CLAIMS:

1. A method for decoding a video sequence, the method comprising:

decoding the video sequence using one of a 4:4:4 chroma format and a 4:2:2 chroma format,

when decoding the video sequence using the 4:4:4 chroma format, copying an affine motion vector of one 4x4 luma block using an operation other than an averaging operation and associating the affine motion vector to a co-located 4x4 chroma block,

when decoding the video sequence using the 4:2:2 chroma format, associating each 4x4 chroma block with two 4x4 co-located luma blocks such that an affine motion vector of one 4x4 chroma block is an average of the motion vectors of the two co-located luma blocks,

wherein a maximum allowed transform size is the same for different color components, and

wherein when decoding the video sequence using the 4:2:2 chroma format, a maximum vertical transform size is the same among different color components, and a maximum horizontal transform size for color components is half of a maximum horizontal transform size for luma components.

2. The method of claim 1, further comprising,

regardless of the chroma format, dividing a current 4x4 chroma block into four 2x2 sub-blocks,

deriving a first affine motion vector of a co-located luma block for a top-left 2x2 chroma sub-block,

deriving a second affine motion vector of the co-located luma block for a bottom-right 2x2 chroma sub-block, and

deriving an affine motion vector of the current 4x4 chroma block using the average of the first affine motion vector and the second affine motion vector.

3. The method of claim 1, further comprising

aligning an interpolation filter used for motion compensation between luma and chroma components.

4. The method of claim 3, wherein when a video sequence is input using a 4:2:0 chroma format, applying an 8-tap interpolation filter for luma components and chroma components.

5. The method of claim 1, further comprising,

coding as three separate trees, components Y, Cb, and Cr, and

wherein each tree of the three separate trees codes one component of the components Y, Cb, and Cr.

6. The method of claim 5, wherein the coding as three separate trees is performed for an I slice or an I tile group.

7. The method of claim 1, wherein at least one of a Position-Dependent Predictor combination (PDPC), a Multiple Transform Selection (MTS), a Non-Separable Secondary Transform (NSST), an Intra-Sub Partitioning (ISP), and a Multiple reference line (MRL) intra prediction is applied to both a luma component and a chroma component.

8. The method of claim 7, wherein

when the Multiple reference line (MRL) intra prediction is applied to both the luma component and the chroma component, and when decoding the video sequence is performed using the 4:4:4 chroma format, the method further comprises selecting an Nth reference for intra prediction, and using a same reference line without explicit signaling for chroma components,

when the Intra-Sub Partitioning (ISP) is applied to both the luma component and the chroma component, the method further comprises applying the Intra-Sub Partitioning (ISP) at a block level for a current block for components Y, Cb, and Cr, and

when different trees are used for different color components, the method further comprises implicitly deriving coding parameters for U and V components from collocated Y components without signaling.

9. A device for decoding a video sequence, the device comprising;

at least one memory configured to store program code;

at least one processor configured to read the program code and operate as instructed by the program code, the program code including:

first decoding code configured to cause the at least one processor to decode the video sequence using at least one of a 4:4:4 chroma format and a 4:2:2 chroma format,

wherein when the first decoding code is configured to cause the at least one processor to decode the video sequence using the 4:4:4 chroma format, the first decoding code further comprises code configured to cause the at least one processor to copy an affine motion vector of one 4x4 luma block using an operation other than an averaging operation and to associate the affine motion vector to a co-located 4x4 chroma block,

wherein when the first decoding code is configured to cause the at least one processor to decode the video sequence using the 4:2:2 chroma format, the first decoding code further comprises code configured to cause the at least one processor to associate each 4x4 chroma block with two 4x4 co-located luma blocks such that an affine motion vector of one 4x4 chroma block is an average of the motion vectors of the two co-located luma blocks,

wherein the first decoding code further comprises code configured to cause the at least one processor to allow a maximum transform size to be the same for different color components, and

wherein when the first decoding code is configured to cause the at least one processor to decode the video sequence using the 4:2:2 chroma format, the first decoding code further comprises code configured to cause the at least one processor to set a maximum vertical size to be the same among different color components, and set a maximum horizontal transform size for color components to be half of a maximum horizontal transform size for luma components.

10. The device of claim 9, wherein the first decoding code further comprises code configured to cause the at least one processor to:

divide a current 4x4 chroma block into four 2x2 sub-blocks,

derive a first affine motion vector of a co-located luma block for a top-left 2x2 chroma sub-block,

derive a second affine motion vector of the co-located luma block for a bottom-right 2x2 chroma sub-block, and

derive an affine motion vector of the current 4x4 chroma block using the average of the first affine motion vector and the second affine motion vector.

11. The device of claim 9, wherein the first decoding code further comprises code configured to cause the at least one processor to:

align an interpolation filter used for motion compensation between luma and chroma components.

12. The device of claim 11, wherein when the first r decoding code is configured to cause the at least one processor to decode the video sequence using the 4:2:0 chroma format, the first decoding code further comprises code configured to cause the at least one processor to

apply an 8-tap interpolation filter for luma components and chroma components.

13. The device of claim 9, wherein the first decoding code further comprises code configured to cause the at least one processor to:

code as three separate trees, components Y, Cb, and Cr, and

wherein each tree of the three separate trees codes one component of the components Y, Cb, and Cr.

14. The device of claim 13, wherein the configuration to code as three separate trees is configured to be performed for an I slice or an I tile group.

15. The device of claim 9, wherein the first decoding code further comprises code configured to cause the at least one processor to:

apply at least one of a Position-Dependent Predictor combination (PDPC), a Multiple Transform Selection (MTS), a Non-Separable Secondary Transform (NSST), an Intra-Sub Partitioning (ISP), and a Multiple reference line (MRL) intra prediction to both a luma component and a chroma component.

16. A non-transitory computer-readable medium storing program code, the program code comprising one or more instructions that, when executed by one or more processors of a device, cause the one or more processors to perform the method according to any of claims 1 to 8.