$  represents a cost value of the edge motion vector (-1,0) of an optimal motion vector (0,0) with the optimal motion vector taken as the center;  $E(1,0)$  represents a cost value of the edge motion vector (1,0) of the optimal motion vector (0,0) with the optimal motion vector taken as the center;  $E(0, -1)$  represents a cost value of an edge motion vector (0, -1) of the optimal motion vector (0,0) with the optimal

motion vector taken as the center;  $E(0,1)$  represents a cost value of the edge motion vector  $(0,1)$  of the optimal motion vector  $(0,0)$  with the optimal motion vector taken as the center. For the cost value of each motion vector, the method of determining the cost value of each motion vector can refer to the above embodiments.

[218] After determining the first sub-pixel motion vector adjustment value in the above method, the second sub-pixel motion vector adjustment value can be determined according to the first sub-pixel motion vector adjustment value, where the second sub-pixel motion vector adjustment value is a symmetric value of the first sub-pixel motion vector adjustment value. For example, if the first sub-pixel motion vector adjustment value is  $(1, 0)$ , the second sub-pixel motion vector adjustment value is  $(-1, 0)$ , i.e., a symmetric value of  $(1, 0)$ .

[219] At step c6, the first original motion vector is adjusted according to the first integer-pixel motion vector adjustment value and/or the first sub-pixel motion vector adjustment value to obtain a first target motion vector.

[220] For example, the first target motion vector = first original motion vector + first integer-pixel motion vector adjustment value + first sub-pixel motion vector adjustment value. Of course, the above is merely an example and is not limited thereto.

[221] At step c7, the second original motion vector is adjusted according to the second integer-pixel motion vector adjustment value and/or the second sub-pixel motion vector adjustment value to obtain a second target motion vector.

[222] For example, the second target motion vector = the second original motion vector + the second integer-pixel motion vector adjustment value + the second sub-pixel motion vector adjustment value. Of course, the above is merely an example and is not limited thereto.

[223] Embodiment 22: in step 303 and step 403, for each sub-block of the current block,

the first original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a first target motion vector of the sub-block; and the second original motion vector is adjusted according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain a second target motion vector of the sub-block.

[224] The adjustment process of the original motion vector is described by the processing procedure of one sub-block (e.g. each  $dx*dy$  sized sub-block of the current block, e.g. a  $16*16$  sized sub-block, or a smaller block), for example. The first original motion vector is denoted as Org\_MV0, the second original motion vector is denoted as Org\_MV1, the first target motion vector is denoted as Refined\_MV0, and the second target motion vector is denoted as Refined\_MV1.

[225] At step d1, some or all of the motion vectors are selected from  $(2*SR+1)*(2*SR+1)$  points around the first original motion vector with the first original motion vector taken as the center. For example, if  $SR=2$ , some or all of the motion vectors are selected from 25 points around the first original motion vector as candidate motion vectors. Then, a cost value of the first original motion vector is determined, and a cost value of each candidate motion vector is determined. Then, the motion vector with the minimum cost value is used as the optimal motion vector.

[226] Compared with step b1 of the above embodiment, in step d1, all candidate motion vectors to be processed can be selected at one time without performing an iterative process, instead of selecting some motion vectors in the first iteration and selecting the other motion vectors in the second iteration through an iterative process. Based on this, all of the candidate motion vectors to be processed are selected at one time, so that the candidate motion vectors can be processed in parallel to obtain the cost value of each candidate motion vector, thereby reducing the computational complexity and improving the encoding

performance.

[227] At step d2, the value of IntegerDeltaMV is determined according to the optimal motion vector, wherein the final value of IntegerDeltaMV is the first integer-pixel motion vector adjustment value, and the determination method is not repeated here.

[228] At step d3: the optimal motion vector is taken as the center, the optimal sub-pixel offset MV is obtained, which is denoted as SPMV, and the SPMV is the first sub-pixel motion vector adjustment value.

[229] The implementation process of step d3 can be referred to the above step b2, and will not be described repeatedly herein.

[230] At step d4, BestMVoffset is obtained based on IntegerDeltaMV and SPMV. For example,  $\text{BestMVoffset} = \text{IntegerDeltaMV} + \text{SPMV}$ .

[231] The target motion vector is obtained based on BestMVoffset:  $\text{Refined\_MV0} = \text{Org\_MV0} + \text{BestMVoffset}$ ;  $\text{Refined\_MV1} = \text{Org\_MV1} - \text{BestMVoffset}$ .

[232] Embodiment 23: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiments 21, 22.

[233] In this embodiment, all of the motion vectors are selected from  $(2 \times \text{SR} + 1) \times (2 \times \text{SR} + 1)$  points around the original motion vector with the original motion vector taken as the center. For example, if  $\text{SR} = 2$ , all of the motion vectors are selected from 25 points around the original motion vector, the cost values of these motion vectors are determined, and the cost value of each motion vector is determined. Then, the motion vector with the minimum cost value is used as the optimal motion vector.

[234] Embodiment 24: with respect to step 303 and step 403, the implementation of

adjusting the first and second original motion vectors Org\_MV0 and Org\_MV1 to the first and second target motion vectors Refined\_MV0 and Refined\_MV1 respectively is similar to that of embodiments 21, 22. In this embodiment, all of the candidate motion vectors to be processed are selected at one time, so that the candidate motion vectors can be processed in parallel to obtain the cost value of each candidate motion vector, thereby reducing the computational complexity and improving the encoding performance.

[235] In this embodiment, some of motion vectors having an offset not exceeding SR is selected from  $(2*SR+1)*(2*SR+1)$  points around the original motion vector with the original motion vector taken as the center. For example, N (N is equal to or greater than 1, and equal to or less than  $(2*SR+1)*(2*SR+1)$ ) candidate points are selected from  $(2*SR+1)*(2*SR+1)$  points including the original motion vector. And the cost values of the motion vectors corresponding to the N points are determined. The cost values of the N points are scanned in a certain sequence, and the motion vector with the minimum cost value is selected as the optimal motion vector. If the cost values are equal, the candidate points with the highest sequence are preferentially selected.

[236] In an example, assuming  $SR = 2$ , the number of candidate points can be 25, and the sequence of the candidate points can be from left to right and from top to bottom. Referring to Fig. 7A, the sequence of the candidate points can be: { Mv(-2, -2), Mv(-1, -2), Mv(0, -2), Mv(1, -2), Mv(2, -2), Mv(-2, -1), Mv(-1, -1), Mv(0, -1), Mv(1, -1), Mv(2, -1), Mv(-2, 0), Mv(-1, 0), Mv(0, 0), Mv(1, 0), Mv(2, 0), Mv(-2, 1), Mv(-1, 1), Mv(0, 1), Mv(1, 1), Mv(2, 1), Mv(-2, 2), Mv(-1, 2), Mv(0, 2), Mv(1, 2), Mv(2, 2) }.

[237] The cost values corresponding to the motion vectors of the 25 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset MV, the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be

determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[238] In another example, assuming  $SR = 2$ , the number of candidate points can be 21, and the sequence of the candidate points can be from left to right and from top to bottom. Referring to Fig. 7B, the sequence of the candidate points can be:  $\{Mv(-1,-2), Mv(0,-2), Mv(1,-2), Mv(-2,-1), Mv(-1,-1), Mv(0,-1), Mv(1, -1), Mv(2, -1), Mv(-2, 0), Mv(-1, 0), Mv(0, 0), Mv(1, 0), Mv(2, 0), Mv(-2, 1), Mv(-1, 1), Mv(0, 1), Mv(1, 1), Mv(2, 1), Mv(-1, 2), Mv(0, 2), Mv(1, 2)\}$ . The cost values corresponding to the motion vectors of the 21 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset MV, the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[239] In another example, assuming  $SR = 2$ , the number of candidate points can be 25, and the sequence of these candidate points is the sequence from near to far away from the center, with the motion vector  $(0, 0)$  taken as the center. Referring to Fig. 7C, the sequence of the candidate points can be:  $\{Mv(0, 0), Mv(-1, 0), Mv(0,-1), Mv(1, 0), Mv(0, 1), Mv(-1, 1), Mv(-1,-1), Mv(1,-1), Mv(1, 1), Mv(0, 2), Mv(-2, 0), Mv(0,-2), Mv(2, 0), Mv(1, 2), Mv(-1, 2), Mv(-2, 1), Mv(-2,-1), Mv(-1,-2), Mv(1,-2), Mv(2,-1), Mv(2, 1), Mv(-2, 2), Mv(-2,-2), Mv(2,-2), Mv(2, 2)\}$ . The cost values corresponding to the motion vectors of the 25 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset MV, the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[240] In another example, assuming  $SR = 2$ , the number of candidate points can be 21, and the sequence of these candidate points is from near to far away from the center, with the motion vector  $(0, 0)$  taken as the center. Referring to Fig. 7D, the sequence of the candidate points is: {  $Mv(0,0)$ ,  $Mv(-1,0)$ ,  $Mv(0, -1)$ ,  $Mv(1,0)$ ,  $Mv(0,1)$ ,  $Mv(-1,1)$ ,  $Mv(-1, -1)$ ,  $Mv(1, -1)$ ,  $Mv(1,1)$ ,  $Mv(0,2)$ ,  $Mv(-2,0)$ ,  $Mv(0, -2)$ ,  $Mv(2,0)$ ,  $Mv(1, 2)$ ,  $Mv(-1,2)$ ,  $Mv(-2,1)$ ,  $Mv(-2, -1)$ ,  $Mv(-1, -2)$ ,  $Mv(1, -2)$ ,  $Mv(2, -1)$ ,  $Mv(2, 1)$  }.

[241] The cost values corresponding to the motion vectors of the 21 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset MV, the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[242] In another example, assuming  $SR = 2$ , the number of candidate points can be 13, and the sequence of these candidate points is from near to far away from the center, with the motion vector  $(0, 0)$  taken as the center. Referring to Fig. 7E, the sequence of the candidate points is: { $Mv(0,0)$ ,  $Mv(-1,0)$ ,  $Mv(0,-1)$ ,  $Mv(1,0)$ ,  $Mv(0,1)$ ,  $Mv(-1,1)$ ,  $Mv(-1,-1)$ ,  $Mv(1,-1)$ ,  $Mv(1, 1)$ ,  $Mv(0, 2)$ ,  $Mv(-2, 0)$ ,  $Mv(0,-2)$ ,  $Mv(2, 0)$ }. The cost values corresponding to the motion vectors of the 13 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset MV, the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[243] Embodiment 25: with respect to step 303 and step 403, the implementation of adjusting the first and second original motion vectors  $Org\_MV0$  and  $Org\_MV1$  to the first and second target motion vectors  $Refined\_MV0$  and  $Refined\_MV1$  respectively is

similar to that of embodiments 21, 22. In this embodiment, all of the candidate motion vectors to be processed are selected at one time, so that the candidate motion vectors can be processed in parallel to obtain the cost value of each candidate motion vector, thereby reducing the computational complexity and improving the encoding performance.

[244] In this embodiment, some of motion vectors having a offset not exceeding SR is selected from  $(2*SR+1)*(2*SR+1)$  points around the original motion vector with the original motion vector taken as the center. For example, N (N is equal to or greater than 1, and equal to or less than  $(2*SR+1)*(2*SR+1)$ ) candidate points are selected from  $(2*SR+1)*(2*SR+1)$  points including the original motion vector. The cost values of the motion vectors corresponding to the N points are determined. The cost values of the N points are scanned in a certain sequence, and the motion vector with the minimum cost value is selected as the optimal motion vector. If the cost values are equal, the candidate points with the highest sequence are preferentially selected.

[245] Embodiment 25 differs from embodiment 24 in that the positions of candidate points in embodiment 24 are all fixed, i.e., are independent of the original motion vector, while the positions of candidate points in embodiment 25 are related to the original motion vector, and the following description is made with reference to several specific examples.

[246] In an example, assuming  $SR = 2$ , the number of candidate points can be 13, and the sequence of these candidate points is from near to far away from the center, with the motion vector (0, 0) taken as the center. Furthermore, the sequence of a first layer candidate points from the center is independent of the size of the original motion vector, while the sequence of a second layer candidate points from the center is dependent on the size of the original motion vector. The sequence of the candidate points is: {Mv(0,0), Mv(-1,0), Mv(0,-1), Mv(1,0), Mv(0,1), Mv(-1,1), Mv(-1,-1), Mv(1,-1), Mv(1,1), Mv(sign\_H\*2,0),

$Mv(sign\_H*2,sign\_V*1)$ ,  $Mv(0, sign\_V *2)$ ,  $Mv(0, sign\_V *2)\}$ . The first original motion vector is denoted as  $MV0$ , horizontal component is  $MV0\_Hor$ , and vertical component is  $MV0\_Ver$ . If the  $MV0\_Hor$  is greater than or equal to 0,  $sign\_H = 1$ ; otherwise  $sign\_H = -1$ ; if the  $MV0\_Ver$  is greater than or equal to 0,  $sign\_V = 1$ ; otherwise  $sign\_V = -1$ .

[247] The cost values corresponding to the motion vectors of the 13 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset  $MV$ , the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be determined by using the optimal offset  $MV$ , and the determination method refers to the above embodiments.

[248] In another example, assuming  $SR = 2$ , the number of candidate points can be 13, and the sequence of these candidate points is from near to far away from the center, with the motion vector  $(0, 0)$  taken as the center. Furthermore, the sequence of a first layer candidate points from the center is independent of the size of the original motion vector, while the sequence of a second layer candidate points from the center is dependent on the size of the original motion vector. The sequence of the candidate points is:  $\{Mv(0,0)$ ,  $Mv(-1,0)$ ,  $Mv(0,-1)$ ,  $Mv(1,0)$ ,  $Mv(0,1)$ ,  $Mv(-1,1)$ ,  $Mv(-1,-1)$ ,  $Mv(1,-1)$ ,  $Mv(1,1)$ ,  $Mv(sign\_H*2,0)$ ,  $Mv(sign\_H*2,sign\_V*1)$ ,  $Mv(0, sign\_V *2)$ ,  $Mv(0, sign\_V *2)\}$ . The first original motion vector is denoted as  $MV0$ , horizontal component is  $MV0\_Hor$ , and vertical component is  $MV0\_Ver$ . If the  $MV0\_Hor$  is greater than 0,  $sign\_H = 1$ ; otherwise  $sign\_H = -1$ ; if the  $MV0\_Ver$  is greater than 0,  $sign\_V = 1$ ; otherwise  $sign\_V = -1$ .

[249] The cost values corresponding to the motion vectors of the 13 points are determined, scanning is performed according to the above sequence, the motion vector with the minimum cost value is obtained as the optimal offset  $MV$ , the integer-pixel motion vector adjustment value and the sub-pixel motion vector adjustment value can be

determined by using the optimal offset MV, and the determination method refers to the above embodiments.

[250] Embodiment 26: in the above embodiments, the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector are obtained according to the first pixel value of the first reference block and the second pixel value of the second reference block. The third cost value corresponding to the first original motion vector and the fourth cost value corresponding to the candidate motion vectors are obtained according to the first pixel value of the first reference block and the second pixel value of the second reference block.

[251] In an example, the first cost value corresponding to the center motion vector, the second cost value corresponding to the edge motion vector, the third cost value corresponding to the first original motion vector, and the fourth cost value corresponding to the candidate motion vector can be obtained according to a first pixel value without down-sampling and a second pixel value without down-sampling.

[252] In another example, down-sampling operation on the first and second pixel values is performed; then, according to the first and second pixel values after down-sampling, the first cost value corresponding to the center motion vector, the second cost value corresponding to the edge motion vector, the third cost value corresponding to the first original motion vector, and the fourth cost value corresponding to the candidate motion vector are obtained.

[253] In another example, down-sampling and shifting operation is performed on the first and second pixel values, then, according to the first and second pixel values after shifting and down-sampling, the first cost value corresponding to the center motion vector, the second cost value corresponding to the edge motion vector, the third cost value corresponding to the first original motion vector, and the fourth cost value corresponding to

the candidate motion vector are obtained.

[254] The method in which the cost value is determined is similar for different situations. For example, in order to obtain the cost value corresponding to the center motion vector, a sub-reference block A1 corresponding to the center motion vector can be obtained by copying from the first reference block, a sub-reference block B1 corresponding to a motion vector symmetrical to the center motion vector can be obtained by copying from the second reference block, and the cost value corresponding to the center motion vector can be obtained according to the first pixel value of the sub-reference block A1 and the second pixel value of the sub-reference block B1. In order to obtain the cost value corresponding to the edge motion vector, a sub-reference block A2 corresponding to the edge motion vector can be obtained by copying from the first reference block, a sub-reference block B2 corresponding to a motion vector symmetrical to the edge motion vector can be obtained by copying from the second reference block, and the cost value corresponding to the edge motion vector can be obtained according to the first pixel value of the sub-reference block A2 and the second pixel value of the sub-reference block B2, and the like.

[255] To sum up, in order to obtain the cost value corresponding to the motion vector, the sub-reference block corresponding to the motion vector is obtained from the first reference block, the sub-reference block corresponding to the symmetric motion vector of the motion vector is obtained from the second reference block, and then the cost value corresponding to the motion vector is obtained according to the pixel values of the two sub-reference blocks.

[256] Embodiment 27: on the basis of embodiment 26, a cost value corresponding to the motion vector is obtained according to the first pixel value without down-sampling (i.e., a pixel value without down-sampling for the sub-reference block in the first reference block) and the second pixel value without down-sampling (i.e., a pixel value without

down-sampling for the sub-reference block in the second reference block).

[257] In an example, assuming that a sub-reference block in the first reference block is  $pred_0$  and a sub-reference block in the second reference block is  $pred_1$ , the cost value is determined according to the SAD of all the pixel values of the sub-reference blocks  $pred_0$ ,  $pred_1$ , and it is not necessary to perform down-sampling vertically on the pixels of sub-reference blocks  $pred_0$ ,  $pred_1$ .

[258] Based on all the pixel values of the sub-reference blocks  $pred_0$ ,  $pred_1$ , the formula of calculating the cost value is:

$$cost = \sum_{i=1}^H \sum_j^W abs(pred_0(i,j) - pred_1(i,j))$$

[259] In the above equation, cost represents a cost value, W represents a width value of the sub-reference block, H represents a height value of the sub-reference block,  $pred_0(i,j)$  represents a pixel value of column i and row j of the sub-reference block  $pred_0$ ,  $pred_1(i,j)$  represents a pixel value of column i and row j of the sub-reference block  $pred_1$ , and  $abs(x)$  represents an absolute value of x.

[260] Embodiment 28: on the basis of embodiment 26, the down-sampling operation can be performed on the first and second pixel values; a cost value corresponding to the motion vector is obtained according to the first pixel value after down-sampling (i.e., a pixel value after down-sampling for the sub-reference block in the first reference block) and the second pixel value after down-sampling (i.e., a pixel value after down-sampling for the sub-reference block in the second reference block).

[261] In an example, assuming that a sub-reference block in the first reference block is  $pred_0$  and a sub-reference block in the second reference block is  $pred_1$ , the cost value is determined according to the SAD of all the pixel values of the sub-reference blocks  $pred_0$ ,  $pred_1$ . When the cost value is determined by using the SAD of all pixel values, the

down-sampling operation can be performed vertically on the pixel values of the sub-reference blocks  $pred_0, pred_1$  by N times (N is an integer greater than 0, and may be 2).

[262] Based on all the pixel values of the sub-reference blocks  $pred_0, pred_1$ , the formula of calculating the cost value is:

$$cost = \sum_{i=1}^{H/N} \sum_j^W abs(pred_0(1 + N(i - 1), j) - pred_1(1 + N(i - 1), j))$$

[263] In the above equation, cost represents a cost value, W represents a width value of the sub-reference block, H represents a height value of the sub-reference block; N represents a parameter of down-sampling, and is an integer greater than 0, which may be 2;  $pred_0(1 + N(i - 1), j)$  represents a pixel value of column  $1 + N(i - 1)$  and row j of the sub-reference block  $pred_0$ ,  $pred_1(1 + N(i - 1), j)$  represents a pixel value of column  $1 + N(i - 1)$  and row j of the sub-reference block  $pred_1$ , and  $abs(x)$  represents an absolute value of x.

[264] Embodiment 29: on the basis of the embodiment 26, down-sampling operation is performed on the first and second pixel values, then, according to the first and second pixel values (a pixel value after shifting and down-sampling for the sub-reference block in the first reference block, and a pixel value after shifting and down-sampling for the sub-reference block in the second reference block) after shifting and down-sampling, the cost values corresponding to the motion vectors are obtained.

[265] In an example, assuming that a sub-reference block in the first reference block is  $pred_0$ , a sub-reference block in the second reference block is  $pred_1$ , and both  $pred_0, pred_1$  are stored in the form of D bits, that is, each pixel value in  $pred_0$  is stored in D bits, and each pixel value in  $pred_1$  is stored in D bits.

[266] If D is equal to or less than 8, the cost value is determined according to the SAD of all the pixel values of the sub-reference blocks  $pred_0, pred_1$ . When the cost value is determined by using the SAD of all pixel values, the down-sampling operation can be

performed vertically on the pixel values of the sub-reference blocks  $pred_0, pred_1$  by N times (N is an integer greater than 0, and may be 2). Based on all the pixel values of the sub-reference blocks  $pred_0, pred_1$ , the cost value calculation equation is:

$$cost = \sum_{i=1}^{H/N} \sum_j^W abs(pred_0(1 + N(i - 1), j) - pred_1(1 + N(i - 1), j))$$

[267] In the above equation, cost represents a cost value, W represents a width value of the sub-reference block, H represents a height value of the sub-reference block; N represents a parameter of down-sampling, and is an integer greater than 0, which may be 2;  $pred_0(1+N(i-1), j)$  represents a pixel value of column  $1+N(i-1)$  and row j of the sub-reference block  $pred_0$ ,  $pred_1(1+N(i-1), j)$  represents a pixel value of column  $1+N(i-1)$  and row j of the sub-reference block  $pred_1$ , and  $abs(x)$  represents an absolute value of x. and it can be seen from the above that only the sum of absolute of the differences of row 1, N+1, 2N+1, ... , is calculated.

[268] If D is greater than 8, all pixel values of the sub-reference blocks  $pred_0, pred_1$  are first shifted to 8 bits to obtain 8-bit  $pred_0$  and 8-bit  $pred_1$ , denoted as  $pred_{0\_8bit(i, j)}$  and  $pred_{1\_8bit(i, j)}$ . The purpose is to save the storage cost of SAD calculation, and the storage of 8 bits can realize higher parallelism.

$$pred_{0\_8bit}(i, j) = pred_0(i, j) \gg (D - 8), \quad pred_{1\_8bit}(i, j) = pred_1(i, j) \gg (D - 8)$$

[269] Then, the down-sampling operation can be performed vertically on the pixel values of 8-bit  $pred_0$  and 8-bit  $pred_1$  by N times (N is an integer greater than 0, and may be 2), so that the cost value can be calculated as:

$$cost = \sum_{i=1}^{H/N} \sum_j^W abs(pred_{0\_8bit}(1 + N(i - 1), j) - pred_{1\_8bit}(1 + N(i - 1), j))$$

[270] In the above equation, the meaning of each expression is shown in the above embodiments, which will not be repeated here.

[271] Embodiment 30: in step 304 and step 404, for each sub-block of the current block, motion compensation is performed on the sub-block according to the first target motion vector of the sub-block and the second target motion vector of the sub-block. Specifically, the third reference block corresponding to the sub-block is determined according to the first target motion vector of the sub-block, and the fourth reference block corresponding to the sub-block is determined according to the second target motion vector of the sub-block. The third pixel value of the third reference block and the fourth pixel value of the fourth reference block are weighted to obtain the prediction value of the sub-block.

[272] For example, the reference blocks (i.e., the third and fourth reference blocks including three component prediction values, since the target motion vector may be sub-pixel, interpolation is required) in two directions are obtained by interpolating (e.g., 8-tap interpolation) based on two target motion vectors of the sub-block. The third pixel value of the third reference block and the fourth pixel value of the fourth reference block are weighted to obtain a final prediction value (three components).

[273] In an example, a fifth reference block corresponding to the sub-block can be determined from a first reference frame based on the first target motion vector of the sub-block, and the third reference block can be obtained by interpolating pixel values in the fifth reference block. A sixth reference block corresponding to the sub-block can be determined from a second reference frame based on the second target motion vector of the sub-block, and a fourth reference block can be obtained by interpolating pixel values in the sixth reference block.

[274] For example, assuming that the size of the current block is  $W \times H$ , the fifth reference block with a size of  $A \times B$  can be determined from the first reference frame based on the first target motion vector, the size  $A \times B$  of the fifth reference block being related to the interpolation method, and where  $A$  is greater than  $W$  and  $B$  is greater than  $H$ , and there

is no restriction on this. The third reference block with the size of  $W \times H$  can be obtained by interpolating pixel values in the fifth reference block, and the interpolation method is not limited here. The sixth reference block of size  $A \times B$  can be determined from the second reference frame based on the second target motion vector, the size  $A \times B$  of the sixth reference block being related to the interpolation method, and where  $A$  is greater than  $W$  and  $B$  is greater than  $H$ . The fourth reference block with the size of  $W \times H$  can be obtained by interpolating pixel values in the sixth reference block, and the interpolation method is not limited here.

[275] In another example, a seventh reference block corresponding to the sub-block is determined from the first reference frame based on the first target motion vector of the sub-block, an eighth reference block is constructed by using pixel values in the seventh reference block, and interpolates pixel values in the eighth reference block to obtain the third reference block. A ninth reference block corresponding to the sub-block is determined from the second reference frame based on the second target motion vector of the sub-block, a tenth reference block is constructed by using pixel values in the ninth reference block, and interpolate pixel values in the tenth reference block to obtain the fourth reference block.

[276] For example, assuming that the size of the current block is  $W \times H$ , the seventh reference block with a size of  $W \times H$  can be determined from the first reference frame based on the first target motion vector. The eighth reference block with a size of  $A \times B$  is constructed based on the pixel values in the seventh reference block, and the construction method is not limited. The size  $A \times B$  of the eighth reference block is related to the interpolation method, where  $A$  is greater than  $W$  and  $B$  is greater than  $H$ , and there is no restriction on this. Interpolate pixel values in the eighth reference block to obtain the third reference block with a size of  $W \times H$ , and the interpolation method is not limited.

[277] For example, assuming that the size of the current block is  $W \times H$ , the ninth

reference block with a size of  $W \times H$  can be determined from the second reference frame based on the second target motion vector. The tenth reference block with a size of  $A \times B$  is constructed based on the pixel values in the ninth reference block, and the construction method is not limited. The size  $A \times B$  of the tenth reference block is related to the interpolation method, where  $A$  is greater than  $W$  and  $B$  is greater than  $H$ , and there is no restriction on this. Interpolate pixel values in the tenth reference block to obtain the fourth reference block with a size of  $W \times H$ , and the interpolation method is not limited.

[278] Embodiment 31: after obtaining the target motion vector, based on the target motion vector of each sub-block, prediction values in two directions (i.e. three components  $Y$ ,  $U$ ,  $V$ , i.e. the prediction value of the third reference block and the prediction value of the fourth reference block) are obtained by an 8-tap interpolation filter, and are weighted to obtain the final prediction values. Referring to Fig. 5, the pixel values of the black area and the white area are obtained from the reference frame, and the pixel values of the gray area are not necessarily obtained from the reference frame, but can be obtained by copying adjacent pixel values.

[279] In an example, firstly,  $W+FS-1$  pixel values in the first row of the white area can be copied to the pixel values in the first  $SR$  rows of the gray area.  $W + FS-1$  pixel values in the last row of the white area are copied to the pixel values in the last  $SR$  rows of the gray area. Then,  $H+FS-1$  pixel values in the first column of the white area, and  $SR$  pixel values of the gray area obtained on the upper side and  $SR$  pixel values of the gray area obtained on the lower side can be copied to the pixel values in the first  $SR$  columns of the gray area.  $H+FS-1$  pixel values in the last column of the white area, and  $SR$  pixel values of the gray area obtained on the upper side and  $SR$  pixel values of the gray area obtained on the lower side can be copied to the pixel values in the last  $SR$  columns of the gray area.

[280] In another example, firstly,  $H+FS-1$  pixel values in the first column of the white

area can be copied to the pixel values in the first SR columns of the gray area.  $H+FS-1$  pixel values in the last column of the white area are copied to the pixel values in the last SR columns of the gray area. Then,  $W+FS-1$  pixel values in the first row of the white area, and SR pixel values of the gray area obtained on the left side and SR pixel values of the gray area obtained on the right side can be copied to the pixel values in the first SR rows of the gray area.  $W+FS-1$  pixel values in the last row of the white area, and SR pixel values of the gray area obtained on the left side and SR pixel values of the gray area obtained on the right side can be copied to the pixel values in the last SR rows of the gray area.

[281] Embodiment 32: after obtaining the target motion vector, based on the target motion vector of each sub-block, prediction values in two directions (i.e. three components Y, U, V, i.e. the prediction value of the third reference block and the prediction value of the fourth reference block) are obtained by an 8-tap interpolation filter, and are weighted to obtain the final prediction values. Referring to Fig. 5, the pixel values of the black area and the white area are obtained from the reference frame, and the pixel values of the gray area are obtained by copying directly from the corresponding area of the reference frame, rather than by copying adjacent pixel values. This method is simple and have better performance, but it increases the amount of access data to the reference frame.

[282] Embodiment 33: after obtaining the target motion vector, based on the target motion vector of each sub-block, prediction values in two directions (i.e. three components Y, U, V, i.e. the predicted value of the third reference block and the predicted value of the fourth reference block) are obtained by a bilinear interpolation filter (instead of the 8-tap interpolation filter), and are weighted to obtain the final prediction values. Referring to Fig. 5, the pixel values of the black area and the white area are obtained from the reference frame. Since the number of taps is relatively small, the pixel values of the gray area are not required.

[283] Embodiment 34: in embodiment 30 to embodiment 33, after obtaining the prediction values in the two directions, the final prediction value is obtained by mean of weighted average (i.e., the prediction values in the two directions have the same weight). Alternatively, after the prediction values in the two directions are obtained, the final predicted value is obtained by weighted average, and the weights of the two prediction values can be different. For example, the weight ratio of two prediction values can be 1:2, 1:3, 2:1, etc.

[284] For the encoding side, weight table can include weight ratio: 1:2, 1:3, 2:1, etc., and the encoding side can determine the cost value of each weight ratio and determine the weight ratio with the minimum cost value, so that the encoding side can obtain a final prediction value through weighted average based on the weight ratio with the minimum cost value.

[285] When the encoding side sends encoding bit stream to the decoding side, the encoding bit stream carries the index value of the weight ratio in the weight table. In this way, the decoding side analyzes the index value of the encoding bit stream, obtains weight ratio corresponding to the index value from the weight table, and obtains the final prediction value by weighted average based on the weight ratio.

[286] In an example, the weight table can include, but is not limited to  $\{-2, 3, 4, 5, 10\}$ . The weight "-2" indicates that the final prediction value = (prediction value 1\*(-2) + prediction value 2\*(8- (-2))), i.e.,  $(-2*\text{prediction value 1} + 10*\text{prediction value 2}) / 8$ . The weight "10" indicates that the weight ratio is 10: -2, i.e. final prediction value = (prediction value 1\*(10) + predicted value 2\*(-2)), i.e.  $(10*\text{prediction value 1} - 2*\text{prediction value 2}) / 8$ . The weight "3" indicates a weight ratio of 3: 5. The weight "5" indicates a weight ratio of 5: 3. The weight "4" indicates a weight ratio of 4: 4, i.e., the weight "4" indicates that the weights are the same.

[287] Embodiment 35: in step 305 and step 405, the first and second target motion vectors of each sub-block of the current block need to be stored, and the first and second target motion vectors are used for encoding and decoding reference of a subsequent block. For example, the first and second target motion vectors are used for loop filtering of the current frame; the first and second target motion vectors are used for time domain reference of subsequent frames; and/or the first and second target motion vectors are used for spatial domain reference of the current frame.

[288] For example, the first and second target motion vectors of each sub-block of the current block can be used for motion compensation of the current block, and can also be used for time domain reference of subsequent frames. In another example, the first and second target motion vectors of each sub-block of the current block can be used for motion compensation of the current block, can also be used for a loop filtering of the current block, and can also be used for time domain reference of subsequent frames.

[289] In another example, the first and second target motion vectors of each sub-block of the current block can be used for motion compensation of the current block, can also be used for a loop filtering of the current block, can also be used for time domain reference of subsequent frames, and can also be used for a spatial domain reference of the current frame, which will be described below.

[290] The first and second target motion vectors of each sub-block of the current block can be used for spatial domain reference of blocks within certain LCUs (Largest Coding units) in spatial domain. Since the encoding sequence is from top to bottom and from left to right, the motion vector of the current block can be referenced by other blocks within the current LCU, as well as by blocks within the subsequent adjacent LCU. Since the amount of calculation required for obtaining the target motion vector is large, if the subsequent blocks refer to the target motion vector of the current block, it will take a long time to wait.

In order to avoid the time delay caused by excessive waiting, only a few spatial domain adjacent blocks are allowed to refer to the target motion vector of the current block, and other blocks refer to the original motion vector of the current block. Referring to Fig. 8, the few blocks include sub-blocks located in a lower LCU and a lower-right LCU located at a lower side of the current LCU, and sub-blocks located in a right-side LCU and a lower-left LCU may not refer to the target motion vector of the current block.

[291] Embodiment 36: the following describes an adjustment process of a motion vector with reference to a specific example. The specific steps of the motion vector adjustment can be as follows, the following "copy" indicates obtaining without interpolation, if the MV is integer-pixel offset, it can be directly copied from the reference frame, otherwise interpolation is needed.

[292] At step e1, if the current block adopts merge or skip mode, the subsequent processing is executed.

[293] At step e2, the reference pixel values are prepared (assuming the width of the current block is W and the height is H).

[294] An integer-pixel block is prepared for step e3: on the basis of the original motion vector (original motion vector of list0 is denoted as Org\_MV0, original motion vector of list1 is denoted as Org\_MV1), integer-pixel blocks of three components having two blocks of area  $(W+FS-1)(H+FS-1)$  are copied at the respective position of respective reference frames.

[295] An integer-pixel block is prepared for step e4: on the basis of the integer-pixel blocks with area of  $(W+FS-1)(H+FS-1)$ , the integer-pixel blocks of three components with area of  $(W+FS-1)(H+FS-1)$  are expanded up, down, left and right by SR rows/columns, and after expansion, integer-pixel blocks of three components with area of  $(W+FS-1+2*SR)(H+FS-1+2*SR)$  are obtained, which are denoted as Pred\_Inter0, Pred\_Inter1 as shown in

Fig. 5.

[296] A first motion compensation is performed based on motion information in two different directions. For example, for the luminance component (since the subsequent search process only calculates a cost value using the luminance component to reduce complexity), initial reference pixel prediction values (denoted as  $Pred\_Bilinear0$ ,  $Pred\_Bilinear1$ ) with the size of  $(W + 2 \cdot SR) \cdot (H + 2 \cdot SR)$  can be obtained by bilinear interpolation based on two integer-pixel reference blocks with the area of  $(W+FS-1) \cdot (H+FS-1)$ . FS is the number of filter taps, and is default 8, and SR is the search range, i.e., the maximum horizontal/vertical component interpolation of the target motion vector and the original motion vector, and is default 2.  $Pred\_Bilinear0/Pred\_Bilinear1$  is used in step e3.

[297] At step e3, for each  $dx \cdot dy$  sub-block ( $16 \cdot 16$  or less integer block) of the current block, the target motion vectors are respectively obtained (two target motion vectors are respectively denoted as  $Refined\_MV0$  and  $Refined\_MV1$ ).

[298] At step e31, SR iterations are performed to obtain an optimal integer-pixel offset of the integer-pixel MV point, which is denoted as  $IntegerDeltaMV$ , and  $IntegerDeltaMV$  is initialized to  $(0, 0)$ , and performing the subsequent processing for each iteration:

[299] at step e311,  $deltaMV$  is set to  $(0, 0)$ . If the first iteration is performed, two prediction value blocks (i.e., block of the most center  $W \cdot H$  of  $Pred\_Bilinear0/1$ ) are obtained by copying based on the original motion vectors in the reference pixel  $Pred\_Bilinear0/1$ ; and an initial cost value, i.e., the SAD after the down-sampling operation is performed vertically on the prediction value blocks in two directions by 2 times, is obtained based on the two prediction value blocks.

[300] If the initial cost value is less than  $4 \cdot dx \cdot dy / 2$ , and  $dx$  and  $dy$  are the width and height of the current sub-block, the subsequent search process is directly skipped, step e32

is executed, and notZeroCost is set to false.

[301] At step e312, as shown in Fig. 6, five offset MVs (which are all referred to as MVOffset) are sequentially obtained according to the sequence of {Mv(0,1), Mv(0,-1), Mv(1,0), Mv(-1,0), Mv(right, down)}, with the above initial point taken as the center, and the calculation and comparison process of the cost values of these five offset MVs is performed.

[302] For example, based on a certain MVOffset, in the reference pixel Pred\_Bilinear0/1, two prediction value blocks (e.g., a W\*H block with center position offset MVOffset in the Pred\_Bilinear0, and a W\*H block with center position offset -MVOffset (opposite to list0) in the Pred\_Bilinear1) are obtained from this MVOffset, and the down-sampling SAD of the two prediction value blocks are calculated as the cost value of MVOffset. Then, the MVOffset (stored in deltaMV) with the minimum cost value is reserved, which is used as a new center offset point of the next iteration.

[303] The default value of Mv(right, down) is (-1, -1), if the cost value of Mv(1,0) is less than that of Mv(-1, 0), right is 1; if the cost value of Mv(0, 1) is less than that of Mv(0, -1), down is 1.

[304] IntegerDeltaMV is updated based on the value of deltaMV: IntegerDeltaMV: = IntegerDeltaMV + deltaMV.

[305] At step e313, after iteration, if the optimal MV is still the initial MV or the minimum cost value is 0, the next iterative search process is not performed, step e32 is executed, and notZeroCost is set as false. Otherwise, if the iteration times reach SR, step e32 is executed, and if the iteration times do not reach SR, the optimal MV is taken as the center, and the next iterative search process is performed, that is, step e311 is returned to.

[306] At step e32, the optimal integer-pixel MV point in step e31 is taken as the center, the optimal sub-pixel offset MV is obtained, which is denoted as SPMV, and the SPMV is

initialized to (0, 0), and then performing the subsequent procedure:

[307] At step e321, subsequent procedure is only performed when notZeroCost is not false and deltaMV is (0, 0), otherwise, the original motion vector is directly adjusted by using IntegerDeltaMV.

[308] At step e322,  $E(x, y)$  is represented as the cost value corresponding to the MV of the optimal MV point offset (x, y) obtained in step e31 (the cost value calculated in step e31). Based on the  $E(x, y)$  of the center, top, bottom, left and right points, the offset  $(x_0, y_0)$  of the point with the smallest  $E(x, y)$  can be obtained as follows:  $x_0 = N * (E(-1, 0) - E(1, 0)) / (E(-1, 0) + E(1, 0) - 2 * E(0, 0))$ ,  $y_0 = N * (E(0, -1) - E(0, 1)) / (E(0, -1) + E(0, 1) - 2 * E(0, 0))$ .

[309] In an example, when the motion vector pixel precision is 1/2, 1/4, 1/8, and 1/16,  $N = 1, 2, 4$ , and 8 respectively. Then,  $(x_0, y_0)$  is assigned to deltaMv,  $SPMV = \text{deltaMv} / 2N$ , and if the current motion vector pixel precision is 1/16, the SPMV can be  $(x_0/16, y_0/16)$ .

[310] If  $E(-1, 0) = E(0, 0)$ , a shift is performed by half a pixel horizontally to the left ( $\text{deltaMv}[0] = -N$ ).

[311] If  $E(1, 0) = E(0, 0)$ , a shift is performed by half a pixel horizontally to the right ( $\text{deltaMv}[0] = N$ ).

[312] If  $E(0, -1) = E(0, 0)$ , a shift is performed by half a pixel vertically upward ( $\text{deltaMv}[1] = -N$ ).

[313] If  $E(0, 1) = E(0, 0)$ , a shift is performed by half a pixel vertically downward ( $\text{deltaMv}[1] = N$ ).

[314] At step e33, the optimal offset MV is obtained based on the integer-pixel offset  $\text{IntegerDeltaMV}$  in step e31 and the sub-pixel offset SPMV in step e32, which is denoted as  $\text{BestMVoffset}$ .  $\text{BestMVoffset} = \text{IntegerDeltaMV} + \text{SPMV}$ . Based on  $\text{BestMVoffset}$ , target motion vectors in two directions can be obtained:  $\text{Refined\_MV0} =$

$\text{Org\_MV0} + \text{BestMVoffset}$ ;  $\text{Refined\_MV1} = \text{Org\_MV1} - \text{BestMVoffset}$ .

[315] At step e4, prediction values in two directions are obtained by 8-tap interpolation based on the target motion vector of each sub-block, and weighted to obtain final the prediction value (three components). For example, based on the target motion vectors  $\text{Refined\_MV0}$  and  $\text{Refined\_MV1}$  of each sub-block, in  $\text{Pred\_Inter0/1}$  obtained by step e2, the corresponding prediction block is obtained by interpolation (the motion vectors may be sub-pixel, and interpolation is required to obtain the corresponding pixel block).

[316] At step e5, the target motion vector is used for motion compensation of the current block and time domain reference of the subsequent frame.

[317] Embodiment 37: the above embodiments can be implemented individually or in any combination. For example, embodiments 13, 15, 24 and 29 can be implemented individually. Embodiment 13 and embodiment 15 are implemented in combination, embodiment 13 and embodiment 24 are implemented in combination, embodiment 13 and embodiment 29 are implemented in combination, embodiment 15 and embodiment 24 are implemented in combination, embodiment 15 and embodiment 29 are implemented in combination, embodiment 24 and embodiment 29 are implemented in combination, embodiment 13, embodiment 15 and embodiment 24 are implemented in combination, embodiment 13, embodiment 15 and embodiment 29 are implemented in combination, embodiment 15, embodiment 24 and embodiment 29 are implemented in combination, embodiment 13, embodiment 15, embodiment 24 and embodiment 29 are implemented in combination, and the like. Of course, the above are only some examples which are not limited hereto. All the embodiments of the present application can be implemented individually or in combination.

[318] Embodiment 38:

[319] Based on the same concept as the method, an embodiment of the present

application further relates to an encoding and decoding apparatus, which is applied to an encoding side or a decoding side, as shown in Fig. 9 which is a structural diagram of the apparatus, where the apparatus includes:

[320] a determining module 91 configured for, if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; a processing module 92 configured for adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector; and an encoding and decoding module 93 configured for encoding or decoding the current block according to the first and second target motion vectors.

[321] The characteristic information includes one or more of the following: a motion information prediction mode corresponding to the current block; a motion information attribute corresponding to the current block; size information of the current block.

[322] If the characteristic information is the motion information prediction mode corresponding to the current block, the determining module 91 is further configured for: if the motion information prediction mode corresponding to the current block is a regular merge mode, determining that the motion information prediction mode corresponding to the current block meets the specific condition; or, if the motion information prediction mode corresponding to the current block is a merge mode for generating a new prediction value by combining the inter-frame prediction value and the intra-frame prediction value, determining that the motion information prediction mode corresponding to the current

block meets the specific condition.

[323] If the characteristic information is a motion information attribute corresponding to the current block, the determining module 91 is further configured for: if the motion information attribute corresponding to the current block is that the motion information of the current block includes motion information in two different directions, determining that the motion information attribute corresponding to the current block meets the specific condition; or, if the motion information attribute corresponding to the current block is that the motion information of the current block includes motion information in two different directions, and the distances between the current frame and each of two reference frames corresponding to the motion information in the two different directions are the same, determining that the motion information attribute corresponding to the current block meets the specific condition; or if the motion information attribute corresponding to the current block is that the current block reuses the motion information of surrounding blocks, determining that the motion information attribute corresponding to the current block meets the specific condition; or, if the motion information attribute corresponding to the current block is that the motion information of each sub-block of the current block is the same, determining that the motion information attribute corresponding to the current block meets the specific condition.

[324] If the characteristic information is size information of the current block, the size information of the current block includes a width value and a height value of the current block, the determining module 91 is further configured for: if the width value of the current block is within a range of a first interval [a first threshold, a second threshold], determining that the size information of the current block meets the specific condition; or, if the height value of the current block is within a range of a second interval [a third threshold, a fourth threshold], determining that the size information of the current block meets the specific condition; or, if the area derived from the width value and the height value of the current

block is within a range of a third interval [a fifth threshold, a sixth threshold], determining that the size information of the current block meets the specific condition; or, if the width value is within the range of the first interval [the first threshold, the second threshold], the height value is within the range of the second interval [the third threshold, the fourth threshold], and the area is within the range of the third interval [the fifth threshold, the sixth threshold], determining that the size information of the current block meets the specific condition.

[325] The determining module 91 is configured for determining the first reference block corresponding to the current block according to the first original motion vector of the current block, and determining the second reference block corresponding to the current block according to the second original motion vector of the current block, which is specifically for: determining the first reference block corresponding to the current block from the first reference frame based on the first original motion vector of the current block; and determining the second reference block corresponding to the current block from the second reference frame based on the second original motion vector of the current block; wherein the first pixel value of each pixel point in the first reference block is obtained by interpolating pixel values of adjacent pixel points in the first reference block, or is obtained by copying the pixel values of adjacent pixel points in the first reference block; the second pixel value of each pixel point in the second reference block is obtained by interpolating pixel values of adjacent pixel points in the second reference block, or by copying the pixel values of adjacent pixel points in the second reference block.

[326] The processing module 92 is configured for adjusting the first and second original motion vectors according to the first pixel value of the first reference block and the second pixel value of the second reference block to obtain the first target motion vector corresponding to the first original motion vector and the second target motion vector corresponding to the second original motion vector, which is specifically for:

[327] when the current block includes at least one sub-block, for each sub-block of the current block, adjusting the first and second original motion vectors according to the first and second pixel values to obtain the first and second target motion vectors of the sub-block.

[328] For example, determining a first integer-pixel motion vector adjustment value and a second integer-pixel motion vector adjustment value of the sub-block, and/or a first sub-pixel motion vector adjustment value and a second sub-pixel motion vector adjustment value of the sub-block according to the first and second pixel values; adjusting the first original motion vector according to the first integer-pixel motion vector adjustment value and/or the first sub-pixel motion vector adjustment value to obtain the first target motion vector of the sub-block; and adjusting the second original motion vector according to the second integer-pixel motion vector adjustment value and/or the second sub-pixel motion vector adjustment value to obtain the second target motion vector of the sub-block.

[329] The processing module 92 is configured for determining the first and second integer-pixel motion vector adjustment values of the sub-block, the first and second sub-pixel motion vector adjustment values of the sub-block according to the first and second pixel values, which is specifically for:

[330] determining the first original motion vector or the second original motion vector as a center motion vector;

[331] determining an edge motion vector corresponding to the center motion vector;

[332] obtaining a first cost value corresponding to the center motion vector and a second cost value corresponding to the edge motion vector according to the first and second pixel values;

[333] selecting a motion vector from the center motion vector and the edge motion vector as an optimal motion vector according to the first cost value and the second cost

value;

[334] judging whether an end condition is met; if not, determining the optimal motion vector as the center motion vector, and returning to execute determining the edge motion vector corresponding to the center motion vector;

[335] if yes, determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector; and determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector.

[336] The processing module 92 is configured for determining the edge motion vector corresponding to the center motion vector, which is specifically for:

[337] shifting the center motion vector  $(x, y)$  towards different directions by  $S$ , and sequentially obtaining edge motion vector  $(x, y+S)$ , edge motion vector  $(x, y-S)$ , edge motion vector  $(x+S, y)$ , edge motion vector  $(x-S, y)$  and edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions; or

[338] shifting the center motion vector  $(x, y)$  towards different directions by  $S$ , and sequentially obtaining edge motion vector  $(x, y-S)$ , edge motion vector  $(x, y+S)$ , edge motion vector  $(x-S, y)$ , edge motion vector  $(x+S, y)$  and edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions;

[339] wherein default value of the edge motion vector  $(x+\text{right}, y+\text{down})$  is  $(x-S, y-S)$ ;

[340] if the cost value of the edge motion vector  $(x+S, y)$  is smaller than the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y+S)$  is less than that of the edge motion vector  $(x, y-S)$ , down is  $S$ ; or if the cost value of the edge motion vector  $(x+S, y)$  is less than or equal to the cost value of the edge motion

vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y + S)$  is less than or equal to the cost value of the edge motion vector  $(x, y-S)$ , down is  $S$ .

[341] The processing module 92 is configured for obtaining the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values, which is specifically for:

[342] acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values without down-sampling; or

[343] performing down-sampling operation on the first and second pixel values; and obtaining the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values after down-sampling; or

[344] performing shifting and down-sampling operation on the first pixel value, and performing shifting and down-sampling operation on the second pixel value; and acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first and second pixel values after shifting and down-sampling.

[345] The processing module 92 is configured for determining the first and second integer-pixel motion vector adjustment values of the sub-block, the first and second sub-pixel motion vector adjustment values of the sub-block according to the first and second pixel values, which is specifically for:

[346] taking the first original motion vector or the second original motion vector as a center, selecting some or all of the motion vectors from surrounding motion vectors, and determining the selected motion vectors as candidate motion vectors;

[347] acquiring a third cost value corresponding to the first original motion vector or the second original motion vector and a fourth cost value corresponding to each of the candidate motion vectors according to the first and second pixel values;

[348] selecting a motion vector from the first original motion vector or the second original motion vector and the candidate motion vector as an optimal motion vector according to the third cost value and the fourth cost value;

[349] determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector; and determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector.

[350] The processing module 92 is configured for determining the first and second integer-pixel motion vector adjustment values of the sub-block according to the optimal motion vector, which is specifically for: determining the first integer-pixel motion vector adjustment value of the sub-block according to the optimal motion vector and the first original motion vector, and determining the second integer-pixel motion vector adjustment value of the sub-block according to the first integer-pixel motion vector adjustment value.

[351] The processing module 92 is configured for determining the first and second sub-pixel motion vector adjustment values of the sub-block according to the optimal motion vector, which is specifically for: determining the first sub-pixel motion vector adjustment value of the sub-block according to the cost value corresponding to the optimal motion vector and the cost value corresponding to the edge motion vector corresponding to the optimal motion vector, and determining the second sub-pixel motion vector adjustment value of the sub-block according to the first sub-pixel motion vector adjustment value.

[352] The encoding and decoding module 93 is configured for encoding or decoding the current block according to the first and second target motion vectors, which is

specifically for: when the current block includes at least one sub-block, for each sub-block of the current block, determining a third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining a fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block; weighting a third pixel value of the third reference block and a fourth pixel value of the fourth reference block to obtain a prediction value of the sub-block; and determining a prediction value of the current block according to the prediction value of each sub-block.

[353] The encoding and decoding module 93 is configured for determining the third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining the fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block, which is specifically for: determining a fifth reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, and interpolating pixel values in the fifth reference block to obtain the third reference block; determining a sixth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, and interpolating pixel values in the sixth reference block to obtain the fourth reference block.

[354] The encoding and decoding module 93 is configured for determining the third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining the fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block, which is specifically for: determining a seventh reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, constructing an eighth reference block by using pixel values in the seventh reference block, and interpolating pixel values in the eighth reference block to obtain the third reference block;

determining a ninth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, constructing a tenth reference block by using pixel values in the ninth reference block, and interpolating pixel values in the tenth reference block to obtain the fourth reference block.

[355] The apparatus further includes: a storing module for storing the first and second target motion vectors for the current block; wherein the first and second target motion vectors are used for loop filtering of the current frame, the first and second target motion vectors are used for time domain reference of subsequent frames, and/or the first and second target motion vectors are used for spatial domain reference of the current frame.

[356] Embodiment 39:

[357] The decoding-side device according to the embodiment of the present application, in terms of a hardware level, a hardware structure diagram of the decoding-side device can specifically refer to Fig. 10. The device includes: a processor 101 and a machine-readable storage medium 102, wherein the machine-readable storage medium 102 has stored thereon machine-executable instructions executable by the processor 101; the processor 101 is configured for executing the machine-executable instructions to implement the methods disclosed in the above examples of the present application. For example, the processor is used to execute machine-executable instructions to implement the steps of:

[358] if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector

and a second target motion vector corresponding to the second original motion vector; and decoding the current block according to the first and second target motion vectors.

[359] The encoding-side device according to the embodiment of the present application, in terms of a hardware level, a hardware structure diagram of the encoding-side device can specifically refer to Fig. 11. The device includes: a processor 111 and a machine-readable storage medium 112, wherein the machine-readable storage medium 112 has stored thereon machine-executable instructions executable by the processor 111; the processor 111 is configured for executing the machine-executable instructions to implement the methods disclosed in the above examples of the present application. For example, the processor 111 is used to execute machine-executable instructions to implement the steps of: if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block; adjusting the first and second original motion vectors according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector; and encoding the current block according to the first and second target motion vectors.

[360] Based on the same concept as the method, the embodiment of the present application further relates to a machine-readable storage medium, wherein the machine-readable storage medium has stored thereon a plurality of computer instructions, and when the computer instructions are executed by a processor, the encoding and decoding methods disclosed in the above examples of the present application can be implemented. The machine-readable storage medium may be any electronic, magnetic, optical, or other physical storage apparatus that can contain or store information such as executable

instructions, data, and the like. For example, the machine-readable storage medium can be: RAM (Random Access Memory), volatile memory, non-volatile memory, flash memory, a storage drive(e.g., a hard drive), a solid state drive, any type of storage disk (e.g., a compact disk, a DVD, etc.), or similar storage medium, or a combination thereof.

[361] The systems, apparatuses, modules or units described in the above embodiments can be specifically implemented by a computer chip or an entity, or implemented by a product with certain functions. A typical implementation device is a computer, which can be in the form of a personal computer, laptop, cellular telephone, camera phone, smart phone, personal digital assistant, media player, navigation device, email messaging device, game console, tablet computer, wearable device, or a combination of any of these devices.

[362] For ease of description, the above devices are separately described in various units according to their functions. Of course, when implementing this application, the functions of each unit can be realized in one or more software and / or hardware.

[363] Those skilled in the art will appreciate that the embodiments of the present application may be disclosed by the methods, systems, or computer program products. Accordingly, the present application may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, embodiments of the present application may take the form of a computer program product embodied on one or more computer-usable storage media (including, but not limited to, disk storage, CD-ROM, optical storage, and the like) having computer-usable program code embodied therein.

[364] The present application is described with reference to flowchart illustrations and/or block diagrams of methods, devices (systems), and computer program products according to embodiments of the application. It will be understood that each flow and/or block of the flowchart illustrations and/or block diagrams, and combinations of flows

and/or blocks of the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions can be provided to a processor of a general purpose computer, special purpose computer, embedded processor, or other programmable data processing device to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing device, produce apparatus for implementing the functions specified in the flowchart flow or flows and/or block diagram block or blocks.

[365] Furthermore, these computer program instructions can also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction apparatus which implement the function specified in the flowchart flow or flows and/or block diagram block or blocks.

[366] These computer program instructions can also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable device to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus relates to steps for implementing the functions specified in the flowchart flow or flows and/or block diagram block or blocks.

[367] The above description is only an embodiment of the present application and is not intended to limit the present application. Various modifications and changes may occur to those skilled in the art to which the present application pertains. Any modification, equivalent replacement, improvement or the like made within the spirit and principle of the present application shall be included in the scope of the claims of the present application.



## CLAIMS

### **What is claimed is:**

1. An encoding and decoding method, comprising:

if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block;

adjusting the first original motion vector and the second original motion vector according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector;

encoding or decoding the current block according to the first target motion vector and the second target motion vector.

2. The method of claim 1, wherein,

the characteristic information comprises one or more of the following: a motion information prediction mode corresponding to the current block; a motion information attribute corresponding to the current block; size information of the current block.

3. The method of claim 2, wherein when the characteristic information is the motion information prediction mode corresponding to the current block, the characteristic information of the current block meets a specific condition, comprises:

if the motion information prediction mode corresponding to the current block is a regular merge mode, determining that the motion information prediction mode

corresponding to the current block meets the specific condition; or

if the motion information prediction mode corresponding to the current block is a merge mode for generating a new prediction value by combining an inter-frame prediction value and an intra-frame prediction value, determining that the motion information prediction mode corresponding to the current block meets the specific condition.

4. The method of claim 2, wherein when the characteristic information is a motion information attribute corresponding to the current block, the characteristic information of the current block meets a specific condition, comprises:

if the motion information attribute corresponding to the current block is that the motion information of the current block comprises motion information in two different directions, determining that the motion information attribute corresponding to the current block meets the specific condition; or

if the motion information attribute corresponding to the current block is that the motion information of the current block comprises motion information in two different directions, and distances between a current frame and each of two reference frames corresponding to the motion information in the two different directions are the same, determining that the motion information attribute corresponding to the current block meets the specific condition; or

if the motion information attribute corresponding to the current block is that the current block reuses motion information of surrounding blocks, determining that the motion information attribute corresponding to the current block meets the specific condition; or

if the motion information attribute corresponding to the current block is that motion information of each sub-block of the current block is the same, determining that the motion information attribute corresponding to the current block meets the specific condition.

5. The method of claim 2, wherein when the characteristic information is size information of the current block, the size information of the current block comprises a width value and a height value of the current block, the characteristic information of the current block meets a specific condition, comprises:

if the width value of the current block is within a range of a first interval [a first threshold, a second threshold], determining that the size information of the current block meets the specific condition; or

if the height value of the current block is within a range of a second interval [a third threshold, a fourth threshold], determining that the size information of the current block meets the specific condition; or

if an area derived from the width value and the height value of the current block is within a range of a third interval [a fifth threshold, a sixth threshold], determining that the size information of the current block meets the specific condition; or

if the width value is within the range of the first interval [the first threshold, the second threshold], the height value is within the range of the second interval [the third threshold, the fourth threshold], and the area is within the range of the third interval [the fifth threshold, the sixth threshold], determining that the size information of the current block meets the specific condition.

6. The method of claim 1, wherein determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block, comprises:

determining the first reference block corresponding to the current block from the first reference frame based on the first original motion vector of the current block;

determining the second reference block corresponding to the current block from the second reference frame based on the second original motion vector of the current block;

wherein, the first pixel value of each pixel point in the first reference block is obtained by interpolating pixel values of adjacent pixel points in the first reference block, or is obtained by copying the pixel values of adjacent pixel points in the first reference block;

the second pixel value of each pixel point in the second reference block is obtained by interpolating pixel values of adjacent pixel points in the second reference block, or is obtained by copying the pixel values of adjacent pixel points in the second reference block.

7. The method of claim 1, wherein adjusting the first original motion vector and the second original motion vector according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector, comprises:

when the current block comprises at least one sub-block, for each sub-block of the current block, adjusting the first original motion vector and the second original motion vector according to the first pixel value and the second pixel value to obtain a first target motion vector of the sub-block and a second target motion vector of the sub-block.

8. The method of claim 7, wherein adjusting the first original motion vector and the second original motion vector according to the first pixel value and the second pixel value to obtain a first target motion vector of the sub-block and a second target motion vector of the sub-block, comprises:

determining first and second integer-pixel motion vector adjustment values of the sub-block, and/or first and second sub-pixel motion vector adjustment values of the sub-block according to the first pixel value and the second pixel value;

adjusting the first original motion vector, according to the first integer-pixel motion vector adjustment value and/or the first sub-pixel motion vector adjustment value, to obtain the first target motion vector of the sub-block;

adjusting the second original motion vector, according to the second integer-pixel motion vector adjustment value and/or the second sub-pixel motion vector adjustment value, to obtain the second target motion vector of the sub-block.

9. The method of claim 8, wherein,

determining first and second integer-pixel motion vector adjustment values of the sub-block, and/or first and second sub-pixel motion vector adjustment values of the sub-block according to the first pixel value and the second pixel value, comprising:

determining the first original motion vector or the second original motion vector as a center motion vector;

determining an edge motion vector corresponding to the center motion vector;

obtaining a first cost value corresponding to the center motion vector and a second cost value corresponding to the edge motion vector according to the first pixel value and the second pixel value;

selecting a motion vector from the center motion vector and the edge motion vector as an optimal motion vector according to the first cost value and the second cost value;

judging whether an end condition is met;

if the end condition is not met, determining the optimal motion vector as the center motion vector, and returning to execute: determining an edge motion vector corresponding to the center motion vector;

if the end condition is met, determining the first integer-pixel motion vector adjustment value and the second integer-pixel motion vector adjustment value of the sub-block

according to the optimal motion vector; and determining the first sub-pixel motion vector adjustment value and the second sub-pixel motion vector adjustment value of the sub-block according to the optimal motion vector.

10. The method of claim 9, wherein,

determining an edge motion vector corresponding to the center motion vector, comprises:

shifting the center motion vector  $(x, y)$  towards different directions by  $S$  to sequentially obtain an edge motion vector  $(x, y+S)$ , an edge motion vector  $(x, y-S)$ , an edge motion vector  $(x+S, y)$ , an edge motion vector  $(x-S, y)$  and an edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions; or

shifting the center motion vector  $(x, y)$  towards different directions by  $S$  to sequentially obtain an edge motion vector  $(x, y-S)$ , an edge motion vector  $(x, y+S)$ , an edge motion vector  $(x-S, y)$ , an edge motion vector  $(x+S, y)$  and an edge motion vector  $(x+\text{right}, y+\text{down})$  in different directions;

wherein, a default value of the edge motion vector  $(x+\text{right}, y+\text{down})$  is  $(x-S, y-S)$ ;

wherein, if the cost value of the edge motion vector  $(x+S, y)$  is smaller than the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y+S)$  is less than the cost value of the edge motion vector  $(x, y-S)$ , down is  $S$ ; or

if the cost value of the edge motion vector  $(x+S, y)$  is less than or equal to the cost value of the edge motion vector  $(x-S, y)$ , right is  $S$ ; if the cost value of the edge motion vector  $(x, y+S)$  is less than or equal to the cost value of the edge motion vector  $(x, y-S)$ , down is  $S$ .

11. The method of claim 9, wherein,

obtaining a first cost value corresponding to the center motion vector and a second cost

value corresponding to the edge motion vector according to the first pixel value and the second pixel value, comprises:

acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first pixel value and the second pixel value without down-sampling; or

performing down-sampling operation on the first pixel value and on the second pixel value; and obtaining the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first pixel value and the second pixel value after down-sampling; or

performing shifting and down-sampling operation on the first pixel value and on the second pixel value; and acquiring the first cost value corresponding to the center motion vector and the second cost value corresponding to the edge motion vector according to the first pixel value and the second pixel value after the shifting and down-sampling operation.

12. The method of claim 8, wherein,

determining first and second integer-pixel motion vector adjustment values of the sub-block, and/or first and second sub-pixel motion vector adjustment values of the sub-block according to the first pixel value and the second pixel value, comprises:

taking the first original motion vector or the second original motion vector as a center, selecting some or all of the motion vectors from surrounding motion vectors, and determining the selected motion vectors as candidate motion vectors;

acquiring a third cost value corresponding to the first original motion vector or the second original motion vector and a fourth cost value corresponding to each of the candidate motion vectors according to the first pixel value and the second pixel value;

selecting a motion vector from the first original motion vector or the second original

motion vector and the candidate motion vectors as an optimal motion vector according to the third cost value and the fourth cost value;

determining the first integer-pixel motion vector adjustment value and the second integer-pixel motion vector adjustment value of the sub-block according to the optimal motion vector; and determining the first sub-pixel motion vector adjustment value and the second sub-pixel motion vector adjustment value of the sub-block according to the optimal motion vector.

13. The method of claim 9 or 12, wherein,

determining the first integer-pixel motion vector adjustment value and the second integer-pixel motion vector adjustment value of the sub-block according to the optimal motion vector, comprises:

determining the first integer-pixel motion vector adjustment value of the sub-block according to the optimal motion vector and the first original motion vector, and determining the second integer-pixel motion vector adjustment value of the sub-block according to the first integer-pixel motion vector adjustment value.

14. The method of claim 9 or 12, wherein,

determining the first sub-pixel motion vector adjustment value and the second sub-pixel motion vector adjustment value of the sub-block according to the optimal motion vector, comprises:

determining the first sub-pixel motion vector adjustment value of the sub-block according to a cost value corresponding to the optimal motion vector and a cost value corresponding to the edge motion vector corresponding to the optimal motion vector, and determining the second sub-pixel motion vector adjustment value of the sub-block according to the first sub-pixel motion vector adjustment value.

15. The method of claim 1, wherein encoding or decoding the current block according to the first target motion vector and the second target motion vector, comprises:

when the current block comprises at least one sub-block, for each sub-block of the current block, determining a third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining a fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block;

weighting a third pixel value of the third reference block and a fourth pixel value of the fourth reference block to obtain a prediction value of the sub-block;

determining a prediction value of the current block according to the prediction value of each sub-block.

16. The method of claim 15, wherein determining a third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining a fourth reference block corresponding to the sub-block according to the second target motion vector of the sub-block, comprises:

determining a fifth reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, and interpolating pixel values in the fifth reference block to obtain the third reference block;

determining a sixth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, and interpolating pixel values in the sixth reference block to obtain the fourth reference block.

17. The method of claim 15, wherein determining a third reference block corresponding to the sub-block according to the first target motion vector of the sub-block, and determining a fourth reference block corresponding to the sub-block according to the

second target motion vector of the sub-block, comprises:

determining a seventh reference block corresponding to the sub-block from the first reference frame based on the first target motion vector of the sub-block, constructing an eighth reference block by using pixel values in the seventh reference block, and interpolating pixel values in the eighth reference block to obtain the third reference block;

determining a ninth reference block corresponding to the sub-block from the second reference frame based on the second target motion vector of the sub-block, constructing a tenth reference block by using pixel values in the ninth reference block, and interpolating pixel values in the tenth reference block to obtain the fourth reference block.

18. The method of claim 1, wherein, after encoding or decoding the current block according to the first target motion vector and the second target motion vector, the method further comprises:

storing the first target motion vector and the second target motion vector for the current block,

wherein, the first target motion vector and the second target motion vector are used for loop filtering of the current frame, the first target motion vector and the second target motion vector are used for time domain reference of subsequent frames, and/or the first target motion vector and the second target motion vector are used for spatial domain reference of the current frame.

19. An encoding and decoding apparatus, comprising:

a determining module configured for, if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original

motion vector of the current block;

a processing module configured for adjusting the first original motion vector and the second original motion vector according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector;

an encoding and decoding module configured for encoding or decoding the current block according to the first target motion vector and the second target motion vector.

20. An encoding-side device, comprising a processor and a machine-readable storage medium, wherein the machine-readable storage medium has stored thereon machine-executable instructions executable by the processor;

the processor is configured for executing the machine-executable instructions to implement the steps of:

if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block;

adjusting the first original motion vector and the second original motion vector according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector;

encoding the current block according to the first target motion vector and the second

target motion vector.

21. A decoding-side device, comprising a processor and a machine-readable storage medium, wherein the machine-readable storage medium has stored thereon machine-executable instructions executable by the processor;

the processor is configured for executing the machine-executable instructions to implement the steps of:

if characteristic information of a current block meets a specific condition, determining a first reference block corresponding to the current block according to a first original motion vector of the current block, and determining a second reference block corresponding to the current block according to a second original motion vector of the current block;

adjusting the first original motion vector and the second original motion vector according to a first pixel value of the first reference block and a second pixel value of the second reference block to obtain a first target motion vector corresponding to the first original motion vector and a second target motion vector corresponding to the second original motion vector;

decoding the current block according to the first target motion vector and the second target motion vector.

22. A machine-readable storage medium, comprising computer instructions stored thereon which, when executed by a processor, cause the processor to implement the method of any one of claims 1-18.