CLAIMS

1. A method for coding a video implemented in an encoder or a decoder including the enhanced interpolation filter, EIF, for motion compensation, the method comprising:

i) determining control point motion vectors, CPMVs, for a block according to affine inter-prediction, the block being an affine block or a sub-block of the affine block;

ii) for a predefined sub-block size determining a reference area for a sub-block with the predefined sub-block size according to values of the CPMVs;

iii) comparing the determined reference area with a predefined threshold;

iv) applying EIF for motion compensation, comprising deriving the pixel-based motion vector field for the block;

wherein if the determined reference area is larger than the predefined threshold, deriving the pixel-based motion vector field for the block further comprises motion vector clipping,

wherein motion vector clipping range is determined based on motion model of the block and the size of the block.

2. The method according to claim 1, wherein using EIF for motion compensation further comprises:

obtaining interpolated samples in a reference picture based on the derived motion vector field using a T taps interpolation filter, where T is the number of interpolation filter taps corresponding to the filter length, where T is larger than zero; and

applying a high-pass filter to the interpolated samples

3. The method of claim 1 or 2, using EIF for motion compensation further comprises:

determining a size of the block; and

comparing the size of the block with a first threshold for the size of the block;

wherein block-based affine transform prediction is performed on the block when the size of the block is greater than or equal to the first threshold for the size of the block.

4. The method of claim 3, wherein the size of the block is determined based on affine motion model parameters.

5. The method of claim 4, wherein the affine motion model parameters include motion vector differences of the CPMVs as well as a width and a height of the block.

6. The method of claim 4 or 5, wherein the affine motion model parameters further include a precision of the CPMVs.

7. The method of any one of claims 1 to 6, wherein the predefined threshold is calculated based on the predefined sub-block size and a predefined ratio of the reference area corresponding to the sub-block in the reference picture to the area of the sub-block.

8. The method of claim 7 wherein the predefined ratio of the reference area corresponding to the sub-block in the reference picture to the area of the sub-block corresponds to a memory bandwidth for the EIF.

9. The method of claim 7, wherein the predefined threshold is the maximum allowed memory access consumption for the predefined sub-block size of the block.

10. The method of any of claims 2 to 9, wherein determining the reference area for a sub-block with the predefined sub-block size according to values of the CPMVs comprises:

determining an expanded block by adding pixel margins to from each border of the sub-block based on a length of the high-pass filter;

deriving motion vectors for each corner of the expanded block based on the CPMVs;

deriving a transformed block in the reference picture based on the derived motion vectors; and

deriving a bounding box for the transformed block;

wherein the area of the reference block corresponds to the bounding box expanded by (T-1) pixel from each border of the bounding box.

11. The method according to claim 10, wherein the size of the bounding box for the transformed block in the reference picture is derived by using the following equation, where the max function returns the maximum value of the arguments, and the min function returns the minimum value of the arguments:

$$\begin{cases} W' = \max(X'_0, X'_1, X'_2, X'_3) - \min(X'_0, X'_1, X'_2, X'_3) + 1 \\ H' = \max(Y'_0, Y'_1, Y'_2, Y'_3) - \min(Y'_0, Y'_1, Y'_2, Y'_3) + 1 \end{cases} \quad (2\text{-}5)$$

where the location of the transformed block in reference picture is describe by the coordinates of the corner samples of the transformed block (top-left, top-right, bottom-left, bottom-right):

$$\begin{cases} (X_0', Y_0') = (0,0) \\ (X_1', Y_1') = (1 + W + (W+1)dHorX, (W+1)dHorY) \\ (X_2', Y_2') = ((H+1)dVerX, 1 + H + (H+1)dVerY) \\ (X_3', Y_3') = (1 + W + (W+1)dHorX + (H+1)dVerX, 1 + H + (W+1)dHorY + (H+1)dVerY) \end{cases}$$

where W and H are the sub-block width and heights respectively, and *dHorX*, *dHorY*, *dVerX*, *dVerY* are the horizontal and vertical gradient parameters of the affine motion model.

12. The method according to claim 11, wherein

the size of the bounding is given by W'xH'; or

the size of the bounding box is given by Ceil(W')xCeil(H'); or

the size of the bound box is given by Floor(W')xFloor(H').

13. The method of any one of claims 2 to 12, wherein the value of the filter length is predefined for both encoder and decoder, or specified in a parameter set of a codec video sequence.

14. The method of claim 13, wherein for a 4x4 sub-block, the predefined threshold T is given by

$$T = \frac{(4 + T' - 1) * (4 + T' - 1)}{4 * 4}$$

where T' is the length of motion compensation MC interpolation filter for a translational motion block.

15. The method of claim 14, wherein for an 8x8 sub-block, the predefined threshold T is given by

$$T = \frac{(8 + T' - 1) * (8 + T' - 1)}{8 * 8}$$

where T' is the length of motion compensation MC interpolation filter for a translational motion block.

16. The method of claim 15, further comprising:

calculating a motion vector for the center of the block and performing translational motion compensation.

17. The method of any one of claims 1 to 16, wherein the predefined size of the sub-block for the enhanced interpolation filter is equal to 4x4; wherein dX[0] corresponds to $dHorX$, dX[1] corresponds to $dHorY$, dY[0] corresponds to $dVerX$ and dY[1] corresponds to $dVerY$; wherein determining reference area for a sub-block with predefined sub-block size according to values of the CPMVs comprises:

setting the value of the variable eifSubblockSize equal to 4;

deriving the variable eifCanBeApplied as follows:

setting the value of eifCanBeApplied to TRUE;

– deriving the arrays X[i], Y[i] as follows:
– X[0] = 0;
– X[1] = (eifSubblockSize + 1) * (dX[0] + (1 << 9));
– X[2] = (eifSubblockSize + 1) * dY[0];
– X[3] = X[1] + X[2];
– Y[0] = 0;
– Y[1] = (eifSubblockSize + 1) * dX[1];
– Y[2] = (eifSubblockSize + 1) * (dY[1] + (1 << 9));
– Y[3] = Y[1] + Y[2];
– setting the value of the variable Xmax equal to maximum of X[i] for i being equal to 0.. 3
– setting the value of the variable Xmin equal to minimum of X[i] for i being equal to 0.. 3
– setting the value of the variable Ymax equal to maximum of Y[i] for i being equal to 0.. 3
– setting the value of the variable Ymin equal to minimum of Y[i] for i being equal to 0.. 3
– setting the value of the variable W equal to the value of (Xmax – Xmin + (1 << 9) – 1) >> 9
– setting the value of the variable H equal to the value of (Ymax – Ymin + (1 << 9) – 1) >> 9.
–

18. The method of any one of claims 1 to 17, wherein in case the affine inter-prediction comprises bi-prediction, the constraint symmetrically applied for both lists.

19. The method of any one of claims 2 to 18, wherein T is equal to 2.

20. The method of any one of the preceding claims, wherein the predefined threshold is equal to 72.

21. The method of any one of the preceding claims, wherein motion model is affine motion model and motion vector clipping range is determined based on motion model means determined based on CPMVs.

22. The method of any one of claims 1 to 21, wherein the method further comprises:

determining the memory access consumption of the sub-block based on the size of the bounding box of the transformed sub-block and the filter length;

determining a maximum allowed memory access consumption for the sub- block based on the size of the sub-block and the filter length; and

upon fulfilling a constraint that the determined memory access consumption is not larger than the maximum allowed memory access consumption determining EIF for motion compensation for the block is to be performed.

23. An encoder (20) comprising processing circuitry for carrying out the method according to any one of the claims 1 to 22.

24. A decoder (30) comprising processing circuitry for carrying out the method according to any one of claims 1 to 22.

25. A computer program product comprising program code for performing the method according to any one of claims 1 to 22 when executed on a computer or a processor.

26. A decoder (30), comprising:

one or more processors; and

a non-transitory computer-readable storage medium coupled to the one or more processors and storing instructions for execution by the one or more processors, wherein the instructions, when executed by the one or more processors, configure the decoder to carry out the method according to any one of claims 1 to 22.

27. An encoder (20), comprising:

one or more processors; and

a non-transitory computer-readable storage medium coupled to the one or more processors and storing instructions for execution by the one or more processors, wherein the instructions, when executed by the one or more processors, configure the encoder to carry out the method according to any one of claims 1 to 22.

28. A non-transitory computer-readable medium carrying a program code which, when executed by a computer device, causes the computer device to perform the method of any one of claims 1 to 22.

29. A decoder (30) or an encoder (20) for coding a video sequence, including the enhanced interpolation filter, EIF, for motion compensation, the decoder (30) or encoder (20) comprising, respectively:

a first determining unit (3001, 2001) configured to determine control point motion vectors, CPMVs, for a block according to affine inter-prediction, the block being an affine block or a sub-block of the affine block;

a second determining unit (3003, 2003) configured to determine for a predefined sub-block size a reference area for a sub-block with the predefined sub-block size according to values of the CPMVs;

a comparing unit (3005, 2005) configured to compare the determined reference area with a predefined threshold;

a motion compensation unit (3007, 2007) configured to apply EIF for motion compensation, comprising deriving the pixel-based motion vector field for the block;

wherein if the determined reference area is larger than the threshold, the motion compensation unit (3007, 2007) is configured to derive the pixel-based motion vector field for the block further comprises motion vector clipping,

wherein motion vector clipping range is determined based on motion model of the block and the size of the block.

30. A computer-readable medium comprising a bitstream encoded or decoded by the method of any one of claims 1 to 22.