# MOTION VECTOR RANGE DERIVATION FOR ENHANCED INTERPOLATION FILTER

## TECHNICAL FIELD

Embodiments of the present application (disclosure) generally relate to the field of picture processing and more particularly to inter prediction, such as motion vector range derivation for Enhanced interpolation filter.

## BACKGROUND

Video coding (video encoding and decoding) is used in a wide range of digital video applications, for example broadcast digital TV, video transmission over internet and mobile networks, real-time conversational applications such as video chat, video conferencing, DVD and Blu-ray discs, video content acquisition and editing systems, and camcorders of security applications.

The amount of video data needed to depict even a relatively short video can be substantial, which may result in difficulties when the data is to be streamed or otherwise communicated across a communications network with limited bandwidth capacity. Thus, video data is generally compressed before being communicated across modern day telecommunications networks. The size of a video could also be an issue when the video is stored on a storage device because memory resources may be limited. Video compression devices often use software and/or hardware at the source to code the video data prior to transmission or storage, thereby decreasing the quantity of data needed to represent digital video images. The compressed data is then received at the destination by a video decompression device that decodes the video data. With limited network resources and ever increasing demands of higher video quality, improved compression and decompression techniques that improve compression ratio with little to no sacrifice in picture quality are desirable.

In Essential Video Coding (EVC), in case of using affine prediction mode, pixel-based affine transform motion compensation can be applied. Specifically, when applying an Enhanced bi-linear Interpolation Filter (EIF), the filtering procedure comprises deriving a pixel-based

CLAIMS1. A method for coding video data, comprising:

obtaining a center motion vector, mv_center of a coding block,

deriving a first motion vector range for the coding block based on the center motion vector and a motion vector spread, wherein the motion vector spread depends on a size of the coding block;

if the first motion vector range is at least partially pointing outside a first area including a reference picture, updating the first motion vector range to be pointing within the first area, such that a minimum value and/or a maximum value of the updated first motion vector range is pointing at the boundary of the first area, wherein the difference between the maximum value and the minimum value of the updated first motion vector range is equal to the minimum of a double value of the motion vector spread and the size of the first area; and

performing pixel-based motion compensation based on the updated first motion vector range.

2. The method of claim 1, wherein the motion vector spread is represented by a horizontal motion vector spread and/or a vertical motion vector spread, and the horizontal motion vector spread is derived based on the width of the coding block, and the vertical motion vector spread is derived based on the height of the coding block.

3. The method of claim 2, wherein the horizontal motion vector spread is represented by deviationMV[ log2CbWidth − 3 ], and the vertical motion vector spread is represented by deviationMV[ log2CbHeight − 3 ], wherein cbWidth and cbHeight represent the width and the height of the coding block.

4. The method of any one of claims 1 to 3,

wherein the first motion vector range is represented by a first MV horizontal component range and/or a first MV vertical component range,

the first MV horizontal component range comprises a first minimum MV horizontal component value, hor_min and a first maximum MV horizontal component value, hor_max, and

the first MV vertical component range comprises a first minimum MV vertical component value, ver_min and a first maximum MV vertical component value, ver_max.

5. The method of claim 4, further comprising:
deriving a second motion vector range based on a size of a picture comprising the coding block,
wherein the second motion vector range is represented by a second MV horizontal component range and/or a second MV vertical component range,
the second MV horizontal component range comprises a second minimum MV horizontal component value, hor_min_pic and a second maximum MV horizontal component value, hor_max_pic and
the second MV vertical component range comprises a second minimum MV vertical component value, ver_min_pic and a second maximum MV vertical component value, ver_max_pic.

6. The method of claim 5, wherein if the first motion vector range is at least partially pointing outside a first area including a reference picture, updating the first motion vector range to be pointing within the first area, comprises:
if the first minimum MV horizontal component value, hor_min is less than the second minimum MV horizontal component value, hor_min_pic,
setting a updated value of the first minimum MV horizontal component value, hor_min equal to the second minimum MV horizontal component value, hor_min_pic, and
deriving a updated value of the first maximum MV horizontal component value, hor_max based on a sum of the second minimum MV horizontal component value, hor_min_pic and the double value of the horizontal motion vector spread.

7. The method of claim 5, wherein if the first motion vector range is at least partially pointing outside a first area including a reference picture, updating the first motion vector range to be pointing within the first area, comprises:
if the first maximum MV horizontal component value, hor_max is greater than the second maximum MV horizontal component value, hor_max_pic,
setting a updated value of the first maximum MV horizontal component value, hor_max equal to the second maximum MV horizontal component value, hor_max_pic, and
deriving a updated value of the first minimum MV horizontal component value, hor_min based on a subtraction value of the second maximum MV horizontal component value, hor_max_pic and the doubled value of the horizontal motion vector spread.

8. The method of any one of claims 1 to 6, wherein

if the first motion vector range is at least partially pointing outside a left boundary of the first area, updating the variables hor_min and hor_max, representing a first MV horizontal component range of the first motion vector range, as:

hor_min = hor_min_pic,

hor_max = min (hor_max_pic, hor_min_pic + 2* horizontal motion vector spread);

wherein hor_min indicates the updated first minimum MV horizontal component, and hor_max indicates the updated first maximum MV horizontal component;

wherein hor_min_pic and hor_max_pic, representing a second MV horizontal component range of a second motion vector range;

hor_min_pic indicates a second minimum MV horizontal component of the second MV horizontal component range, and hor_max_pic indicates a second maximum MV horizontal component of the second MV horizontal component range, and

wherein the second motion vector range depends on a size of a picture comprising the coding block.

9. The method of any one of claims 1 to 5 and 7, wherein

if the first motion vector range is at least partially pointing outside a right boundary of the first area,

updating the variables hor_min and hor_max, representing a first MV horizontal component range of the first motion vector range, as:

hor_min = max (hor_min_pic, hor_max_pic – 2* horizontal motion vector spread),

hor_max = hor_max_pic;

wherein hor_min indicates the updated first minimum MV horizontal component, and hor_max indicates the updated first maximum MV horizontal component;

wherein hor_min_pic and hor_max_pic, representing a second MV horizontal component range of a second motion vector range;

hor_min_pic indicates a second minimum MV horizontal component of the second MV horizontal component range, and hor_max_pic indicates a second maximum MV horizontal component of the second MV horizontal component range, and

wherein the second motion vector range depends on a size of a picture comprising the coding block.

10. The method of any one of claims 1 to 5, wherein

if the first motion vector range is at least partially pointing outside an upper boundary of the first area,

updating the variables ver_min and ver_max, representing a first MV vertical component range of the first motion vector range, as:

ver_min = ver_min_pic,

ver_max = min (ver_max_pic, ver_min_pic + 2* vertical motion vector spread);

wherein ver_min indicates the updated first minimum MV vertical component, and ver_max indicates the updated first maximum MV vertical component;

wherein ver_min_pic and ver_max_pic, representing a second MV vertical component range of a second motion vector range;

ver_min_pic indicates a second minimum MV vertical component of the second MV vertical component range, and ver_max_pic indicates a second maximum MV horizontal component of the second MV horizontal component range, and

wherein the second motion vector range depends on a size of a picture comprising the coding block.

11. The method of any one of claims 1 to 5, wherein

if the first motion vector range is at least partially located outside a lower boundary of the first area,

updating the variables ver_min and ver_max, representing a first MV vertical component range of the first motion vector range, as:

ver_min = max (ver_min_pic, ver_max_pic – 2* vertical motion vector spread),

ver_max = ver_max_pic;

wherein ver_min indicates the updated first minimum MV vertical component of the first MV vertical component range, and ver_max indicates the updated first maximum MV vertical component of the first MV vertical component range;

wherein ver_min_pic and ver_max_pic, representing a second MV vertical component range of a second motion vector range;

ver_min_pic indicates a second minimum MV vertical component of the second MV vertical component range, and ver_max_pic indicates a second maximum MV horizontal component of the second MV horizontal component range, and

wherein the second motion vector range depends on a size of a picture comprising the coding block.

12. The method of claim 1, wherein a minimum or maximum value of a horizontal component of the updated first motion vector range is pointing at a left or right boundary of the first area, respectively, and/or wherein a minimum or maximum value of a vertical component of the updated first motion vector range is pointing at an upper or lower boundary of the first area, respectively.

13. The method of any one of claims 1 to 12, wherein the first area comprises the reference picture and an extended area surrounding the reference picture.

14. The method of any one of claims 1 to 13, further comprising:
performing a clipping operation on the updated first motion vector range to be within the range $[-2^{17}, 2^{17}-1]$;
wherein the performing pixel-based motion compensation based on the updated first motion vector range, comprising: performing pixel-based motion compensation based on the updated and clipped first motion vector range.

15. The method of any of claims 1 to 14, wherein the performing pixel-based motion compensation based on the updated motion vector range, comprises:
performing clipping operation on a motion vector of a pixel of the coding block to be within a range, to obtain a clipped motion vector, the range depends on the updated first motion vector range; and
performing pixel-based motion compensation based on the clipped motion vector.

16. A coder comprising processing circuitry for carrying out the method according to any one of claims 1 to 15.

17. The coder of claim 15, wherein the coder comprises an encoder or a decoder.

18. A computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method according to any one of the claims 1 to 15.

19. A decoder (30, 1800), comprising:
one or more processors (1810); and

a non-transitory computer-readable storage medium (1820) coupled to the one or more processors (1810) and storing instructions for execution by the one or more processors (1810), wherein the programming, when executed by the one or more processors (1810), configures the decoder (30, 1800) to carry out the method according to any one of claims 1 to 15.

20. An encoder (20, 1900), comprising:

one or more processors (1910); and

a non-transitory computer-readable storage medium (1920) coupled to the one or more processors (1910) and storing instructions for execution by the one or more processors (1910), wherein the programming, when executed by the processors (1910), configures the encoder (20, 1900) to carry out the method according to any one of the claims 1 to 15.