

where (v_{0x}, v_{0y}) represents the motion vector $Mv0$ at the upper-left corner 110, (v_{1x}, v_{1y}) represents the motion vector $Mv1$ at the upper-right corner 112, and w represents a width of the current block. For block-based affine motion compensation, when the motion vectors $Mv0$ and $Mv1$ of the two control points are decoded, the motion vector of each 4×4 block of the current block 102 can be determined according to Equation (3). In other words, the four-parameter affine motion model for the current block 102 can be specified by the two motion vectors $Mv0$ and $Mv1$ at the two control points. Furthermore, while the upper-left corner and the upper-right corner of the block are used as the two control points, other two control points may also be used.

[0007] A six-parameter affine motion model can be described by Equation (4). In this model, a total of six parameters a_0, a_1, a_2, b_0, b_1 and b_2 and three control points are used. For each pixel $A(x, y)$, the motion vector (v_x, v_y) between this pixel $A(x, y)$ and its corresponding reference pixel $A'(x', y')$ is shown in Equation (5).

$$x' = a_0 + a_1 * x + a_2 * y, \text{ and}$$

$$y' = b_0 + b_1 * x + b_2 * y. \quad (4)$$

$$v_x = (a_1 - 1) * x + a_2 * y + a_0, \text{ and}$$

$$v_y = (b_2 - 1) * y + b_1 * x + b_0. \quad (5)$$

[0008] The motion vector for each pixel predicted by the six-parameter affine motion model is also location dependent. Fig. 1B illustrates an example of affine motion compensation according to the six-parameter affine motion model, where a current block 122 is mapped to a reference block 124 in a reference picture. The correspondences between three corner pixels 130, 132, and 134 of the current block 122 and three corner pixels of the reference block 124 can be determined by the three arrows as shown in Fig. 1B. The six parameters for the affine motion model can be derived based on three known motion vectors $Mv0, Mv1, Mv2$ of the upper-left, upper-right, and lower-left control points of the current block 122. Parameter derivation for the affine motion model is known in the field and the details are omitted here.

[0009] Various implementations of affine Inter mode and affine Merge mode have been discussed, for example, an affine flag is used to indicate whether the affine Inter mode is applied, and this affine flag is signaled for each Inter-coded coding unit (CU) when the CU is equal to or larger than 16×16 . A candidate

first sub-block relative to the current picture, and a width of the neighboring block. A second affine MV of the affine MVs is derived using the first and second MVs, the width of the neighboring block, and a width of the current block.

[0014] In another embodiment, the retrieved MVs includes first, second and third MVs for a six-parameter affine motion model, for example, the three MVs are for first, second and third sub-blocks in the neighboring block, and a first affine MV is derived using at least two of the three MVs, a pixel position of the current block relative to the current picture, a pixel position of the first sub-block relative to the current picture, and a width of the neighboring block. A second affine MV is derived using the first and second motion vectors, the width of the neighboring block, and a width of the current block, and a third affine MV is derived using the first and third motion vectors and at least one of a height of the neighboring block and a height of the current block. In this embodiment, the first, second MVs are replacing MVs replacing original MVs at an upper-left corner and an upper-right corner of the neighboring block, and the third MV is an original MV at a lower-left corner of the neighboring block if the neighboring block is above the current block. The first, second, and third MVs are replacing MVs replacing original MVs at an upper-left corner, upper-right corner, and lower left corner of the neighboring block if the neighboring block is adjacent to an upper-left corner of the current block. The first and third MVs are replacing MVs replacing original MVs at an upper-left corner and lower-left corner of the neighboring block, and the second MV is an original MV at an upper-right corner of the neighboring block if the neighboring block is on the left side of the current block.

[0015] The current block is predicted by the motion compensated blocks by the affine motion vectors in the affine candidate according to a four-parameter affine motion model with two control points or according to a six-parameter affine motion model with three control points. The temporal buffer stores the selective motion vectors, for example, the temporal buffer stores MVs of $M \times N$ block rows above the current block, and M is less than a largest coding unit height divided by N (CTU_height/N). In another embodiment, the temporal buffer stores MVs of $K \times N$ block columns on a left side of the current block, and K is less than a largest coding unit width divided by N (CTU_width/N).

[0016] In some embodiments of the method, the retrieved MVs are original MVs at two or more control points of the neighboring block. The control points include at least two of an upper-left corner, an upper-right corner, a bottom-right corner, and a bottom-left corner of the neighboring block. In an embodiment, the temporal buffer stores MVs of $2 \times N$ block rows and $2 \times N$ block columns, including original MVs of a first neighboring $N \times N$ block row, a first neighboring $N \times N$ block column, a top $N \times N$

block row set and a left-most NxN block column set. The first neighboring NxN block row is a last row in upper neighboring blocks that is closest to the top boundary of the current block, the first neighboring NxN block column is a last column in left neighboring blocks that is closest to the left boundary of the current block, the top NxN block row set is the first rows in the upper neighboring blocks, and the
5 left-most NxN block column set is the first columns in the left neighboring blocks.

[0017] In yet another embodiment, the method comprises receiving input data associated with a current block coded or to be coded in an affine mode, calculating and storing affine parameters for a plurality of coded blocks in the current picture, retrieving the affine parameter of one or more coded block from the temporal buffer corresponding to a neighboring block of the current block to derive an affine candidate
10 including affine MVs. The current block is encoded or decoded by predicting the current block using one or more motion compensated blocks by the derived affine candidate. Each of the affine motion vectors predicts a movement between a point of the current block and a corresponding point of the one or more motion compensated blocks.

[0018] The affine parameters in an embodiment include a horizontal direction motion vector (MV) offset
15 and one motion vector in the neighboring block when the affine motion model is a four-parameter affine motion model using two of an upper-left corner, an upper-right corner, a lower-left corner, and a lower-right corner as the control points, or the affine parameters include a vertical direction motion vector (MV) offset and one motion vector in the neighboring block when the affine motion model is a four-parameter affine motion model using two of an upper-left corner, an upper-right corner, a lower-left
20 corner, and a lower-right corner as control points. The affine parameters in another embodiment include a horizontal direction MV offset, a vertical direction MV offset, and a motion vector in the neighboring block when the affine motion model is a six-parameter affine motion model. An example of the motion vector in the neighboring block is a motion vector at an upper-left corner, upper-right corner, lower-right corner, or lower-left corner of the neighboring block. The affine parameters in another embodiment
25 include a scaled MV offset for the coded block. The affine parameters in yet another embodiment include two or three affine motion vectors representing motion vectors at two or three control points, and the temporal buffer stores the two or three affine motion vectors for the coded block.

[0019] Another embodiment of the method comprises receive input data associated with a current block coded or to be coded in an affine mode, retrieving two or more MVs of a valid neighboring block from a
30 temporal buffer for the current block, deriving an affine candidate including affine MVs using the retrieved two or more MVs of the valid neighboring block, and encoding or decoding the current block

neighboring block rows and two neighboring block columns. If motion vectors at control points of a neighboring block are not stored in the temporal MV buffer as the corresponding sub-blocks are not in the two neighboring block rows and not in the two neighboring block columns, some replacing motion vectors are retrieved to derive affine motion vectors in the affine candidate.

5 **[0039]** In the example shown in Fig. 4A, neighboring coded blocks 41, 42 and 43 of a current block 40 are all coded in affine mode. Replacing motion vectors $V_{B0'}$ and $V_{B1'}$ of sub-blocks 423 and 425 in a second neighboring block row above the current block 40 are used to derive affine motion vectors in the affine candidate instead of original motion vectors V_{B0} and V_{B1} of sub-blocks 422 and 424 in a first row of the neighboring coded block 42. Replacing motion vectors $V_{E0'}$, $V_{E1'}$ and $V_{E2'}$ of sub-blocks 433, 435,
10 437 in either the second neighboring block row above the current block 40 or a second neighboring block column on the left of the current block 40 are used to derive affine motion vectors instead of original motion vectors V_{E0} , V_{E1} and V_{E2} of sub-blocks 432, 434 and 436. Replacing motion vectors $V_{A0'}$ and $V_{A2'}$ of sub-blocks 413 and 415 in the second neighboring block column on the left of the current block 40 are used to derive the affine motion vectors instead of original motion vectors V_{A0} and V_{A2} of sub-block
15 412 and 416 in a first column of the neighboring coded block 41. Original motion vectors V_{B2} and V_{A1} of sub-blocks 426 and 414 are used to derive the affine motion vectors for the current block 40 because these two sub-blocks 426 and 414 are within the first and second neighboring block rows or the first and second neighboring block columns of the current block 40. The derived affine motion vectors in the affine candidate are used to predict a movement between points of the current block 40 and corresponding
20 points of one or more compensated blocks using an affine motion model.

[0040] Fig. 4B is an alternative solution to the affine candidate derivation method illustrated in Fig. 4A. In Fig. 4B, when deriving an affine candidate for encoding or decoding a current block 44, all the original motion vectors at control points of a neighboring affine-coded block are replaced by motion vectors of other sub-blocks in first and second neighboring block rows and first and second neighboring block
25 columns of the current block 44. The control points for an affine-coded block predicted by a six-parameter affine motion model include upper-left corner, upper-right corner, and lower-left corner of the affine-coded block. Replacing motion vectors $V_{B0'}$, $V_{B1'}$ and $V_{B2'}$ of sub-blocks 463, 465 and 467 in a neighboring affine-coded block 46 are used to derive the affine motion vectors instead of using original motion vectors V_{B0} , V_{B1} and V_{B2} of sub-blocks 462, 464 and 466. Replacing motion vectors $V_{E0'}$, $V_{E1'}$ and
30 $V_{E2'}$ of sub-blocks 473, 475 and 477 in a neighboring affine-coded block 47 are used to derive the affine motion vectors instead of original motion vectors V_{E0} , V_{E1} and V_{E2} of sub-blocks 472, 474 and

476. Replacing motion vectors $V_{A0'}$, $V_{A1'}$ and $V_{A2'}$ of sub-blocks 453, 455 and 457 in a neighboring affine-coded block 45 are used to derive the affine motion vectors instead of original motion vectors V_{A0} , V_{A0} and V_{A2} of sub-blocks 452, 454 and 456. In general, other positions in the M neighboring block rows and the K block columns may be used for affine candidate derivation in this embodiment.

- 5 **[0041]** Without loss of generality, only the affine candidate derivation method illustrated in Fig. 4A is further described in the following. In a first example, an affine candidate including three affine motion vectors $Mv0$, $Mv1$ and $Mv2$ are derived for predicting the current block 40 using a six-parameter affine motion model. In the first example, the affine motion vectors of the affine candidate are derived from the neighboring affine-coded block 42. Affine motion vector $Mv0 = (V0_x, V0_y)$ for a first control point at an upper-left corner is derived by Equation (6).

$$\begin{aligned} V0_x &= V_{B0'_x} + (V_{B2_x} - V_{B0'_x}) * (\text{posCurPU_Y} - \text{posB0'_Y}) / (2*N) + \\ & (V_{B1'_x} - V_{B0'_x}) * (\text{posCurPU_X} - \text{posB0'_X}) / \text{RefPU_width}, \\ V0_y &= V_{B0'_y} + (V_{B2_y} - V_{B0'_y}) * (\text{posCurPU_Y} - \text{posB0'_Y}) / (2*N) \\ & + (V_{B1'_y} - V_{B0'_y}) * (\text{posCurPU_X} - \text{posB0'_X}) / \text{RefPU_width}; \end{aligned} \quad (6)$$

- 15 where motion vectors $V_{B0'} = (V_{B0'_x}, V_{B0'_y})$, $V_{B1'} = (V_{B1'_x}, V_{B1'_y})$, and $V_{B2} = (V_{B2_x}, V_{B2_y})$ are the three motion vectors retrieved from M neighboring $N \times N$ block rows and K neighboring $N \times N$ block columns. It is also to be understood that these motion vectors can be replaced by corresponding motion vectors of any other selected sub-blocks in the M neighboring $N \times N$ block rows and K neighboring $N \times N$ block columns of the current block. A coordinate $(\text{posCurPU_X}, \text{posCurPU_Y})$ represents a pixel position of an upper-left corner sample of the current block 40 relative to an upper-left corner sample of the current picture. A coordinate $(\text{posB0'_X}, \text{posB0'_Y})$ represents a pixel position of an upper-left corner sample of the sub-block 422 relative to the upper-left corner sample of the current picture. RefPU_width represents the width of the neighboring block 42. Affine motion vectors $Mv1 = (V1_x, V1_y)$ and $Mv2 = (V2_x, V2_y)$ for second and third control points at upper-right and lower-left corner respectively are derived by Equation (7).

$$V1_x = V0_x + (V_{B1'_x} - V_{B0'_x}) * \text{PU_width} / \text{RefPU_width},$$

$$V1_y = V0_y + (V_{B1'_y} - V_{B0'_y}) * \text{PU_width} / \text{RefPU_width};$$

$$V2_x = V0_x + (V_{B2_x} - V_{B0'_x}) * \text{PU_height} / (2*N),$$

[0045] For a six-parameter affine motion model, the motion vector $Mv0 = (V_{0x}, V_{0y})$ of a top-left $N \times N$ sub-block, the motion vector $Mv1 = (V_{1x}, V_{1y})$ of a top-right $N \times N$ sub-block, and the motion vector $Mv2 = (V_{2x}, V_{2y})$ of a bottom-left $N \times N$ sub-block are used to derive motion vectors of all $N \times N$ sub-blocks in the current block. The motion vector field of each pixel $A(x,y)$ in the current block may be derived based on the three motion vectors $Mv0$, $Mv1$ and $Mv2$ according to Equation (10).

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{2x} - v_{0x})}{h}y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x + \frac{(v_{2y} - v_{0y})}{h}y + v_{0y} \end{cases} \quad (10)$$

[0046] Similarly, for deriving a motion vector (V_x, V_y) of an $N \times N$ sub-block at position (x,y) according to the six-parameter affine motion model, a horizontal direction MV offset H_MV_offset and a vertical direction MV offset V_MV_offset are shown in Equation (11).

$$\begin{aligned} H_MV_offset &= (v_{1x} - v_{0x}) * N / w, (v_{1y} - v_{0y}) * N / w; \\ V_MV_offset &= (v_{2x} - v_{0x}) * N / h, (v_{2y} - v_{0y}) * N / h; \end{aligned} \quad (11)$$

where w and h in Equations (9) and (11) are the width and height of the current block coded in affine mode.

[0047] In the case of assigning a motion vector at a center pixel of an $N \times N$ sub-block as a motion vector of the control point, the denominator in Equations (6) and (8) is decreased by N . For example, Equation (6) may be rewritten as follow.

$$\begin{aligned} V_{0_x} &= V_{B0'_x} + (V_{B2_x} - V_{B0'_x}) * (\text{posCurPU}_Y - \text{posB0}'_Y) / (N) + \\ & (V_{B1'_x} - V_{B0'_x}) * (\text{posCurPU}_X - \text{posB0}'_X) / (\text{RefPU_width} - N) \\ V_{0_y} &= V_{B0'_y} + (V_{B2_y} - V_{B0'_y}) * (\text{posCurPU}_Y - \text{posB0}'_Y) / (N) \\ & + (V_{B1'_y} - V_{B0'_y}) * (\text{posCurPU}_X - \text{posB0}'_X) / (\text{RefPU_width} - N) \end{aligned} \quad (12)$$

[0048] The third embodiment stores affine parameters such as the horizontal and vertical direction MV offsets for the coded block. The coded block may be either a fixed size $M \times M$ block or a coding unit (CU). The size of the fixed size $M \times M$ block may depend on the minimum size allowed for applying affine motion prediction. In an example, M is 8 if the smallest affine Inter mode or affine Merge mode block size is 8×8 . For each $M \times M$ block or for each CU, the affine parameters including a horizontal

information such as the Merge index and the MVD. The data associated with the side information are also provided to Entropy Encoder 934. When motion compensation prediction mode is used, a reference picture or pictures have to be reconstructed at the encoder end as well. The transformed and quantized residual signal is processed by Inverse Quantization (IQ) 922 and Inverse Transformation (IT) 924 to recover the prediction residual signal of the reference picture or pictures. As shown in Fig. 9, the prediction residual signal is recovered by adding back to the selected predictor at Reconstruction (REC) 926 to produce reconstructed video data. The reconstructed video data may be stored in Reference Picture Buffer (Ref. Pict. Buffer) 932 and used for prediction of other pictures. The reconstructed video data from REC 926 may be subject to various impairments due to the encoding processing, consequently, in-loop processing Deblocking Filter (DF) 928 and Sample Adaptive Offset (SAO) 930 is applied to the reconstructed video data before storing in the Reference Picture Buffer 932 to further enhance picture quality. DF information from DF 928 and SAO information from SAO 930 are also provided to Entropy Encoder 934 for incorporation into the encoded video bitstream. The temporal MV buffer for storing motion vectors for affine candidate derivation may be implemented in Ref. Pict. Buffer 932 or any other memory coupling with the Affine Prediction 912.

[0064] A corresponding Video Decoder 1000 for the Video Encoder 900 of Fig. 9 is shown in Fig. 10. The encoded video bitstream is the input to the Video Decoder 1000 and is decoded by Entropy Decoder 1010 to recover the transformed and quantized residual signal, DF and SAO information, and other system information. The decoding process of Decoder 1000 is similar to the reconstruction loop at the Encoder 900, except Decoder 1000 only requires Motion Compensation (MC) 1014 in Affine Prediction 1014. Affine Prediction 1014 includes Affine Inter Prediction 10142 and Affine Merge Prediction 10144. Blocks coded in affine Inter mode is decoded by Affine Inter Prediction 10142 and blocks coded in affine Merge mode is decoded by Affine Merge Prediction 10144. A final affine candidate is selected for a block coded in affine Inter mode or affine Merge mode, and one or more compensated blocks are derived according to the final affine candidate. The final affine candidate may be derived according to one of the embodiments of the present invention with a temporal MV buffer storing selective MVs less than all MVs of coded blocks in the current picture. Switch 1016 selects intra predictor from Intra Prediction 10512 or affine predictor from Affine Prediction 1014 according to decoded mode information. The transformed and quantized residual signal is recovered by Inverse Quantization (IQ) 1020 and Inverse Transformation (IT) 1022. The recovered transformed and quantized residual signal is reconstructed by adding back the predictor in REC 1018 to produce reconstructed video. The reconstructed video is further processed by DF 1024 and SAO 1026 to generate final decoded

CLAIMS

1. A method of video coding with affine motion compensation in a video coding system, comprising:

receiving input data associated with a current block in a current picture at a video encoder or
 5 receiving a video bitstream corresponding to compressed data including the current block in the current picture at a video decoder, wherein the current block is coded or to be coded in an affine mode according to an affine motion model;

retrieving two or more motion vectors of a neighboring block from a temporal buffer for the current block, wherein at least one of the retrieved motion vectors is not corresponding to a sub-block in a first
 10 neighboring NxN block row closest to a top boundary of the current block or a first neighboring NxN block column closest to a left boundary of the current block, wherein the temporal buffer stores selective motion vectors less than all motion vectors of previous coded blocks in the current picture, and NxN is a block size for storing one motion vector in the temporal buffer;

deriving an affine candidate including affine motion vectors using the retrieved two or more motion
 15 vectors of the neighboring block; and

encoding or decoding the current block by predicting the current block using one or more motion compensated blocks by the derived affine candidate, wherein the affine motion vectors predict a movement between a point of the current block and a corresponding point of the one or more motion compensated blocks.

20 2. The method of Claim 1, wherein the temporal buffer stores motion vectors of 2 NxN block rows and 2 NxN block columns, and the 2 NxN block rows include the first neighboring NxN block row and a second neighboring NxN block row that are closest to the top boundary of the current block, and the 2 NxN block columns include the first neighboring NxN block column and a second neighboring NxN block column closest to the left boundary of the current block.

25 3. The method of Claim 1, wherein the retrieved two or more motion vectors includes first and second motion vectors, the first motion vector is a replacing motion vector replacing an original motion vector at an upper-left corner of the neighboring block and the second motion vector is a replacing motion vector replacing an original motion vector at an upper-right corner of the neighboring block if the neighboring block is above the current block or adjacent to an upper-left corner of the current block;
 30 wherein the first motion vector is a replacing motion vector replacing an original motion vector at an

upper-left corner of the neighboring block and the second motion vector is an original motion vector at an upper-right corner of the neighboring block if the neighboring block is on the left side of the current block.

4. The method of Claim 1, wherein the retrieved two or more motion vectors includes first and second motion vectors of first and second sub-blocks in the neighboring block, said deriving an affine candidate further comprises:

deriving a first affine motion vector of the affine motion vectors using the first and second motion vectors, a pixel position of the current block relative to the current picture, a pixel position of the first sub-block relative to the current picture, and a width of the neighboring block; and

10 deriving a second affine motion vector using the first and second motion vectors, the width of the neighboring block, and a width of the current block.

5. The method of Claim 1, wherein the retrieved two or more motion vectors includes first, second and third motion vectors of first, second and third sub-blocks in the neighboring block, wherein deriving the affine candidate further comprises:

15 deriving a first affine motion vector of the affine motion vectors using at least two of the first, second and third motion vectors, a pixel position of the current block relative to the current picture, a pixel position of the first sub-block relative to the current picture, and a width of the neighboring block;

deriving a second affine motion vector using the first and second motion vectors, the width of the neighboring block, and a width of the current block; and

20 deriving a third affine motion vector using the first and third motion vectors and at least one of a height of the neighboring block and a height of the current block.

6. The method of Claim 2, wherein the retrieved two or more motion vectors includes first, second and third motion vectors, the first motion vector is a replacing motion vector replacing an original motion vector at an upper-left corner of the neighboring block, the second motion vector is a replacing motion vector replacing an original motion vector at an upper-right corner of the neighboring block, and the third motion vector is an original motion vector at a lower-left corner of the neighboring block if the neighboring block is above the current block; wherein the first motion vector is a replacing motion vector replacing an original motion vector at an upper-left corner of the neighboring block, the second motion vector is a replacing motion vector replacing an original motion vector at an upper-right corner of the neighboring block, and the third motion vector is a replacing motion vector replacing an original motion vector at a lower-left corner of the neighboring block if the neighboring block is adjacent to an upper-left

corner of the current block; and wherein the first motion vector is a replacing motion vector replacing an original motion vector at an upper-left corner of the neighboring block, the second motion vector is an original motion vector at an upper-right corner of the neighboring block, and the third motion vector is a replacing motion vector replacing an original motion vector at a lower-left corner of the neighboring block if the neighboring block is on the left side of the current block.

7. The method of Claim 1, wherein the current block is predicted by said one or more motion compensated blocks by the affine motion vectors in the affine candidate according to a four-parameter affine motion model with two control points or according to a six-parameter affine motion model with three control points.

8. The method of Claim 1, wherein the temporal buffer stores M $N \times N$ block rows above the current block, and M is less than a largest coding unit height divided by N .

9. The method of Claim 1, wherein the temporal buffer stores K $N \times N$ block columns on a left side of the current block, and K is less than a largest coding unit width divided by N .

10. The method of Claim 1, wherein the retrieved two or more motion vectors are original motion vectors at two or more control points of the neighboring block.

11. The method of Claim 10, wherein the control points of the neighboring block include at least two of an upper-left corner, an upper-right corner, a bottom-right corner, and a bottom-left corner of the neighboring block.

12. The method of Claim 10, wherein the temporal buffer stores original motion vectors of a first neighboring $N \times N$ block row, a first neighboring $N \times N$ block column, a top $N \times N$ block row set and a left-most $N \times N$ block column set, where the first neighboring $N \times N$ block row is a last row in at least one upper neighboring block that is closest to the top boundary of the current block, the first neighboring $N \times N$ block column is a last column in at least one left neighboring block that is closest to the left boundary of the current block, the top $N \times N$ block row set comprises a first row in the at least one upper neighboring block, and the left-most $N \times N$ block column set comprises a first column in the at least one left neighboring block.

13. A method of video coding with affine motion compensation in a video coding system, comprising:

receiving input data associated with a current block in a current picture at a video encoder or

receiving a video bitstream corresponding to compressed data including the current block in the current picture at a video decoder, wherein the current block is coded or to be coded in an affine mode according to an affine motion model;

5 retrieving two or more motion vectors of a valid neighboring block from a temporal buffer for the current block, wherein the temporal buffer stores motion vectors of one neighboring $N \times N$ block row and one neighboring $N \times N$ block column of the current block, and $N \times N$ is a block size for storing one motion vector in the temporal buffer;

deriving an affine candidate including affine motion vectors using the retrieved two or more motion vectors of the valid neighboring block; and

10 encoding or decoding the current block by predicting the current block using one or more motion compensated blocks by the derived affine candidate, wherein the affine motion vectors predict a movement between a point of the current block and a corresponding point of the one or more motion compensated blocks.

14. The method of Claim 13, wherein the retrieved two or more motion vectors correspond to two motion vectors, one of said two motion vectors is an original motion vector at a lower-left corner of the valid neighboring block and the other one of said two motion vectors is an original motion vector at a lower-right corner of the valid neighboring block if the valid neighboring block is above the current block; wherein one of said two motion vectors is an original motion vector at an upper-right corner of the valid neighboring block and the other one of said two motion vectors is an original motion vector at a lower-right corner of the valid neighboring block if the valid neighboring block is on the left side of the current block.

15. The method of Claim 13, wherein the current block is predicted by said one or more motion compensated blocks by the affine motion vectors in the affine candidate according to a four-parameter affine motion model with two control points.

25 16. An apparatus of video coding with affine motion compensation in a video coding system, the apparatus comprising one or more electronic circuits configured for:

receiving input data associated with a current block in a current picture at a video encoder or receiving a video bitstream corresponding to compressed data including the current block in the current picture at a video decoder, wherein the current block is coded or to be coded in an affine mode according to an affine motion model;

30 retrieving two or more motion vectors of a neighboring block from a temporal buffer for the current

block, wherein at least one of the retrieved motion vectors is not corresponding to a sub-block in a first neighboring NxN block row closest to a top boundary of the current block or a first neighboring NxN block column closest to a left boundary of the current block, wherein the temporal buffer stores selective motion vectors less than all motion vectors of previous coded blocks in the current picture, and NxN is a
 5 block size for storing one motion vector in the temporal buffer;

deriving an affine candidate including affine motion vectors using the retrieved two or more motion vectors of the neighboring block; and

encoding or decoding the current block by predicting the current block using one or more motion compensated blocks by the derived affine candidate, wherein the affine motion vectors predict a
 10 movement between a point of the current block and a corresponding point of the one or more motion compensated blocks.

17. An apparatus of video coding with affine motion compensation in a video coding system, the apparatus comprising one or more electronic circuits configured for:

receiving input data associated with a current block in a current picture at a video encoder or
 15 receiving a video bitstream corresponding to compressed data including the current block in the current picture at a video decoder, wherein the current block is coded or to be coded in an affine mode according to an affine motion model;

retrieving two or more motion vectors of a valid neighboring block from a temporal buffer for the current block, wherein the temporal buffer stores motion vectors of one neighboring NxN block row and
 20 one neighboring NxN block column of the current block, and NxN is a block size for storing one motion vector in the temporal buffer;

deriving an affine candidate including affine motion vectors using the retrieved two or more motion vectors of the valid neighboring block; and

encoding or decoding the current block by predicting the current block using one or more motion compensated blocks by the derived affine candidate, wherein the affine motion vectors predict a
 25 movement between a point of the current block and a corresponding point of the one or more motion compensated blocks.