United States Patent Application Publication	20250267250
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	ZHAO; Lei et al.

METHOD, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING

Abstract

Embodiments of the present disclosure provide a solution for video processing. A method for video processing is proposed. The method comprises: determining, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and performing the conversion based on the reference template.

Inventors: **ZHAO; Lei (Beijing, CN), ZHANG; Kai (Los Angeles, CA), ZHANG; Li (Los Angeles, CA)**

Applicant: **Douyin Vision Co., Ltd.** (Beijing, CN); **Bytedance Inc.** (Los Angeles, CA)

Family ID: **1000008612317**

Appl. No.: **19/183691**

Filed: **April 18, 2025**

Foreign Application Priority Data

WO	PCT/CN2022/126512	Oct. 20, 2022

Related U.S. Application Data

parent WO continuation PCT/CN2023/125506 20231019 PENDING child US 19183691

Publication Classification

Int. Cl.: **H04N19/105** (20140101); **H04N19/159** (20140101); **H04N19/176** (20140101)

U.S. Cl.:

CPC **H04N19/105** (20141101); **H04N19/159** (20141101); **H04N19/176** (20141101);

Background/Summary

CROSS REFERENCE TO RELATED APPLICATION [0001] This application is a continuation of International Application No. PCT/CN2023/125506, filed on Oct. 19, 2023, which claims the benefit of International Application No. PCT/CN2022/126512 filed on Oct. 20, 2022. The entire contents of these applications are hereby incorporated by reference in their entireties.

FIELDS

[0002] Embodiments of the present disclosure relate generally to video processing techniques, and more particularly, to motion candidate list construction.

BACKGROUND

[0003] In nowadays, digital video capabilities are being applied in various aspects of peoples' lives. Multiple types of video compression technologies, such as MPEG-2, MPEG-4, ITU-TH.263, ITU-TH.264/MPEG-4 Part 10 Advanced Video Coding (AVC), ITU-TH.265 high efficiency video coding (HEVC) standard, versatile video coding (VVC) standard, have been proposed for video encoding/decoding. However, coding efficiency of video coding techniques is generally expected to be further improved.

SUMMARY

[0004] Embodiments of the present disclosure provide a solution for video processing.

[0005] In a first aspect, a method for video processing is proposed. The method comprises: determining, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and performing the conversion based on the reference template. The method in accordance with the first aspect of the present disclosure determines the reference template based on subblock level motion information. The determined reference template can thus be improved. In this way, the coding efficiency and coding effectiveness can be improved.

[0006] In a second aspect, an apparatus for video processing is proposed. The apparatus comprises a processor and a non-transitory memory with instructions thereon. The instructions upon execution by the processor, cause the processor to perform a method in accordance with the first aspect of the present disclosure.

[0007] In a third aspect, a non-transitory computer-readable storage medium is proposed. The non-transitory computer-readable storage medium stores instructions that cause a processor to perform a method in accordance with the first aspect of the present disclosure.

[0008] In a fourth aspect, another non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing. The method comprises: determining a reference template for a current video block of the video based on a current template of the current video block and motion information of at least one subblock of the current video block; and generating the bitstream based on the reference template.

[0009] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: determining a reference template for a current video block of the video based on a current template of the current video block and motion information of at least one subblock of the current video block; generating the bitstream based on the reference template; and storing the bitstream in a non-transitory computer-readable recording medium.

[0010] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Through the following detailed description with reference to the accompanying drawings, the above and other objectives, features, and advantages of example embodiments of the present disclosure will become more apparent. In the example embodiments of the present disclosure, the same reference numerals usually refer to the same components.

[0012] FIG. **1** illustrates a block diagram that illustrates an example video coding system, in accordance with some embodiments of the present disclosure;

[0013] FIG. **2** illustrates a block diagram that illustrates a first example video encoder, in accordance with some embodiments of the present disclosure;

[0014] FIG. **3** illustrates a block diagram that illustrates an example video decoder, in accordance with some embodiments of the present disclosure;

[0015] FIG. **4** illustrates positions of spatial and temporal neighboring blocks used in advanced motion vector prediction (AMVP) or merge candidate list constructure;

[0016] FIG. **5** illustrates an example diagram showing positions of non-adjacent candidate in ECM;

[0017] FIG. **6** illustrates an example diagram showing template matching performs on a search area around initial MV;

[0018] FIG. **7** illustrates an example diagram showing a template and the corresponding reference template;

[0019] FIG. **8** illustrates an example diagram showing template and reference template for block with sub-block motion using the motion information of the subblocks of current block;

[0020] FIG. **9** illustrates an example diagram showing an example of the positions for non-adjacent temporal motion vector prediction (TMVP) candidates;

[0021] FIG. **10** illustrates an example diagram showing an example of the template;

[0022] FIG. **11** illustrates an example of the template for a block with sub-block level motion information;

[0023] FIG. **12** illustrates a flowchart of a method for video processing in accordance with embodiments of the present disclosure; and

[0024] FIG. **13** illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0025] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0026] Principle of the present disclosure will now be described with reference to some embodiments. It is to be understood that these embodiments are described only for the purpose of illustration and help those skilled in the art to understand and implement the present disclosure, without suggesting any limitation as to the scope of the disclosure. The disclosure described herein can be implemented in various manners other than the ones described below.

[0027] In the following description and claims, unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skills in the art to which this disclosure belongs.

[0028] References in the present disclosure to “one embodiment,” “an embodiment,” “an example embodiment,” and the like indicate that the embodiment described may include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an example embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0029] It shall be understood that although the terms “first” and “second” etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term “and/or” includes any and all combinations of one or more of the listed terms.

[0030] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises”, “comprising”, “has”, “having”, “includes” and/or “including”, when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/or combinations thereof.

Example Environment

[0031] FIG. **1** is a block diagram that illustrates an example video coding system **100** that may utilize the techniques of this disclosure. As shown, the video coding system **100** may include a source device **110** and a destination device **120**. The source device **110** can be also referred to as a video encoding device, and the destination device **120** can be also referred to as a video decoding device. In operation, the source device **110** can be configured to generate encoded video data and the destination device **120** can be configured to decode the encoded video data generated by the source device **110**. The source device **110** may include a video source **112**, a video encoder **114**, and an input/output (I/O) interface **116**.

[0032] The video source **112** may include a source such as a video capture device. Examples of the video capture device include, but are not limited to, an interface to receive video data from a video content provider, a computer graphics system for generating video data, and/or a combination thereof.

[0033] The video data may comprise one or more pictures. The video encoder **114** encodes the video data from the video source **112** to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. The I/O interface **116** may include a modulator/demodulator and/or a transmitter. The encoded video data may be transmitted directly to destination device **120** via the I/O interface **116** through the network **130**A. The encoded video data may also be stored onto a storage medium/server **130**B for access by destination device **120**.

[0034] The destination device **120** may include an I/O interface **126**, a video decoder **124**, and a display device **122**. The I/O interface **126** may include a receiver and/or a modem. The I/O interface **126** may acquire encoded video data from the source device **110** or the storage medium/server **130**B. The video decoder **124** may decode the encoded video data. The display device **122** may display the decoded video data to a user. The display device **122** may be integrated with the destination device **120**, or may be external to the destination device **120** which is configured to interface with an external display device.

[0035] The video encoder **114** and the video decoder **124** may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video

Coding (VVC) standard and other current and/or further standards.

[0036] FIG. **2** is a block diagram illustrating an example of a video encoder **200**, which may be an example of the video encoder **114** in the system **100** illustrated in FIG. **1**, in accordance with some embodiments of the present disclosure.

[0037] The video encoder **200** may be configured to implement any or all of the techniques of this disclosure. In the example of FIG. **2**, the video encoder **200** includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video encoder **200**. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0038] In some embodiments, the video encoder **200** may include a partition unit **201**, a predication unit **202** which may include a mode select unit **203**, a motion estimation unit **204**, a motion compensation unit **205** and an intra-prediction unit **206**, a residual generation unit **207**, a transform unit **208**, a quantization unit **209**, an inverse quantization unit **210**, an inverse transform unit **211**, a reconstruction unit **212**, a buffer **213**, and an entropy encoding unit **214**.

[0039] In other examples, the video encoder **200** may include more, fewer, or different functional components. In an example, the predication unit **202** may include an intra block copy (IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0040] Furthermore, although some components, such as the motion estimation unit **204** and the motion compensation unit **205**, may be integrated, but are represented in the example of FIG. **2** separately for purposes of explanation.

[0041] The partition unit **201** may partition a picture into one or more video blocks. The video encoder **200** and the video decoder **300** may support various video block sizes.

[0042] The mode select unit **203** may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra-coded or inter-coded block to a residual generation unit **207** to generate residual block data and to a reconstruction unit **212** to reconstruct the encoded block for use as a reference picture. In some examples, the mode select unit **203** may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. The mode select unit **203** may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-predication.

[0043] To perform inter prediction on a current video block, the motion estimation unit **204** may generate motion information for the current video block by comparing one or more reference frames from buffer **213** to the current video block. The motion compensation unit **205** may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from the buffer **213** other than the picture associated with the current video block.

[0044] The motion estimation unit **204** and the motion compensation unit **205** may perform different operations for a current video block, for example, depending on whether the current video block is in an I-slice, a P-slice, or a B-slice. As used herein, an “I-slice” may refer to a portion of a picture composed of macroblocks, all of which are based upon macroblocks within the same picture. Further, as used herein, in some aspects, “P-slices” and “B-slices” may refer to portions of a picture composed of macroblocks that are not dependent on macroblocks in the same picture.

[0045] In some examples, the motion estimation unit **204** may perform uni-directional prediction for the current video block, and the motion estimation unit **204** may search reference pictures of list 0 or list 1 for a reference video block for the current video block. The motion estimation unit **204** may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. The motion estimation unit **204** may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. The motion compensation unit **205** may generate the predicted video block of the current video block based on the reference video block indicated by the motion information of the current video block.

[0046] Alternatively, in other examples, the motion estimation unit **204** may perform bi-directional prediction for the current video block. The motion estimation unit **204** may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. The motion estimation unit **204** may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. The motion estimation unit **204** may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. The motion compensation unit **205** may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[0047] In some examples, the motion estimation unit **204** may output a full set of motion information for decoding processing of a decoder. Alternatively, in some embodiments, the motion estimation unit **204** may signal the motion information of the current video block with reference to the motion information of another video block. For example, the motion estimation unit **204** may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0048] In one example, the motion estimation unit **204** may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder **300** that the current video block has the same motion information as the another video block.

[0049] In another example, the motion estimation unit **204** may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder **300** may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0050] As discussed above, video encoder **200** may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder **200** include advanced motion vector predication (AMVP) and merge mode signaling.

[0051] The intra prediction unit **206** may perform intra prediction on the current video block. When the intra prediction unit **206** performs intra prediction on the current video block, the intra prediction unit **206** may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0052] The residual generation unit **207** may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[0053] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and the residual generation unit **207** may not perform the subtracting operation.

[0054] The transform processing unit **208** may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0055] After the transform processing unit **208** generates a transform coefficient video block associated with the current video block, the quantization unit **209** may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0056] The inverse quantization unit **210** and the inverse transform unit **211** may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. The reconstruction unit **212** may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit **202** to produce a reconstructed video block associated with the current video block for storage in the buffer **213**.

[0057] After the reconstruction unit **212** reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0058] The entropy encoding unit **214** may receive data from other functional components of the video encoder **200**. When the entropy encoding unit **214** receives the data, the entropy encoding unit **214** may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0059] FIG. **3** is a block diagram illustrating an example of a video decoder **300**, which may be an example of the video decoder **124** in the system **100** illustrated in FIG. **1**, in accordance with some embodiments of the present disclosure.

[0060] The video decoder **300** may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. **3**, the video decoder **300** includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder **300**. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0061] In the example of FIG. **3**, the video decoder **300** includes an entropy decoding unit **301**, a motion compensation unit **302**, an intra prediction unit **303**, an inverse quantization unit **304**, an inverse transformation unit **305**, and a reconstruction unit **306** and a buffer **307**. The video decoder **300** may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder **200**.

[0062] The entropy decoding unit **301** may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). The entropy decoding unit **301** may decode the entropy coded video data, and from the entropy decoded video data, the motion compensation unit **302** may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. The motion compensation unit **302** may, for example, determine such information by performing the AMVP and merge mode. AMVP is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. Motion information typically includes the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. As used herein, in some aspects, a “merge mode” may refer to deriving the motion information from spatially or temporally neighboring blocks.

[0063] The motion compensation unit **302** may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0064] The motion compensation unit **302** may use the interpolation filters as used by the video encoder **200** during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit **302** may determine the interpolation filters used by the video encoder **200** according to the received syntax information and use the interpolation filters to produce predictive blocks.

[0065] The motion compensation unit **302** may use at least part of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence. As used herein, in some aspects, a “slice” may refer to a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture.

[0066] The intra prediction unit **303** may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. The inverse quantization unit **304** inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit **301**. The inverse transform unit **305** applies an inverse transform.

[0067] The reconstruction unit **306** may obtain the decoded blocks, e.g., by summing the residual blocks with the corresponding prediction blocks generated by the motion compensation unit **302** or intra-prediction unit **303**. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the buffer **307**, which provides reference blocks for subsequent motion compensation/intra predication and also produces decoded video for presentation on a display device.

[0068] Some exemplary embodiments of the present disclosure will be described in detailed hereinafter. It should be understood that section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Brief Summary

This disclosure is related to video coding technologies. Specifically, it is about motion vector prediction (MVP) construction method in video coding. The ideas may be applied individually or in various combination, to any video coding standard or non-standard video codec.

2. Introduction

[0069] The exponential increasing of multimedia data poses a critical challenge for video coding. To satisfy the increasing demands for more efficient compression technology, ITU-T and ISO/IEC have developed a series of video coding standards in the past decades. In particular, the ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 visual, and the two organizations jointly developed the H.262/MPEG-2 Video, H.264/MPEG-4 Advanced Video Coding (AVC), H.265/HEVC and the latest VVC standards. Since H.262/MPEG-2, hybrid video coding framework is employed wherein in intra/inter prediction plus transform coding are utilized.

[0070] FIG. **4** illustrates a diagram **400** showing positions of spatial and temporal neighboring blocks used in AMVP/merge candidate list construction.

2.1. MVP in Video Coding

Inter prediction aims to remove the temporal redundancy between adjacent frames, which serves as an indispensable component in the hybrid video coding framework. Specifically, inter prediction makes use of the contents specified by motion vector (MV) as the predicted version of the current to-be-coded block, thus only residual signals and motion information are transmitted in the bitstream. To reduce the cost for MV signaling, motion vector prediction (MVP) came into being as an effective mechanism to convey motion information. Early strategies simply use the MV of a specified neighboring block or the median MV of neighboring blocks as MVP. In H.265/HEVC, competing mechanism was involved where the optimal MVP is selected from multiple candidates through rate distortion optimization (RDO). In particular, advanced MVP (AMVP) mode and merge mode are devised with different motion information signaling strategy. With the AMVP mode, a reference index, an MVP candidate index referring to an AMVP candidate list and motion vector difference (MVD) is signaled. Regarding the merge mode, only a merge index referring to a merge candidate list is signaled, and all the motion information associated with the merge candidate is inherited. Both AMVP mode and merge mode need to construct MVP candidate list, and the details of the construction process for these two modes are described as follows.

AMVP mode: AMVP exploits spatial-temporal correlation of motion vector with neighboring blocks, which is used for explicit transmission of motion parameters. For each reference picture list, a motion vector candidate list is constructed by firstly checking availability of left, above temporally neighboring positions, removing redundant candidates and adding zero vector to make the candidate list to be constant length. For spatial motion vector candidate derivation, two motion vector candidates are eventually derived based on motion vectors of blocks located in five different positions as depicted in FIG. **4**. The five neighboring blocks located at B**0**, B**1**, B**2**, and A**0**, A**1** are classified into two groups, where Group A includes the three above spatial neighboring blocks and Group B includes the two left spatial neighboring blocks. The two MV candidates are respectively derived with the first available candidate from Group A and Group B in a predefined order. For temporal motion vector candidate derivation, one motion vector candidate is derived based on two different co-located positions (bottom-right (C**0**) and central (C**1**)) checked in order, as depicted in FIG. **4**. To avoid redundant MV candidates, duplicated motion vector candidates in the list are abandoned. If the number of potential candidates is smaller than two, additional zero motion vector candidates are added to the list.

FIG. **5** illustrates a diagram **500** of positions of non-adjacent candidate in ECM.

[0071] Merge mode: Similar to AMVP mode, MVP candidate list for merge mode comprises of spatial and temporal candidates as well. For spatial motion vector candidate derivation, at most four candidates are selected with order A**1**, B**1**, B**0**, A**0** and B**2** after performing availability and redundant checking. For temporal merge candidate (TMVP) derivation, at most one candidate is selected from two temporal neighboring blocks (C**0** and C**1**). When there are not enough merge candidates with spatial and temporal candidates, combined bi-predictive merge candidates and zero MV candidates are added to MVP candidate list. Once the number of available merge candidates reaches the signaled maximally allowed number, the merge candidate list construction process is terminated.

[0072] In VVC, the construction process for merge mode is further improved by introducing the history-based MVP (HMVP), which incorporates the motion information of previously coded blocks which may be far away from current block. In VVC, HMVP merge candidates are appended to merge list after the spatial MVP and TMVP. In this method, the motion information of a previously coded block is stored in a table and used as MVP for the current CU. The table with multiple HMVP candidates is maintained with first-in-first-out strategy during the encoding/decoding process. Whenever there is a non-subblock inter-coded CU, the associated motion information is added to the last entry of the table as a new HMVP candidate.

[0073] During the standardization of VVC, Non-adjacent MVP was proposed to facilitate better motion information derivation by exploiting the non-adjacent area. In ECM software, Non-adjacent MVP are inserted between TMVP and HMVP, where the distances between non-adjacent spatial candidates and current coding block are based on the width and height of current coding block as depicted in FIG. **5**.

2.2. Interpolation Filters in VVC

[0074] In VVC, interpolations filters are used in both intra and inter coding process. Intra coding takes advantage of interpolation filters to generate fractional positions in angular prediction modes. In HEVC, a two-tap linear interpolation filter has been used to generate the intra prediction block in the directional prediction modes (i.e., excluding Planar and DC predictors). While in VVC, four-tap intra interpolation filters are utilized to improve the angular intra prediction accuracy. In particular, two sets of 4-tap interpolation filters are utilized in VVC intra coding, which are DCT-based interpolation filter (DCTIF) and smoothing interpolation filter (SIF). The DCTIF is constructed in the same way as the one used for chroma component motion compensation in both HEVC and VVC. The SIF is obtained by convolving the 2-tap linear interpolation filter with [1 2 1]/4 filter.

[0075] In VVC, the highest precision of explicitly signaled motion vectors is quarter-luma-sample. In some inter prediction modes such as the affine mode, motion vectors are derived at 1/16th-luma-sample precision and motion compensated prediction is performed at 1/16th-sample-precision. VVC allows different MVD precision ranging from 1/16-luma-sample to 4-luma-sample. For half-luma-sample precision, 6-tap interpolation filter is used. While for other fractional precisions, default 8-tap filter is used. Besides, the bilinear interpolation filter is used to generate the fractional samples for the searching process of decoder side motion vector refinement (DMVR) in VVC.

2.3. Template Matching Merge/AMVP Mode in ECM

[0076] Template matching (TM) merge/AMVP mode is a decoder-side MV derivation method to refine the motion information of the current CU by finding the closest match between a template (i.e., top and/or left neighboring blocks of the current CU) in the current picture and a block (i.e., same size to the template) in a reference picture. As illustrated in FIG. **6**, a better MV is to be searched around the initial motion of the current CU within a [−8, +8]-pel search range.

FIG. **6** illustrates a diagram **600** of template matching performs on a search area around initial MV.

[0077] In AMVP mode, an MVP candidate is determined based on the template matching error to pick up the one which reaches the minimum difference between the current block and the reference block templates, and then TM performs only for this particular MVP candidate for MV refinement. TM refines this MVP candidate, starting from full-pel MVD precision (or 4-pel for 4-pel AMVR mode) within a [−8, +8]-pel search range by using iterative diamond search. The AMVP candidate may be further refined by using cross search with full-pel MVD precision (or 4-pel for 4-pel AMVR mode), followed sequentially by half-pel and quarter-pel ones depending on AMVR mode. This search process ensures that the MVP candidate still keeps the same MV precision as indicated by adaptive motion vector resolution (AMVR) mode after TM process.

[0078] In the merge mode, similar search method is applied to the merge candidate indicated by the merge index. TM merge may perform all the way down to ⅛-pel MVD precision or skipping those beyond half-pel MVD precision, depending on whether the alternative interpolation filter (that is used when AMVR is of half-pel mode) is used according to merged motion information. Besides, when TM mode is enabled, template matching may work as an independent process or an extra MV refinement process between block-based and subblock-based bilateral matching (BM) methods, depending on whether BM can be enabled or not according to its enabling condition check. When BM and TM are both enabled for a CU, the search process of TM stops at half-pel MVD precision and the resulted MVs are further refined by using the same model-based MVD derivation method as in DMVR.

2.4. Adaptive Reorder of Merge Candidates (ARMC)

[0079] Inspired by the spatial correlation between reconstructed neighboring pixels and the current coding block, adaptive reorder of merge candidates (ARMC) was proposed to refine the candidates order in a given candidate list. The underlying assumption is that the candidates with less template matching cost have higher probability to be chosen through RDO process, hence should be placed in front positions within the list to reduce the signaling cost.

[0080] The reordering method is applied to regular merge mode, template matching (TM) merge mode, and affine merge mode (excluding the SbTMVP candidate). For the TM merge mode, merge candidates are reordered before the refinement process.

[0081] After a merge candidate list is constructed, merge candidates are divided into several subgroups. The subgroup size is set to 5. Merge candidates in each subgroup are reordered ascendingly according to cost values based on template matching. For simplification, merge candidates in the last but not the first subgroup are not reordered.

[0082] The template matching cost is measured by the sum of absolute differences (SAD) between samples of a template of the current block and their corresponding reference template. The template comprises a set of reconstructed samples neighboring to the current block, while reference template is located by the same motion information of the current block, as illustrated in FIG. **7**. When a merge candidate utilizes bi-directional prediction, the reference samples of the template of the merge candidate are also generated by bi-prediction.

FIG. **7** illustrates a diagram **700** of template and the corresponding reference template.

FIG. **8** illustrates a diagram **800** of template and reference template for block with sub-block motion using the motion information of the subblocks of current block.

[0083] For subblock-based merge candidates with subblock size equal to Wsub*Hsub, the above template comprises several sub-templates with the size of Wsub×1, and the left template comprises several sub-templates with the size of 1×Hsub. As shown in FIG. **8**. the motion information of the subblocks in the first row and the first column of current block is used to derive the reference samples of each sub-template.

FIG. **9** illustrates an example diagram **900** of the positions for non-adjacent TMVP candidates.

2.5. Enhanced MVP Candidate Derivation (EMCD)

[0084] EMCD based on template matching cost reordering has been proposed. Instead of constructing the MVP list based on a predefined traversing order, we investigate an optimized MVP selecting approach by taking advantage of the matching cost in the reconstructed template region, such that more appropriate candidates are included in the list.

[0085] It should be noted that the proposed strategy for MVP list construction can be utilized in normal merge and AMVP list construction process and can also be easily extended to other modules that require MVP derivation, e.g., merge with motion vector difference (MMVD), Affine motion compensation, Subblock-based temporal motion vector prediction (SbTMVP) and so on.

Non-Adjacent TMVP

[0086] 1. It is proposed to make use of the TMVP in a non-adjacent area to further improve the effectiveness of the MVP list. [0087] a) In one example, a non-adjacent area may be any block (such as 4×4 block) in a reference picture and neither inside nor adjacent to the collocated block in the reference picture of the current block. [0088] b) In one example, the positions of the non-adjacent TMVP candidates are illustrated in FIG. **9**, where black blocks represent the potential non-adjacent TMVP positions. It should be noted that this figure only provides an example for non-adjacent TMVP, and the positions are not limited to the indicated blocks. In other cases, non-adjacent TMVP may locate in any other positions in one or more reconstructed frames. [0089] 2. The maximum allowed non-adjacent TMVP number in the MVP list may be signaled in the bitstream. [0090] a) In one example, the maximum allowed number can be signaled in SPS or PPS. [0091] 3. The non-adjacent TMVP candidates may locate in the nearest reconstructed frame, but it may also locate in other reconstructed frames. [0092] a) Alternatively, non-adjacent TMVP candidates may locate in the collocated picture. [0093] b) Alternatively, it is signaled in which picture non-adjacent TMVP candidates may locate. [0094] 4. Non-adjacent TMVP candidates may locate in multiple reference pictures. [0095] FIG. **10** illustrates an example diagram **1000** of the template. [0096] 5. The distances between a non-adjacent area associated with a TMVP candidate and current coding block may be related to the property of the current block. [0097] a) In one example, the distances depend on the width and height of current coding block. [0098] b) In other cases, the distances may be signaled in the bitstream as a constant.

Definition of the Template

[0099] 6. Template represents the reconstructed region that can be used to estimate the priority of an MVP candidate, which may locate in different positions with variable shape. [0100] a) In one example, a template may comprise of the reconstructed regions in three positions, which are upper pixels, left pixels and upper-left pixels, as presented in FIG. **10**. [0101] b) It should be noted that the template may not necessarily be in rectangular shape, it can be in arbitrary shape, e.g., triangle or polygon. [0102] c) In one example, the template regions may be utilized either in separate or combined manner. [0103] d) A template may only comprise samples from one component such as luma, or from multiple components such as luma and chroma. [0104] 7. The template may not necessarily locate in the current frame, it may locate in any other reconstructed frame. [0105] 8. In one example, a reference template region with the same shape as the template of the current block may be located with an MV, as shown in FIG. **7**. [0106] 9. In one example, the template may not necessarily locate in adjacent area, it may locate in non-adjacent areas that are far away from the current block. [0107] 10. In one example, a template may not necessarily contain all the pixels in a certain region, it may contain part of the pixels in a region.

Template Matching Based MVP Candidate Ordering

[0108] 11. In this disclosure, template matching cost associated with a certain MVP candidate serves as a measurement to evaluate the consistency of this candidate and true motion information. Based on this measurement, a more efficient order is generated by sorting the priority of each MVP candidate. [0109] a) In one example, the template matching cost C is evaluated with mean of square error (MSE), as calculated below:

[00001]$C = \frac{{}_{(i,j) \in T}(T(i,j) - \mathrm{RT}(i,j))^2}{N}$, [0110] where T represents the template region, RT represents the corresponding reference template region specified by the MV within MVP candidate (FIG. **7**), N is the pixel number within the template. [0111] b) In one example, the template matching cost can be evaluated with sum of square error (SSE), sum of absolute difference (SAD), sum of absolute transformed difference (SATD) or any other criterion that can measure the difference between two regions. [0112] 12. All the MVP candidates are sorted in an ascending order regarding the corresponding template matching cost, and the MVP list is constructed by traversing the candidates in the sorted order until the MVP amount reaches the maximum allowed number. In this way, a candidate with a lower matching cost has a higher priority to be included in the ultimate MVP list. [0113] a) In one example, the sorting process may be conducted towards all the MVP candidates. [0114] b) Alternatively, this process may also be applied to part of candidates, e.g., non-adjacent MVP candidates, HMVP candidates or any other group of candidates. [0115] c) Alternatively, furthermore, which categories of MVP candidates (e.g., non-adjacent MVP candidates are belonging to one category, HMVP candidates are belonging to another category) and/or what kinds of group of candidates should be reordered may be dependent on the decoded information, e.g., block dimension/coding methods (e.g., CIIP/MMVD) and/or how many available MVP candidates before being reordered for a given kind/group. [0116] 1. In one example, the sorting process may be conducted for a joint group which contains only one category of MVP candidates. [0117] 2. In one example, the sorting process may be conducted for a joint group which contains more than one category of MVP candidates. [0118] a) In one example, for a first coding method (e.g., regular/CIIP/MMVD/GPM/TPM/subblock merge mode), the sorting process can be conducted for a joint group of non-adjacent MVP, non-adjacent TMVP and HMVP candidates. For a second coding method (e.g., the template matching merge mode), the sorting process can be conducted for a joint group of adjacent MVP, non-adjacent TMVP, non-adjacent MVP and HMVP candidates. [0119] b) Alternatively, for a first coding method (e.g., regular/CIIP/MMVD/GPM/TPM/subblock merge mode), the sorting process can be conducted for a joint group of non-adjacent MVP and HMVP candidates. For a second coding method (e.g., the template matching merge mode), the sorting process can be conducted for a joint group of adjacent MVP, non-adjacent MVP and HMVP candidates. [0120] 3. In one example, the sorting process may be conducted for a joint group which contains partial of available MVP candidates within the categories. [0121] a) In one example, for regular/CIIP/MMVD/TM/GPM/TPM/subblock merge mode, or for regular/affine AMVP mode, the

sorting process can be conducted for a joint group of all or partial candidates from one or multiple categories. [0122] 4. In above examples, the category may be [0123] i. adjacent neighboring MVPs; [0124] ii. adjacent neighboring MVPs at specific location(s); [0125] iii. TMVP MVPs; [0126] iv. HMVP MVPs; [0127] v. Non-adjacent MVPs; [0128] vi. Constructed MVPs (such as pairwise MVPs); [0129] vii. Inherited affine MV candidates; [0130] viii. Constructed affine MV candidates; [0131] ix. SbTMVP candidates. [0132] d) In one example, this process may be conducted multiple times on different set of candidates. [0133] 1. For example, a set of candidates (such as non-adjacent MVP candidates) may be sorted, and the N non-adjacent MVP candidates with the lowest costs may be put into the candidate list. After the whole candidate list is constructed, the costs of candidates in the list may be calculated and the candidates may be reordered based on the costs. [0134] 13. It is proposed that the MVP list construction process may involve both reordering of a single group/category and a joint group which contains candidates from more than one category. [0135] a) In one example, the joint group may include candidates from a first and a second category. [0136] 1. Alternatively, furthermore, the first and second category may be defined as the non-adjacent MVP category and HMVP category. [0137] 2. Alternatively, furthermore, the first and second category may be defined as the non-adjacent MVP category and HMVP category, and the joint group may include candidates from a third category, e.g., TMVP category. [0138] b) In one example, the single group may include candidates from a fourth category. [0139] 1. Alternatively, furthermore, the fourth category may be defined as the adjacent MVP category. [0140] 14. Multiple groups or categories can be respectively reordered to construct MVP list. [0141] a) In one example, only one single group (all the candidates belong to one category, e.g. adjacent MVP, non-adjacent MVP, HMVP, etc.) is built and reordered in MVP list construction process. [0142] b) In one example, only one joint group (contains partial or all the candidates from multiple categories) is built and reordered in MVP list construction process. [0143] c) In one example, more than one group (regardless of single or joint) are respectively built and reordered in MVP list construction process. [0144] 1. In one example, two or more single groups are respectively built and reordered in MVP list construction process. [0145] 2. In one example, two or more joint groups are respectively built and reordered in MVP list construction process. [0146] 3. In one example, one or multiple single groups and one or multiple joint groups are respectively reordered in MVP list construction process. [0147] a) In one example, one single groups and one joint groups are respectively built and reordered to construct MVP list. [0148] b) In one example, one single groups and multiple joint groups are respectively built and reordered to construct MVP list. [0149] c) In one example, multiple single groups and one joint groups are respectively built and reordered to construct MVP list. [0150] d) In one example, multiple single groups and multiple joint groups are respectively built and reordered to construct MVP list. [0151] d) In one example, candidates that belong to the same category can be divided into different groups, and are respectively reordered in the corresponding groups. [0152] e) In one example, only partial candidates in specific category are put into the single or joint group, and rest candidates in this category are not reordered. [0153] f) In above examples, the category may be [0154] 1. adjacent neighboring MVPs; [0155] 2. adjacent neighboring MVPs at specific location(s); [0156] 3. TMVP MVPs; [0157] 4. HMVP MVPs; [0158] 5. Non-adjacent MVPs; [0159] 6. Constructed MVPs (such as pairwise MVPs); [0160] 7. Inherited affine MV candidates; [0161] 8. Constructed affine MV candidates; [0162] 9. SbTMVP candidates. [0163] 15. The proposed sorting method can also be applied to AMVP mode. [0164] a) In one example, the MVP in AMVP mode can be extended with non-adjacent MVP, non-adjacent TMVP and HMVP. [0165] b) In one example, MVP list for AMVP mode comprises K candidates, which are selected from M categories, such as adjacent MVPs, non-adjacent MVPs, non-adjacent TMVPs and HMVPs wherein K and M are integers. [0166] 1. In one example, K could be smaller than M, or equal to M or greater than M. [0167] 2. In one example, one candidate is selected from each category. [0168] 3. Alternatively, for a given category, no candidate is selected. [0169] 4. Alternatively, for a given category, more than 1 candidate is selected. [0170] 5. In one example, MVP list for AMVP mode comprises 4 candidates, which are selected from adjacent MVPs, non-adjacent MVPs, non-adjacent TMVPs and HMVPs. [0171] 6. In one example, each category of MVP candidates is respectively sorted with template matching cost, and the one with minimum cost in the corresponding category is selected and included in the MVP list. [0172] 7. Alternatively, adjacent MVP candidates and a joint group of non-adjacent MVP, non-adjacent TMVP together with HMVP candidates are respectively sorted with template matching cost. One adjacent candidate with the minimum template matching cost is selected from adjacent MVP candidates, and three other candidates are derived by traversing the candidates in the joint group in an ascending order of template matching cost. [0173] 8. In one example, MVP list for AMVP mode comprises 2 candidates, one comes from adjacent MVP and the other comes from non-adjacent MVP, non-adjacent TMVP or HMVP. In particular, adjacent MVP candidates and a joint group of non-adjacent MVP, non-adjacent TMVP together with HMVP are respectively sorted with template matching cost, and the one with minimum cost in the corresponding category (or group) is included in the MVP list. [0174] 16. The proposed sorting methods may be applied to other coding methods, e.g., for constructing a block vector list of IBC coded blocks. [0175] a) In one example, it may be used for affine coded blocks. [0176] b) Alternatively, furthermore, how to define the template cost may be dependent on the coding methods. [0177] 17. The usage of this method may be controlled with different coding level syntax, including but not limit to one or multiple of PU, CU, CTU, slice, picture, sequence levels. [0178] 18. On how to insert sorted candidates to MVP list. [0179] a) In one example, which candidates within the joint or separate group are included into MVP list depends on the sorting results of template matching cost. [0180] b) In one example, whether put the candidates within the separate or joint group into MVP list depends on the sorting results of template matching cost. [0181] c) In one example, how many candidates within the separate or joint group are included into MVP list depends on the sorting results of template matching cost. [0182] 1. In one example, only one candidate with the smallest template matching cost is included into MVP list. [0183] 2. In one group, top-N candidates regarding the template matching cost in an ascending order are included into MVP list, where N is the maximum allowed candidate number can be inserted into MVP list in the corresponding single or joint group. [0184] a) In one example, N can be a predefined constant for each single or joint group. [0185] b) Alternatively, N can be adaptively derived based on the template matching cost within the single or joint group. [0186] c) Alternatively, N can be signaled in the bitstream. [0187] d) In one example, different candidate groups share a same N value. [0188] e) Alternatively, different single or joint groups may have different N value.

Pruning for MVP Candidates

[0189] 19. The pruning for MVP candidates aims to increase the diversity within the MVP list, which can be realized by using appropriate threshold TH. [0190] a) In one example, if the two candidates point to same reference frame, they may both be included to MVP list only if the absolute difference between the corresponding X and Y components are either or both larger (or no smaller) than TH. [0191] 20. The pruning threshold can be signaled in the bitstream. [0192] b) In one example, the pruning threshold can be signaled either in PU, CU, CTU or slice level. [0193] 21. The pruning threshold may depend on the characteristics of the current block. [0194] c) In one example, the threshold may be derived by analyzing the diversity among the candidates. [0195] d) In one example, the optimal threshold can be derived through RDO. [0196] 22. The pruning for MVP candidates may be firstly performed within a single or joint group before being sorted. [0197] a) Alternatively, furthermore, for two MVP candidates belonging to two different groups or one belonging to a joint group and the other doesn't, pruning among these two MVP candidates are not performed before sorting. [0198] b) Alternatively, furthermore, pruning among multiple groups may be applied after the sorting. [0199] 23. The pruning for MVP candidates may be firstly performed among multiple groups and the sorting may be further applied to one or multiple single/joint groups. [0200] a) Alternatively, an MVP list may be firstly constructed with pruning among available MVP candidates involved. Afterwards, sorting may be further applied to reorder one or multiple single/joint groups. [0201] b) Alternatively, furthermore, for two MVP candidates belonging to two different groups or one belonging to a joint group and the other doesn't, pruning among these two MVP candidates is performed before sorting.

Interaction with Other Coding Tools [0202] 24. After an MVP list with above sorting methods applied, the Adaptive Reordering Merge Candidates (ARMC) process may be further applied. [0203] a) In one example, the template costs used in the sorting process during MVP list construction may be further utilized in the ARMC. [0204] b) In another example, different template costs may be used in the sorting process and ARMC process. [0205] 1. In one example, the template may be different for the sorting and ARMC process. [0206] 25. Whether to and/how to enable the sorting process may be dependent on the coding tool. [0207] a) In one example, when a certain tool (e.g., MMVD or affine mode) is enabled for a block, the sorting is disabled. [0208] b) In one example, for two different tools, the sorting rules may be different (e.g., being applied to different groups or different template settings).

2.6. Simplifications for Template Matching Based Video Coding Methods

[0209] The template matching based video coding methods is optimized in two aspects. Firstly, reference template derivation process is revised that the interpolation process in the prediction block generation process is replaced by different ways. Secondly, several fast strategies are devised to speedup the tools related to template matching.

[0210] It should be noted that the proposed methods can be utilized in ARMC, EMCD and template matching MV refinement, and can also be easily extended to other potential utilizations that require template matching process, e.g., template matching based candidates reorder for merge with motion vector difference (MMVD), Affine motion compensation, Subblock-based temporal motion vector prediction (SbTMVP) and so on. In yet another example, the proposed methods could be applied to other coding tools that requires motion information refinement processes, e.g., bilateral matching-based coding tools.

[0211] The detailed embodiments below should be considered as examples to explain general concepts. These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner. Combination between this patent application and others are also applicable. [0212] 1. It is proposed to replace the interpolation filtering process involved in the motion compensation process of an inter prediction signal generation process by other ways in the reference template generation process. [0213] a) It is proposed to exclude interpolation filtering process to generate a reference template even the motion vector point to fractional positions. [0214] i. In one example, it is proposed to use an integer precision to generate a reference template. [0215] ii. In one example, if a motion vector points to a fractional position, it is rounded to be an integer MV firstly. [0216] 1. In one example, the fractional position is rounded toward zero (that is, a negative motion vector predictor is rounded toward positive infinity and a positive motion vector predictor is rounded toward negative infinity). [0217] 2. In one example, the round step may larger than 1. [0218] b) It is proposed to use a different interpolation filter to generate reference templates for motion vectors pointing to fractional positions. [0219] i. In one example, a simplified interpolation filter may be applied. [0220] 1. In one example, the simplified interpolation filter can be 2-tap bilinear, alternatively, it can also be 4-tap, 6-tap or 8-tap filter that belongs to DCT, DST, Lanczos or any other interpolation types. [0221] ii. In one example, a more complex interpolation filter (e.g., with longer filter taps) may be applied. [0222] c) The above methods may be used to reorder the merge candidates for template matching merge mode. [0223] i. In one example, integer precision can be used in ARMC, EMCD, LIC and any other potential scenarios. [0224] ii. The above methods may be used to reorder the candidates for regular merge mode. [0225] 1. In one example, integer precision can be used to reorder the candidates for regular merge mode. [0226] d) In one example, whether to use above methods (e.g., integer precision, different interpolation filters) or not and/or how to use above methods can be signaled in the bitstream or determined on-the-fly according to decoded information. [0227] i. In one example, which method to be applied may be dependent on the coding tool. [0228] ii. In one example, which method to be applied may be dependent on block dimension. [0229] iii. In one example, integer precision may be used for a given color component (e.g., luma only). [0230] iv. Alternatively, integer precision may be used all of the three components. [0231] 2. Whether to and/or how to perform EMCD may be based on the maximum allowed candidate number within candidate list and/or available candidate number before being added to a candidate list. [0232] a) In one example, assuming the number of available candidates (valid candidates that can be used to build candidate list) is NAVAL, and the maximum allowed candidate number is NMAX (that is, at most NMAX candidates can be included into the ultimate merge list), then EMCD is enabled only when NAVAL-NMAX larger than a constant or adaptively derived threshold T. [0233] 3. It is proposed to organize the available merge candidates into subgroups. [0234] a) In one example, the available candidates can be categorized into subgroups, each subgroup contains a fixed or adaptively derived number of candidates, and each subgroup selects a fix number of candidates into the list. In the decoder side, only the candidates within a chosen subgroup need to be reordered. [0235] b) In one example, the candidates can be categorized into subgroups according to the candidates' category, such as non-adjacent MVP, temporal MVP (TMVP) or HMVP, etc. [0236] 4. It is proposed that a piece of information calculated by a first coding tool utilizing at least one template cost may be reused by a second coding tool utilizing at least one template cost. [0237] a) It is proposed to build a unified storage shared by ARMC, EMCD and any other potential tools to store the information of each merge candidate. [0238] b) In one example, this storage can be a map, table or other data structure. [0239] c) In one example, the stored information can be template matching cost. [0240] d) In one example, EMCD first traverses all the MVs associated with the available candidates and store the corresponding information (including but not limited to template matching cost) in this storage. Then ARMC and/or other potential tools can simply access the needed information from this shared storage without performing repeating calculation.

2.7. Extensions of Motion Vector Prediction List Construction Based on Template Matching Cost Ordering

[0241] An optimized MVP list derivation method based on template matching cost ordering is proposed. Instead of constructing the MVP list based on a predefined traversing order, we investigate an optimized MVP selecting approach by taking advantage of the matching cost in the reconstructed template region, such that more appropriate candidates are included in the list.

[0242] It should be noted that the proposed strategy for MVP list construction can be utilized in normal merge and AMVP list construction process and can also be easily extended to other modules that require MVP derivation, e.g., merge with motion vector difference (MMVD), Affine motion compensation, Subblock-based temporal motion vector prediction (SbTMVP) and so on.

[0243] In the following discussion, category represents the belongingness of an MVP candidate, e.g., non-adjacent MVP candidates belong to one category, HMVP candidates belonging to another category. A group denotes an MVP candidate set which contains one or multiple MVP candidates. In one example, a single group denotes an MVP candidate set in which all the candidates belong to one category, e.g., adjacent MVP, non-adjacent MVP, HMVP, etc. In another example, a joint group denotes an MVP candidate set which contains candidates from multiple categories.

[0244] The detailed embodiments below should be considered as examples to explain general concepts. These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner. Combination between this patent application and others are also applicable. [0245] 5. Multiple thresholds to determine whether a candidate could be added to a candidate list may be utilized in the candidate pruning process. [0246] a) A threshold may be used to determine whether a potential candidate can be put into a candidate list. [0247] i. For example, if the absolute difference of at least one component of the MV of the potential candidate and that of a candidate existing in the candidate list is smaller than a threshold, the potential candidate is not put into the list. [0248] ii. For example, if the absolute difference of all components of the MV of the potential candidate and that of a candidate existing in the candidate list is smaller than a threshold, the potential candidate is not put into the list. [0249] b) In one example, the candidate is an MVP candidate, the candidate pruning process is the MVP candidate pruning process, and the candidate list is a motion candidates list. [0250] i. In one example, the motion candidate list is a merge candidate list. [0251] ii. In one example, the motion candidate list is an AMVP candidate list. [0252] iii. In one example, the motion candidate list is an extend merge or AMVP list, such as sub-block merge candidate list, affine merge candidate list, MMVD list, GPM list, template matching merge list, biliteral matching merge list etc. [0253] c) In one example, the pruning thresholds may be different for two groups, where the group can be either a single group (containing only one category of candidates) or a joint group (containing at least two categories of candidates). [0254] d) Alternatively, only one threshold is used for all potential MVP candidates regardless of category and/or groups. [0255] e) In one example, N (e.g., N=2) thresholds are used in the pruning process. [0256] i. Assume A is the MVP set which contains all available MVP candidates regardless of category, in one example, a first threshold is used for a first subset of candidates in set A, and a second threshold is used for a second subset of candidates (e.g. the rest candidates excluding those in the first subset) in set A. [0257] ii. In one example, a first threshold is used for a single group denoted by A, and a second threshold is used for another group (single or joint)/multiple other groups/rest of candidates which are not with the same category as those in A. [0258] 1. In one example, a first threshold is used for the single group of adjacent candidates, and a second threshold is used for the rest candidates, including but not limited to non-adjacent MVP, HMVP, pairwise MVP and zero MVP. [0259] iii. The first threshold may be larger than or smaller than the second threshold. [0260] f) Alternatively, furthermore, the threshold for an MVP category or group may be dependent on the decoded information, e.g., block dimension/coding methods (e.g., CIIP/MMVD) and/or the variance of motion information within the category or group. [0261] 6. Multi-pass reordering can be performed to construct an MVP list. [0262] a) In one example, the multi-pass may involve different reordering criteria. [0263] b) In one example, multi-pass reordering can be performed to multiple single/joint groups, wherein at least two single/joint groups may have overlap MVP candidates or not. [0264] c) In one example, K-pass (e.g., K=2) reordering is used to construct an MVP list. [0265] i. In one example, in the first pass, a single/joint group A is firstly reordered based on a first cost (e.g. template matching cost) sorting, and the candidate with the largest cost (CL) in A is identified and then transferred to another single/joint group B (e.g. B may comprise the rest of candidates which are not with the same category as those in A). Subsequently, group B conduct the 2 to K pass reorder based on the first cost (or other cost metrics) sorting. Finally, the candidates in group A (except CL) and B (CL included) are included in the MVP list in accordance with the sorted order. [0266] ii. In one example, the group A in above case is a single group of adjacent candidates, and group B is a joint group of non-adjacent candidates and HMVP. [0267] iii. Alternatively, group A and B may be any other single or joint candidate group. [0268] iv. In one example, in the first pass, one or multiple single/joint groups are firstly reordered based on a first cost (e.g. template matching cost) sorting. Then a preliminary MVP list is constructed by inserting some of the candidates in each group into the list with the sorted order. Subsequently, the preliminary MVP list performs the second pass reorder to select partial candidates into the ultimate MVP list. [0269] 1. In one example, different single/joint groups may have overlap candidates or not. [0270] 2. In one example, all of the candidates in the preliminary MVP list are selected from the sorted single/joint groups. [0271] 3. Alternatively, partial candidates in the preliminary MVP list are selected from the sorted groups, and the rest candidates are included into the list with other rules. [0272] 4. In one example, in the second pass, all the candidates in the preliminary list, regardless of the corresponding categories, are sorted based on a cost (e.g. template matching cost), and only limited number of candidates are included into the ultimate MVP list based on the sorted order. a) Alternatively, furthermore, all the candidates in the preliminary MVP list are included in the ultimate MVP list in accordance with the sorted order. [0273] 5. The cost (e.g. template matching cost) calculated in a former pass can be re-used in a later pass. a) In one example, when the cost for a certain candidate is calculated in a former pass, it will be saved in a variable or any other data structure in case the same cost is needed in a later pass. b) In one example, in a later pass, if the cost for a certain candidate is needed, it will first check whether this cost has been calculated before or not. If this cost has been calculated and/or saved before, and/or is accessible in the current pass, it will be fetched in the current pass instead of calculating again. [0274] 7. At least one virtual candidate (e.g., pairwise MVP and zero MVP) may be involved in the at least one group. [0275] a) In one example, all the virtual candidates are treated with one joint group. [0276] i. Alternatively, each category of virtual candidates is treated as a single group. [0277] ii. In one example, the pairwise MVP and/or zero MVP are included in a single/joint group. [0278] iii. Alternatively, furthermore, the group which contains the virtual candidates is reordered and then put into a candidate list. [0279] b) Alternatively, the virtual candidates (e.g., pairwise MVP and/or zero MVP) are not included in any single/joint group. [0280] i. Alternatively, furthermore, no reordering process is applied to virtual candidates. [0281] 1. Alternatively, furthermore, they may be further appended to candidate list. [0282] ii. In one example, one or more single/joint groups are constructed, where partial or all of the groups are reordered. In this case, at least one position in MVP list is preserved for the virtual candidates (e.g., pairwise MVP and/or zero MVP), which are appended to MVP list as the last or any other entry. [0283] iii. In one example, furthermore, a single group of adjacent candidates is firstly included in the MVP list, then a joint group of non-adjacent and HMVP are reordered and subsequently appended to MVP list. In this case, at least one position is preserved for the virtual candidates (e.g., pairwise MVP and/or zero MVP), which are appended to MVP list as the last or any other entry. [0284] iv. In one example, furthermore, a joint group of adjacent candidates, non-adjacent and HMVP are reordered and subsequently appended to MVP list, and the virtual candidates (e.g., pairwise MVP and/or zero MVP) are appended to MVP list as the last or any other entry. [0285] c) Alternatively, the virtual candidates (e.g., pairwise MVP) of one category is included in a single/joint group and the virtual candidates of another category is not included. [0286] d) In one example, no virtual candidates (e.g., pairwise MVP and/or zero MVP) appear in the ultimate MVP list when reordering operation is performed for MVP list construction. [0287] 8. The number of candidates of a single/joint group may not be allowed to exceed a maximum candidate number. [0288] a) In one example, a single/joint group is constructed with limited amount of candidates constrained by maximum number N.sub.i, where i∈[0, 1, . . . , K] is the index of the corresponding group. N.sub.i may be the same or they may be different for different i. [0289] b) In one example, partial candidates in a single/joint group are limited by maximum number N.sub.i. [0290] i. In one example, one or multiple categories of candidates in a group are constructed with limited amount N.sub.i, while other categories in the same group can be included with arbitrary number. [0291] 1. In one example, the categories include but not limited to adjacent candidates, non-adjacent candidates, HMVP, pairwise candidates, etc. [0292] c) Alternatively, a first single/joint group may be constructed with at most N.sub.i MVP candidates, while a second single/joint groups may not have such constraint. [0293] d) In one example, N.sub.i is a fix value shared by both encoder and decoder. [0294] i. Alternatively, N.sub.i is determined by encoder and signalled in the bitstream. And decoder decodes N.sub.i value and then construct the corresponding i.sub.th single/joint group with at most N.sub.i candidates. [0295] ii. Alternatively, N.sub.i is derived in both encoder and decoder with the same operations, such that there is no need to signal the N.sub.i value. [0296] 1. In one example, encoder and decoder may derive the N.sub.i value based on the variance of all available motion information for i.sub.th group. [0297] 2. Alternatively, encoder and decoder may derive the N.sub.i value based on the number of all available candidates for i.sub.th group. [0298] 3. In one example, encoder and decoder may derive the N.sub.i value based on the number of the available adjacent candidates. a) In one example, N.sub.i is set to N-

NADJ, where N is a constant, N.sub.ADJ is the number of the available adjacent candidates. [0299] 4. Alternatively, furthermore, encoder and decoder may derive the N.sub.i value based on any information that encoder/decoder can both access to when constructing the MVP list. [0300] e) In one example, all or partial of the single/joint groups may share a same maximum candidate number N. [0301] 9. The construction of a single/joint group may depend on the maximum number constraint N.sub.i. [0302] a) In one example, all available MVP candidates for i.sub.th group are included in the group in accordance with a certain order. Once the candidate number in the current group reaches N.sub.i, the construction for group i is terminated. [0303] b) In one example, in above case, the order for group construction may be derived based on the distance between to-be-coded CU and MVP candidates, where a closer MVP candidate is assigned with a higher priority. [0304] c) Alternatively, the order may be derived based on a cost (such as a template matching) cost, where an MVP with a less cost has a higher priority. [0305] d) In one example, the construction of single/joint group is performed with at least one pruning operation in at least one group, or between groups. [0306] e) In one example, the constructed single/joint group is further reordered based on at least one cost method (e.g., template matching cost), then some or all of the candidates in this group may be included in the MVP list. [0307] i. Alternatively, the candidates in the constructed single/joint group will not be further reordered, and some or all of the candidates in this group are included into the MVP list in the same order as they are included in the group. [0308] 10. On how to prune MVP candidates. [0309] a) In one example, K-pass (e.g., K=2) pruning is performed to build an MVP list. [0310] 1. In one example, a first pruning may be performed inside at least one single/joint group, and a second pass pruning may be performed between at least two candidates that belong to different groups. [0311] a) In one example, in the first pass pruning, the pruning thresholds for two single/joint groups may be the same, or may be different. [0312] b) In one example, furthermore, in the first pass pruning, some of single/joint groups may share a same threshold value, while other single/joint groups may use different threshold values. [0313] 2. In one example, furthermore, the threshold for a certain pass or group is determined by the decoding information, including but not limited to the block size, coding tools been used (e.g., TM, DMVR, adaptive DMVR, CIIP, AFFINE, AMVP-merge). [0314] a) Alternatively, a threshold may be determined by at least one syntax element signaled to the decoder.

3. Problems

[0315] 1) Existing MVP list construction methods target at building a subset with constant MVP number from a given candidate set, which is normally realized by selecting the available candidates in a predefined order. This strategy, however, does not exploit the prior information produced during encoding/decoding process, which may lead to the mismatch between the true motion information and that of the candidates in the constructed MVP list. [0316] 2) Existing pruning process for MVP candidate only regards identical MVs as redundancy. Consequently, the constructed MVP list may contain quite similar MVs such that the diversity within the list is limited.

4. Detailed Solutions

[0317] In this disclosure, an enhanced MVP list derivation method based on template matching cost ordering is proposed. Instead of constructing the MVP list based on a predefined traversing order, an optimized MVP selecting approach is investigated by taking advantage of the matching cost in the reconstructed template region, such that more appropriate candidates are included in the list. [0318] It should be noted that the proposed strategy for MVP list construction can be utilized in normal merge and AMVP list construction process and can also be easily extended to other modules that require MVP derivation, e.g., merge with motion vector difference (MMVD), Affine motion compensation, Subblock-based temporal motion vector prediction (SbTMVP) and so on. [0319] In the following discussion, category represents the belongingness of an MVP candidate, e.g., non-adjacent MVP candidates belong to one category, HMVP candidates belonging to another category. A group denotes an MVP candidate set which contains one or multiple MVP candidates.

In one example, a single group denotes an MVP candidate set in which all the candidates belong to one category, e.g., adjacent MVP, non-adjacent MVP, HMVP, etc. In another example, a joint group denotes an MVP candidate set which contains candidates from multiple categories.
[0320] In the following discussion, "cost" of a candidate may be derived based on template matching or Bilateral matching, with functions such as SAD/SATD/SSD/MR-SAD (mean removal SAD).
[0321] The detailed embodiments below should be considered as examples to explain general concepts. These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner. Combination between this patent application and others are also applicable. [0322] 1. Multiple thresholds may be utilized to determine whether a candidate could be added to a candidate list in the candidate pruning process. [0323] a) A threshold may be used to determine whether a potential candidate can be put into a candidate list. [0324] i. For example, if the absolute difference of at least one component of the MV of the potential candidate and that of a candidate existing in the candidate list is smaller than a threshold, the potential candidate is not put into the list. [0325] ii. For example, if the absolute difference of all components of the MV of the potential candidate and that of a candidate existing in the candidate list is smaller than a threshold, the potential candidate is not put into the list. [0326] b) In one example, the candidate is an MVP candidate, the candidate pruning process is the MVP candidate pruning process, and the candidate list is a motion candidates list. [0327] i. In one example, the motion candidate list is a merge candidate list. [0328] ii. In one example, the motion candidate list is a AMVP candidate list. [0329] iii. In one example, the motion candidate list is an extend merge or AMVP list, such as sub-block merge candidate list, affine merge candidate list, MMVD list, GPM list, template matching merge list, biliteral matching merge list etc. [0330] iv. In one example, the motion candidate list is an IBC merge candidate list. [0331] v. In one example, the motion candidate list is an IBC AMVP candidate list. [0332] vi. In one example, the motion candidate list is an extend IBC merge or IBC AMVP list, such as IBC-MMVD list. [0333] c) In one example, the pruning thresholds may be different for two groups, where the group can be either a single group (containing only one category of candidates) or a joint group (containing at least two categories of candidates). [0334] d) In one example, N (e.g., N=2) thresholds are used in the pruning process. [0335] i. Assume A is the MVP set which contains all available MVP candidates regardless of category, in one example, a first threshold is used for a first subset of candidates in set A, and a second threshold is used for a second subset of candidates (e.g. the rest candidates excluding those in the first subset) in set A. [0336] ii. In one example, a first threshold is used for a single group denoted by A, and a second threshold is used for another group (single or joint)/multiple other groups/rest of candidates which are not with the same category as those in A. [0337] 1) In one example, a first threshold is used for the single group of adjacent candidates, and a second threshold is used for the rest candidates, including but not limited to non-adjacent MVP, HMVP, pairwise MVP and zero MVP. [0338] iii. The first threshold may be larger than or smaller than the second threshold. [0339] e) In one example, K-pass (e.g., K=4) pruning are conducted to construct MVP list. [0340] i. In one example, the 1.sup.st pass pruning (termed as P1) is performed within single or joint group to avoid duplicate candidates. [0341] 1) In one example, partial or all of the groups may sort (i.e., ARMC) after P1. [0342] ii. In one example, the 2.sup.nd pass pruning (termed as P2) is performed when multiple groups are merged into one or multiple hybrid group(s). [0343] 1) In one example, the hybrid group(s) may perform sorting or not after P2. [0344] iii. In one example, some new candidates may be inserted into the hybrid group, and the 3rd pass pruning is triggered to ensure no duplicate exists after the new candidates added. [0345] iv. In one example, the 4.sup.th pass pruning (termed as P4) is performed to further increase the diversity within the hybrid group(s). [0346] v. In one example, the multiple pass pruning described above may be utilized in a separate or combined way. [0347] 1) In one example, only partial pass is used to construct MVP list, i.e., P1->P2->p4, P1->P2->p3, P1->P2, P1->P3, P1->P3->p4, P1->p4 etc.

[0348] 2) In one example, the order of each pass may change during the construction process, i.e., a later pass pruning may perform before a former pass pruning. [0349] 3) In one example, certain pruning pass may perform multiple time during the construction.   a) In one example, the pruning may performed in order of P1->P4->P2->P3->p4. [0350] vi. In one example, the threshold used in different passes may be the same or different. [0351] 1) In one example, the threshold in a certain pass pruning may be a constant. [0352] 2) In one example, the threshold in a certain pass pruning may be derived from the bitstream.   a) In one example, all available threshold values may be stored in a look-up table or any other data structure, and the index of the selected threshold is signalled in the bitstream. The decoder may first parse the threshold index and then fetches the threshold value from the corresponding look-up table or other data structure.   b) In one example, the threshold may be derived based on the information of current block, i.e. QP or Lagrange multiplier (Lamda) used in RDO process. [0353] 2. On how to construct MVP list. [0354] a) One or multiple groups may be firstly constructed, where each group comprises the candidates belong to one or multiple categories. [0355] i. In one example, the category may include but not limited to adjacent MVP, non-adjacent MVP, HMVP, pairwise MVP, constructed MVP, etc. [0356] ii. In one example, the candidate number in each group may not be allowed to exceed a certain value. [0357] 1) In one example, the maximum allowed number for each group may be a constant or determined on-the-fly. [0358] 2) In one example, the maximum allowed number for each group may be different. [0359] iii. In one example, if only one group is constructed, the candidates belong to different categories are inserted into the group based on a pre-defined order. [0360] 1) In one example, specifically, in the constructed group, the candidates number of certain one or multiple categories cannot exceed a constant or a value that determined on-the-fly. [0361] iv. Pruning operation may be performed or not during the construction of each group. [0362] 1) In one example, specifically, the pruning is performed within the group, i.e., no duplicate exists for arbitrary two candidates that are from arbitrary one group. [0363] 2) Alternatively, the pruning is performed among the group, i.e., no duplicate exists for arbitrary two candidates that are from arbitrary one or two group. [0364] v. Pruning threshold for arbitrary two groups may be the same or not. [0365] b) If multiple groups have been constructed, some or all groups may then merge into a hybrid group. [0366] i. In one example, if only one group is constructed in a), then no merging process is conducted and this group will be regarded as a special case of hybrid group. [0367] ii. In one example, specifically, if the group(s) before merging have not performed pruning or already performed within-group pruning, then a second pass pruning is performed during the merging process. [0368] iii. Alternatively, specifically, if the group(s) before merging have already performed among-group pruning, no pruning is performed during the merging process. [0369] c) Subsequently, the hybrid group may be sorted based on ARMC or any other metric. [0370] i. In one example, specifically, before or after the hybrid group is sorted, all or partial candidates within the group may be refined by template or bilateral matching. [0371] ii. In one example, specifically, zero MVPs are excluded in the sorting process, which may be forced to place at the end of the sorted list. [0372] d) Constructed candidates (i.e., Pairwise candidates) may be generated and/or inserted into the hybrid group, and/or another round of sorting may be evoked to reorder the extended group. [0373] i. In one example, the constructed candidates may be generated based on the sorted group. [0374] 1) In one example, specifically, the constructed candidates can be pairwise candidates. [0375] 2) In one example, specifically, the constructed candidates are inserted in the hybrid group along with pruning operation. [0376] e) Finally, the last round pruning is performed to further increase the diversity within the larger group(s). [0377] i. In one example, the template matching cost for all the candidates in the sorted list are calculated, and the minimum cost difference between a candidate and its predecessor among all candidates in the list is determined. If this minimum cost difference is smaller than TH, the candidate will be discarded and it is moved at a further position in the list. This further position is the first position where the cost difference relative to its predecessor is larger than TH. This algorithm stops after a finite number of iterations, or the remaining candidates number reaches the target value for the MVP list. [0378] a) In one example, the TH may be derived based on the information of current block, i.e. QP or Lagrange multiplier (Lamda) used in RDO process. [0379] 3. The disclosed methods above can be applied on potential candidates before being put into the candidate list, or may be applied on candidates after being put into the candidate list. [0380] 4. General claims [0381] 1) Whether to and/or how to apply the disclosed methods above may be signalled at sequence level/group of pictures level/picture level/slice level/tile group level, such as in sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header. [0382] 2) Whether to and/or how to apply the disclosed methods above may be signalled at PB/TB/CB/PU/TU/CU/VPDU/CTU/CTU row/slice/tile/sub-picture/other kinds of region contain more than one sample or pixel. [0383] 3) Whether to and/or how to apply the disclosed methods above may be dependent on coded information, such as block size, colour format, single/dual tree partitioning, colour component, slice/picture type. [0384] 5. On how to generate reference template for a block with sub-block level motion information. [0385] a) The template comprises of the reconstructed regions in above or/and left position adjacent to the current block. [0386] b) In one example, to get the reference template of the current template T, T may firstly be split into one or more template segment(s). [0387] i. In one example, the width (for above template segment) and/or height (for left template segment) of each segment equals to the size of sub-block in the current block, as shown in FIG. **11**, which illustrates a diagram **1100** of an example of the template for a block with sub-block level motion information. [0388] c) In one example, for each template segment, the motion information of at least one adjacent/non-adjacent sub-block is used to get reference template segment, e.g., the motion information of sub-block A may be used to generate the reference template segment for TU_A and TL_A, and/or the motion information of sub-block C may be used to generate the reference template segment for TU_C, as shown in FIG. **11**. [0389] i. In one example, the reference template segment may be uni-predicted or bi-predicted. [0390] 1) In one example, all the reference template segments of the current block are uni-predicted (or bi-predicted). [0391] 2) In one example, for the template segments belong to one coding block, some of them may be uni-predicted, while others may be bi-predicted. [0392] ii. In one example, for arbitrary template segment T_SEG, if the adjacent sub-block is bi-predicted (i.e. with two sets of motion information, each set includes respective MV, reference and etc), then both sets of information or either one of them may be used to generate the reference template (segment). [0393] 1) In one example, the motion information in one set corresponds to a certain reference list. [0394] 2) In one example, specifically, both sets of motion information may be used to generate the predictions of the template. In particular, two reference segments predictions are respectively generated by fetching the reference regions of the current template segment specified by the corresponding set of motion information, then the weighted average of the two predictions are used as the reference template segment. [0395] 3) In one example, only one set of motion information of the two is used to generate the reference template segment. a) In one example, which set of the motion information is used may depend on the motion information of the centre sub-block of the current block. [0396] 4) In one example, the MV(s) may firstly be scaled to a certain reference frame before generating the prediction or reference template. [0397] iii. In one example, for arbitrary template segment T_SEG, if the adjacent sub-block is uni-predicted (i.e. with only one set of motion information), then the reference region of the current template segment specified by the same set of motion information is used as the reference template region. [0398] 1) In one example, alternatively, for arbitrary template segment T_SEG, even if the corresponding sub-block is uni-predicted, the reference template segment of T_SEG may be generated with bi-prediction. a) In one example, besides the uni-directional motion information of the adjacent sub-block, an additional set of motion information may be constructed based on one or multiple of the following methods: i. use zero MV. ii. Scale the existing MV to another reference frame. iii. Mirror the existing MV. iv. Fetch from non-adjacent sub-blocks. [0399] 2) In one example, alternatively, for arbitrary template segment T_SEG, if the adjacent sub-block is uni-predicted, then zero MV instead of the motion information of the adjacent sub-block may be used to generate reference template. [0400] d) In one example, for each template segment of the current template, whether the corresponding reference template is uni-predicted or bi-predicted, and/or which reference list is used to provided motion information for the reference template segment, may depend on the motion information of a certain sub-block (such as the centre sub-block or a sub-block at any other position of the current coding block). [0401] i. In one example, if the certain sub-block of the current coding block is bi-predicted, then all the reference template segments may be bi-predicted. [0402] 1) In one example, specifically, for each template segment, the motion information of the adjacent sub-block is fetched to generate the reference template segment. If the adjacent sub-block has only uni-directional motion information (i.e., the motion information is from one reference list), then additional motion information (which may be associated with the other reference list) is constructed based on one or multiple of the following methods: i. use zero MV. ii. Scale the existing MV to another reference frame. iii. Mirror the existing MV. iv. Fetch from non-adjacent sub-blocks. [0403] 2) In one example, in above case, the reference index of the constructed motion information may be the same as that of the certain sub-block's motion information associated with the same reference list. [0404] ii. In one example, alternatively, if the certain sub-block of the current coding block is uni-predicted, then all the reference template segments are uni-predicted. [0405] 1) In one example, specifically, for each template segment, the motion information associated with the same reference list as the certain sub-block which is fetched from the adjacent sub-block, which is used to generate the reference template segment. If the corresponding motion information doesn't exist in the adjacent sub-block, then it may be constructed based on one or multiple of the following methods: i. use zero MV. ii. Scale the existing MV to another reference frame. iii. Mirror the existing MV. iv. Fetch from non-adjacent sub-blocks. [0406] 2) In one example, in above case, the reference index of the constructed motion information may be the same as that of the certain sub-block's motion information associated with the same reference list. [0407] e) The above mentioned methods may be applied to any coding tool with sub-block level motion information, including but not limited to AFFINE, SbTMVP and etc.

5. Embodiments

[0408] In one example, when encoder/decoder starts to build an MVP candidate list, multiple small groups will be firstly constructed, where each group comprises the candidates from one or multiple categories. In particular, the number of the candidate in each group should not exceed the maximum allowed number, wherein the maximum number may vary from one group to another. Besides, within-group pruning operation with a constant threshold is conducted along with the construction of each group. After each group is constructed, all or partial of them will further merge into a hybrid group, here the 2.sup.nd pass pruning is triggered to exclude the redundant candidates in the larger group. Then, all or partial of the candidates in the mixed group are sorted based on ARMC method, and it should be noted that all or partial candidates before ARMC may be firstly refined by template matching or bilateral matching. Based on the sorted hybrid group, some constructed candidates, i.e., pairwise candidates, may be generated and then insert into the hybrid group along with the 3.sup.rd pass pruning operation. And the extended hybrid group performs ARMC again and all the candidates are sorted based on the TM cost. Lastly, if the candidate number in the hybrid group is larger than the maximum allowed value for the MVP list, the final pass pruning operation is conducted. In particular, the template matching cost for all the candidates in the sorted group are calculated, and the minimum cost difference between a candidate and its predecessor among all candidates is determined. If this minimum cost difference is smaller than a constant TH, the candidate will be discarded and it is moved at a further position in the list. This further position is the first position where the cost difference relative to its predecessor is larger than TH. This algorithm stops after a finite number of iterations, or the remaining candidates number reaches the target value for the MVP list.

[0409] FIG. **12** illustrates a flowchart of a method **1200** for video processing in accordance with embodiments of the present disclosure. The method **1200** may be implemented for a conversion between a current video block of a video and a bitstream of the video.

[0410] At block **1210**, a reference template for the current video block is determined based on a current template of the current video block and motion information of at least one subblock of the current video block. At block **1220**, the conversion is performed based on the reference template.

[0411] The method **1200** enables determines the reference template for a block based on subblock level motion information. The determined reference template may be more accurate. In this way, the coding effectiveness and coding efficiency can be improved.

[0412] In some embodiments, the current template of the current video block comprises at least one of: a first set of reconstructed regions above and adjacent to the current video block, or a second set of reconstructed regions left and adjacent to the current video block. In other words, the current template (also referred to as "template") of the current video block may include of the reconstructed regions in above or/and left position adjacent to the current video block, as shown in FIG. **11**.

[0413] In some embodiments, determining the reference template comprises: determining a plurality of template segments of the current template; determining a plurality of reference template segments based on the plurality of template segments and the motion information of the at least one subblock; and determining the reference template based on the plurality of reference template segments. For example, the template segments of the current template may include segments T.sub.U_A, T.sub.U_B, T.sub.U_C, T.sub.U_D, T.sub.L_A, T.sub.L_E, T.sub.L_F, T.sub.L_G, T.sub.L_A, T.sub.L_A, T.sub.L_A, T.sub.L_A as shown in FIG. **11**. A plurality of reference template segments may be determined based on these template segments and motion information of subblocks associated with these template segments.

[0414] In some embodiments, determining a plurality of reference template segments comprises: for a first template segment in the plurality of template segments, determining, from the at least one subblock, a first subblock associated with the first template segment; and determining a first reference template segment in the plurality of reference template segments based on motion information of the first subblock. By way of example, the first subblock may be a subblock adjacent to the first template segment, or a subblock non-adjacent to the first template segment. That is, motion information of an adjacent or non-adjacent subblock of a template segment may be used to determine the corresponding reference template segment. For example, motion information of subblock A in FIG. **11** may be used to generate the reference template segment for T.sub.U_A, T.sub.L_A, or for T.sub.U_B. For another example, motion information of subblock C in FIG. **11** may be used to generate the reference template segment for T.sub.U_C.

[0415] In some embodiments, the first subblock adjacent to the first template segment is bi-predicted, at least one of two sets of motion information of the first subblock is used to determine the first reference template segment.

[0416] In some embodiments, a set of motion information of the first subblock is associated with a reference list, and the set of motion information comprises at least one of: a motion vector, or a reference frame in the reference list.

[0417] In some embodiments, determining the first reference template segment comprises: determining two predictions of the first reference segment based on the two sets of motion information; and determining the first reference template segment based on a weighted average of the two predictions. For example, two reference segments predictions may be respectively generated by fetching the reference regions of the current template segment specified by the corresponding set of motion information. Then, the weighted average of the two predictions may be used as the reference template segment.

[0418] In some embodiments, determining two prediction of the first reference segment comprises:

updating a set of motion information of the two sets of motion information by scaling a motion vector in the set of motion information to a reference frame; and determining a corresponding prediction of the first reference segment based on the updated set of motion information. For example, the MV(s) may be firstly scaled to a certain reference frame before generating the prediction or reference template.

[0419] In some embodiments, determining the first reference template segment comprises: selecting a set of motion information from the two set of motion information based on motion information of a second subblock of the current video block; and determining the first reference template segment based on the selected set of motion information.

[0420] In some embodiments, the second subblock comprises a center subblock of the current video block. Alternatively, the second subblock may be a subblock at a certain position of the current video block.

[0421] In some embodiments, the first subblock adjacent to the first template segment is uni-predicted. In some embodiments, a region of the first reference template segment is determined based on a single set of motion information of the first subblock. For example, for an arbitrary template segment T_.sub.SEG, if the adjacent subblock is uni-predicted, that is, with only one set of motion information, then the reference region of the current template segment specified by the same set of motion information is used as the reference template region.

[0422] In some embodiments, a first set of motion information is associated with the first subblock, and determining the first reference template segment comprises: determining a second set of motion information based on at least one of: a zero motion vector, a scaled motion vector of a first motion vector in the first set of motion information, the scaled motion vector being associated with a second reference frame different from a first reference frame of the first motion vector, a mirrored motion vector of a second motion vector in the first set of motion information, a third set of motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the first set of motion information and the second set of motion information. In other words, for the arbitrary template segment, even if the corresponding subblock is uni-predicted, the reference template segment of the template segment may be generated with bi-prediction. For example, besides the uni-directional motion information of the adjacent subblock, an additional set of motion information may be constructed based on one or more of the above methods.

[0423] In some embodiments, a first reference index of the second set of motion information is the same with a second reference index of motion information of a second subblock of the current video block, the first set of motion information and the motion information of the second subblock being from a same reference list. That is, the reference index of the constructed motion information may be the same as that of a certain subblock's motion information associated with the same reference list. The certain subblock may be the center subblock or a subblock at a certain position.

[0424] In some embodiments, motion information associated with a reference list of the first subblock is unavailable, and determining the first reference template segment comprises: determining motion information of the first subblock based on at least one of: a zero motion vector, a scaled motion vector of an existing motion vector associated with the current video block, the scaled motion vector being associated with a second reference frame different from a first reference frame of the existing motion vector, a mirrored motion vector of an existing motion vector associated with the current video block, further motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the motion information of the first subblock.

[0425] In some embodiments, if a certain subblock of the current video block is uni-predicted, then all the reference template segments are uni-predicted. In one example, for each template segment, the motion information associated with the same reference list as the certain subblock which is fetched from the adjacent subblock, which is used to generate the reference template segment. If the corresponding motion information does not exist in the adjacent subblock, then it may be constructed based on one or more methods as described above.

[0426] In some embodiments, the reference list of the first subblock is the same with a reference list of a second subblock, and a first reference index of the motion information of the first subblock is the same with a second reference index of motion information of the second subblock. For example, the reference index of the constructed motion information may be the same as that of the certain subblock's motion information associated with the same reference list.

[0427] In some embodiments, the second subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

[0428] In some embodiments, the first reference template segment is determined based on a zero motion vector. That is, for arbitrary template segment, if the adjacent subblock is uni-predicted, then zero MV instead of the motion information of the adjacent subblock may be used to generate the reference template segment of the reference template.

[0429] In some embodiments, a template segment in the plurality of template segments is above to the current video block, and a width of the template segment is equal to a width of a subblock in the current video block.

[0430] In some embodiments, a template segment in the plurality of template segments is left to the current video block, and a height of the template segment is equal to a height of a subblock in the current video block.

[0431] In some embodiments, motion information of a single subblock of the current video block is used to determine one or more reference template segments.

[0432] In some embodiments, a reference template segment in the plurality of reference template segments is uni-predicted or bi-predicted.

[0433] In some embodiments, the plurality of reference template segments is uni-predicted or bi-predicted.

[0434] In some embodiments, for a set of template segments in the plurality of template segments, the set of template segments belonging to a coding block, a first subset of the set of template segments is uni-predicted, and a second subset of the set of template segments is bi-predicted.

[0435] In some embodiments, for a first template segment in the plurality of template segments, whether a corresponding reference template segment is uni-predicted or bi-predicted is based on motion information of a predefined subblock of the current video block.

[0436] In some embodiments, for a first template segment in the plurality of template segments, a reference list associated with motion information for a corresponding reference template segment is determined based on motion information of a predefined subblock of the current video block.

[0437] In some embodiments, the predefined subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

[0438] In some embodiments, the predefined subblock is bi-predicted, and the plurality of reference template segments is bi-predicted.

[0439] In some embodiments, the predefined subblock is uni-predicted, and the plurality of reference template segments is bi-predicted.

[0440] In some embodiments, the current video block is coded with a coding tool with subblock level motion information. By way of example, the coding tool may include at least one of: an affine coding tool, or a subblock-based temporal motion vector prediction (SbTMVP) coding tool.

[0441] In some embodiments, information regarding applying the method is included in the bitstream.

[0442] In some embodiments, the information is included in at least one of: a sequence level, a group of pictures level, a picture level, a slice level, a tile group level, a sequence header. a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a decoded parameter set (DPS), decoding capability information (DCI), a picture parameter set (PPS), an adaptation parameter set (APS), a slice header, or a tile group header.

[0443] In some embodiments, the information is included in a region containing more than one sample or pixel. By way of example, the region comprising one of: a prediction block (PB), a transform block (TB), a coding block (CB), a prediction unit (PU), a transform unit (TU), a coding unit (CU), a virtual pipeline data unit (VPDU), a coding tree unit (CTU), a CTU row, a slice, a tile, a subpicture.

[0444] In some embodiments, the information is based on coded information of the current video block. In some embodiments, the coded information comprises at least one of: a coding mode, a block size, a colour format, a single or dual tree partitioning, a colour component, a slice type, or a picture type.

[0445] In some embodiments, the conversion includes encoding the current video block into the bitstream. Alternatively, or in addition, in some embodiments, the conversion includes decoding the current video block from the bitstream.

[0446] According to further embodiments of the present disclosure, a non-transitory computer-readable recording medium is provided. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing. In the method, a reference template for a current video block of the video is determined based on a current template of the current video block and motion information of at least one subblock of the current video block. The bitstream is generated based on the reference template.

[0447] According to still further embodiments of the present disclosure, a method for storing bitstream of a video is provided. In the method, a reference template for a current video block of the video is determined based on a current template of the current video block and motion information of at least one subblock of the current video block. The bitstream is generated based on the reference template. The bitstream is stored in a non-transitory computer-readable recording medium.

[0448] Implementations of the present disclosure can be described in view of the following clauses, the features of which can be combined in any reasonable manner.

[0449] Clause 1. A method for video processing, comprising: determining, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and performing the conversion based on the reference template.

[0450] Clause 2. The method of clause 1, wherein the current template of the current video block comprises at least one of: a first set of reconstructed regions above and adjacent to the current video block, or a second set of reconstructed regions left and adjacent to the current video block.

[0451] Clause 3. The method of clause 1 or clause 2, wherein determining the reference template comprises: determining a plurality of template segments of the current template; determining a plurality of reference template segments based on the plurality of template segments and the motion information of the at least one subblock; and determining the reference template based on the plurality of reference template segments.

[0452] Clause 4. The method of clause 3, wherein determining a plurality of reference template segments comprises: for a first template segment in the plurality of template segments, determining, from the at least one subblock, a first subblock associated with the first template segment; and determining a first reference template segment in the plurality of reference template segments based on motion information of the first subblock.

[0453] Clause 5. The method of clause 4, wherein the first subblock comprises at least one of: a subblock adjacent to the first template segment, or a subblock non-adjacent to the first template segment.

[0454] Clause 6. The method of clause 4 or clause 5, wherein the first subblock adjacent to the first template segment is bi-predicted, at least one of two sets of motion information of the first subblock is used to determine the first reference template segment.

[0455] Clause 7. The method of clause 6, wherein a set of motion information of the first subblock is associated with a reference list, and the set of motion information comprises at least one of: a motion vector, or a reference frame in the reference list.

[0456] Clause 8. The method of clause 6 or clause 7, wherein determining the first reference template segment comprises: determining two predictions of the first reference segment based on the two sets of motion information; and determining the first reference template segment based on a weighted average of the two predictions.

[0457] Clause 9. The method of clause 8, wherein determining two prediction of the first reference segment comprises: updating a set of motion information of the two sets of motion information by scaling a motion vector in the set of motion information to a reference frame; and determining a corresponding prediction of the first reference segment based on the updated set of motion information.

[0458] Clause 10. The method of clause 6 or clause 7, wherein determining the first reference template segment comprises: selecting a set of motion information from the two set of motion information based on motion information of a second subblock of the current video block; and determining the first reference template segment based on the selected set of motion information.

[0459] Clause 11. The method of clause 10, wherein the second subblock comprises a center subblock of the current video block.

[0460] Clause 12. The method of clause 4 or clause 5, wherein the first subblock adjacent to the first template segment is uni-predicted.

[0461] Clause 13. The method of clause 12, wherein a region of the first reference template segment is determined based on a single set of motion information of the first subblock.

[0462] Clause 14. The method of clause 12, wherein a first set of motion information is associated with the first subblock, and determining the first reference template segment comprises: determining a second set of motion information based on at least one of: a zero motion vector, a scaled motion vector of a first motion vector in the first set of motion information, the scaled motion vector being associated with a second reference frame different from a first reference frame of the first motion vector, a mirrored motion vector of a second motion vector in the first set of motion information, a third set of motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the first set of motion information and the second set of motion information.

[0463] Clause 15. The method of clause 14, wherein a first reference index of the second set of motion information is the same with a second reference index of motion information of a second subblock of the current video block, the first set of motion information and the motion information of the second subblock being from a same reference list.

[0464] Clause 16. The method of clause 12, wherein motion information associated with a reference list of the first subblock is unavailable, and determining the first reference template segment comprises: determining motion information of the first subblock based on at least one of: a zero motion vector, a scaled motion vector of an existing motion vector associated with the current video block, the scaled motion vector being associated with a second reference frame different from a first reference frame of the existing motion vector, a mirrored motion vector of an existing motion vector associated with the current video block, further motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the motion information of the first subblock.

[0465] Clause 17. The method of clause 16, wherein the reference list of the first subblock is the same with a reference list of a second subblock, and a first reference index of the motion information of the first subblock is the same with a second reference index of motion information of the second subblock.

[0466] Clause 18. The method of clause 15 or clause 17, wherein the second subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

[0467] Clause 19. The method of clause 12, wherein the first reference template segment is determined based on a zero motion vector.

[0468] Clause 20. The method of any of clauses 3-19, wherein a template segment in the plurality of template segments is above to the current video block, and a width of the template segment is equal to a width of a subblock in the current video block.

[0469] Clause 21. The method of any of clauses 3-19, wherein a template segment in the plurality of template segments is left to the current video block, and a height of the template segment is equal to a height of a subblock in the current video block.

[0470] Clause 22. The method of any of clauses 3-21, wherein motion information of a single subblock of the current video block is used to determine one or more reference template segments.

[0471] Clause 23. The method of any of clauses 3-22, wherein a reference template segment in the plurality of reference template segments is uni-predicted or bi-predicted.

[0472] Clause 24. The method of any of clauses 3-22, wherein the plurality of reference template segments is uni-predicted or bi-predicted.

[0473] Clause 25. The method of any of clauses 3-24, wherein for a set of template segments in the plurality of template segments, the set of template segments belonging to a coding block, a first subset of the set of template segments is uni-predicted, and a second subset of the set of template segments is bi-predicted.

[0474] Clause 26. The method of any of clauses 3-25, wherein for a first template segment in the plurality of template segments, whether a corresponding reference template segment is uni-predicted or bi-predicted is based on motion information of a predefined subblock of the current video block.

[0475] Clause 27. The method of any of clauses 3-25, wherein for a first template segment in the plurality of template segments, a reference list associated with motion information for a corresponding reference template segment is determined based on motion information of a predefined subblock of the current video block.

[0476] Clause 28. The method of clause 26 or clause 27, wherein the predefined subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

[0477] Clause 29. The method of any of clauses 26-28, wherein the predefined subblock is bi-predicted, and the plurality of reference template segments is bi-predicted.

[0478] Clause 30. The method of any of clauses 26-28, wherein the predefined subblock is uni-predicted, and the plurality of reference template segments is bi-predicted.

[0479] Clause 31. The method of any of clauses 1-30, wherein the current video block is coded with a coding tool with subblock level motion information.

[0480] Clause 32. The method of clause 31, wherein the coding tool comprises at least one of: an affine coding tool, or a subblock-based temporal motion vector prediction (SbTMVP) coding tool.

[0481] Clause 33. The method of any of clauses 1-32, wherein information regarding applying the method is included in the bitstream.

[0482] Clause 34. The method of clause 33, wherein the information is included in at least one of: a sequence level, a group of pictures level, a picture level, a slice level, a tile group level, a sequence header. a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a decoded parameter set (DPS), decoding capability information (DCI), a picture parameter set (PPS), an adaptation parameter set (APS), a slice header, or a tile group header.

[0483] Clause 35. The method of clause 33, wherein the information is included in a region containing more than one sample or pixel.

[0484] Clause 36. The method of clause 35, wherein the region comprising one of: a prediction block (PB), a transform block (TB), a coding block (CB), a prediction unit (PU), a transform unit (TU), a coding unit (CU), a virtual pipeline data unit (VPDU), a coding tree unit (CTU), a CTU row, a slice, a tile, a subpicture.

[0485] Clause 37. The method of any of clauses 33-36, wherein the information is based on coded information of the current video block.

[0486] Clause 38. The method of clause 37, wherein the coded information comprises at least one of: a coding mode, a block size, a colour format, a single or dual tree partitioning, a colour component, a slice type, or a picture type.

[0487] Clause 39. The method of any of clauses 1-38, wherein the conversion includes encoding the current video block into the bitstream.

[0488] Clause 40. The method of any of clauses 1-38, wherein the conversion includes decoding the current video block from the bitstream.

[0489] Clause 41. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of clauses 1-40.

[0490] Clause 42. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of clauses 1-40.

[0491] Clause 43. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: determining a reference template for a current video block of the video based on a current template of the current video block and motion information of at least one subblock of the current video block; and generating the bitstream based on the reference template.

[0492] Clause 44. A method for storing a bitstream of a video, comprising: determining a reference template for a current video block of the video based on a current template of the current video block and motion information of at least one subblock of the current video block; generating the bitstream based on the reference template; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[0493] FIG. **13** illustrates a block diagram of a computing device **1300** in which various embodiments of the present disclosure can be implemented. The computing device **1300** may be implemented as or included in the source device **110** (or the video encoder **114** or **200**) or the destination device **120** (or the video decoder **124** or **300**).

[0494] It would be appreciated that the computing device **1300** shown in FIG. **13** is merely for purpose of illustration, without suggesting any limitation to the functions and scopes of the embodiments of the present disclosure in any manner.

[0495] As shown in FIG. **13**, the computing device **1300** includes a general-purpose computing device **1300**. The computing device **1300** may at least comprise one or more processors or processing units **1310**, a memory **1320**, a storage unit **1330**, one or more communication units **1340**, one or more input devices **1350**, and one or more output devices **1360**.

[0496] In some embodiments, the computing device **1300** may be implemented as any user terminal or server terminal having the computing capability. The server terminal may be a server, a large-scale computing device or the like that is provided by a service provider. The user terminal may for example be any type of mobile terminal, fixed terminal, or portable terminal, including a mobile phone, station, unit, device, multimedia computer, multimedia tablet, Internet node, communicator, desktop computer, laptop computer, notebook computer, netbook computer, tablet computer, personal communication system (PCS) device, personal navigation device, personal digital assistant (PDA), audio/video player, digital camera/video camera, positioning device, television receiver, radio broadcast receiver, E-book device, gaming device, or any combination thereof, including the accessories and peripherals of these devices, or any combination thereof. It would be contemplated that the computing device **1300** can support any type of interface to a user (such as “wearable” circuitry and the like).

[0497] The processing unit **1310** may be a physical or virtual processor and can implement various processes based on programs stored in the memory **1320**. In a multi-processor system, multiple processing units execute computer executable instructions in parallel so as to improve the parallel processing capability of the computing device **1300**. The processing unit **1310** may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[0498] The computing device **1300** typically includes various computer storage medium. Such medium can be any medium accessible by the computing device **1300**, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory **1320** can be a volatile memory (for example, a register, cache, Random Access Memory (RAM)), a non-volatile memory (such as a Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory), or any combination thereof. The storage unit **1330** may be any detachable or non-detachable medium and may include a machine-readable medium such as a memory, flash memory drive, magnetic disk or another other media, which can be used for storing information and/or data and can be accessed in the computing device **1300**.

[0499] The computing device **1300** may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in FIG. **13**, it is possible to provide a magnetic disk drive for reading from and/or writing into a detachable and non-volatile magnetic disk and an optical disk drive for reading from and/or writing into a detachable non-volatile optical disk. In such cases, each drive may be connected to a bus (not shown) via one or more data medium interfaces.

[0500] The communication unit **1340** communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device **1300** can be implemented by a single computing cluster or multiple computing machines that can communicate via communication connections. Therefore, the computing device **1300** can operate in a networked environment using a logical connection with one or more other servers, networked personal computers (PCs) or further general network nodes.

[0501] The input device **1350** may be one or more of a variety of input devices, such as a mouse, keyboard, tracking ball, voice-input device, and the like. The output device **1360** may be one or more of a variety of output devices, such as a display, loudspeaker, printer, and the like. By means of the communication unit **1340**, the computing device **1300** can further communicate with one or more external devices (not shown) such as the storage devices and display device, with one or more devices enabling the user to interact with the computing device **1300**, or any devices (such as a network card, a modem and the like) enabling the computing device **1300** to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[0502] In some embodiments, instead of being integrated in a single device, some or all components of the computing device **1300** may also be arranged in cloud computing architecture. In the cloud computing architecture, the components may be provided remotely and work together to implement the functionalities described in the present disclosure. In some embodiments, cloud computing provides computing, software, data access and storage service, which will not require end users to be aware of the physical locations or configurations of the systems or hardware providing these services. In various embodiments, the cloud computing provides the services via a wide area network (such as Internet) using suitable protocols. For example, a cloud computing provider provides applications over the wide area network, which can be accessed through a web browser or any other computing components. The software or components of the cloud computing architecture and corresponding data may be stored on a server at a remote position. The computing resources in the cloud computing environment may be merged or distributed at locations in a remote data center. Cloud computing infrastructures may provide the services through a shared data center, though they behave as a single access point for the users. Therefore, the cloud computing architectures may be used to provide the components and functionalities described herein from a service provider at a remote location. Alternatively, they may be provided from a conventional server or installed directly or otherwise on a client device.

[0503] The computing device **1300** may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory **1320** may include one or more video coding modules **1325** having one or more program instructions. These modules are accessible and executable by the processing unit **1310** to perform the functionalities of the various embodiments described herein.

[0504] In the example embodiments of performing video encoding, the input device **1350** may receive video data as an input **1370** to be encoded. The video data may be processed, for example, by the video coding module **1325**, to generate an encoded bitstream. The encoded bitstream may be provided via the output device **1360** as an output **1380**.

[0505] In the example embodiments of performing video decoding, the input device **1350** may receive an encoded bitstream as the input **1370**. The encoded bitstream may be processed, for example, by the video coding module **1325**, to generate decoded video data. The decoded video data may be provided via the output device **1360** as the output **1380**.

[0506] While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

Claims

1. A method for video processing, comprising: determining, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and performing the conversion based on the reference template.

2. The method of claim 1, wherein the current template of the current video block comprises at least one of: a first set of reconstructed regions above and adjacent to the current video block, or a second set of reconstructed regions left and adjacent to the current video block.

3. The method of claim 1, wherein determining the reference template comprises: determining a plurality of template segments of the current template; determining a plurality of reference template segments based on the plurality of template segments and the motion information of the at least one subblock; and determining the reference template based on the plurality of reference template segments.

4. The method of claim 3, wherein determining a plurality of reference template segments comprises: for a first template segment in the plurality of template segments, determining, from the at least one subblock, a first subblock associated with the first template segment; and determining a first reference template segment in the plurality of reference template segments based on motion information of the first subblock, wherein the first subblock comprises at least one of: a subblock adjacent to the first template segment, or a subblock non-adjacent to the first template segment,

5. The method of claim 4, wherein the first subblock adjacent to the first template segment is bi-predicted, at least one of two sets of motion information of the first subblock is used to determine the first reference template segment, and/or wherein a set of motion information of the first subblock is associated with a reference list, and the set of motion information comprises at least one of: a motion vector, or a reference frame in the reference list.

6. The method of claim 5, wherein determining the first reference template segment comprises: determining two predictions of the first reference segment based on the two sets of motion information; and determining the first reference template segment based on a weighted average of the two predictions, wherein determining two predictions of the first reference segment comprises: updating a set of motion information of the two sets of motion information by scaling a motion vector in the set of motion information to a reference frame; and determining a corresponding prediction of the first reference segment based on the updated set of motion information.

7. The method of claim 5, wherein determining the first reference template segment comprises: selecting a set of motion information from the two sets of motion information based on motion information of a second subblock of the current video block; and determining the first reference template segment based on the selected set of motion information, wherein the second subblock comprises a center subblock of the current video block.

8. The method of claim 4, wherein the first subblock adjacent to the first template segment is uni-predicted.

9. The method of claim 8, wherein a region of the first reference template segment is determined based on a single set of motion information of the first subblock, and/or wherein the first reference template segment is determined based on a zero motion vector.

10. The method of claim 8, wherein a first set of motion information is associated with the first subblock, and determining the first reference template segment comprises: determining a second set of motion information based on at least one of: a zero motion vector, a scaled motion vector of a first motion vector in the first set of motion information, the scaled motion vector being associated with a second reference frame different from a first reference frame of the first motion vector, a mirrored motion vector of a second motion vector in the first set of motion information, or a third set of motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the first set of motion information and the second set of motion information, wherein a first reference index of the second set of motion information is the same as a second reference index of motion information of a second subblock of the current video block, the first set of motion information and the motion information of the second subblock being from a same reference list, and/or wherein the second subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

11. The method of claim 8, wherein motion information associated with a reference list of the first subblock is unavailable, and determining the first reference template segment comprises: determining motion information of the first subblock based on at least one of: a zero motion vector, a scaled motion vector of an existing motion vector associated with the current video block, the scaled motion vector being associated with a second reference frame different from a first reference frame of the existing motion vector, a mirrored motion vector of an existing motion vector associated with the current video block, or further motion information of a non-adjacent subblock of the current video block; and determining the first reference template segment based on the motion information of the first subblock, wherein the reference list of the first subblock is the same with a reference list of a second subblock, and a first reference index of the motion information of the first subblock is the same with a second reference index of motion information of the second subblock, and/or wherein the second subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block.

12. The method of claim 3, wherein a template segment in the plurality of template segments is above the current video block, and a width of the template segment is equal to a width of a subblock in the current video block, or wherein a template segment in the plurality of template segments is left to the current video block, and a height of the template segment is equal to a height of a subblock in the current video block.

13. The method of claim 3, wherein motion information of a single subblock of the current video block is used to determine one or more reference template segments, and/or wherein a reference template segment in the plurality of reference template segments is uni-predicted or bi-predicted, and/or wherein the plurality of reference template segments is uni-predicted or bi-predicted, and/or wherein for a set of template segments in the plurality of template segments, the set of template segments belonging to a coding block, a first subset of the set of template segments is uni-predicted, and a second subset of the set of template segments is bi-predicted.

14. The method of claim 3, wherein for a first template segment in the plurality of template segments, whether a corresponding reference template segment is uni-predicted or bi-predicted is based on motion information of a predefined subblock of the current video block, or wherein for a first template segment in the plurality of template segments, a reference list associated with motion information for a corresponding reference template segment is determined based on motion information of a predefined subblock of the current video block.

15. The method of claim 14, wherein the predefined subblock comprises one of: a center subblock of the current video block, or a subblock at a predefined position of the current video block, and/or wherein the predefined subblock is bi-predicted, and the plurality of reference template segments is bi-predicted, or wherein the predefined subblock is uni-predicted, and the plurality of reference template segments is bi-predicted.

16. The method of claim 1, wherein the current video block is coded with a coding tool with subblock level motion information, wherein the coding tool comprises at least one of: an affine coding tool, or a subblock-based temporal motion vector prediction (SbTMVP) coding tool.

17. The method of claim 1, wherein the conversion includes encoding the current video block into the bitstream, or wherein the conversion includes decoding the current video block from the bitstream.

18. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to: determine, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and perform the conversion based on the reference template.

19. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method comprising: determining, for a conversion between a current video block of a video and a bitstream of the video, a reference template for the current video block based on a current template of the current video block and motion information of at least one subblock of the current video block; and performing the conversion based on the reference template.

20. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: determining a reference template for a current video block of the video based on a current template of the current video block and motion information of at least one subblock of the current video block; and generating the bitstream based on the reference template.