CLAIMS

1. A method of coding video data, the method comprising:

coding a first motion vector difference (MVD) representing a first difference between a first motion vector of a current block of video data predicted using affine prediction and a first motion vector predictor (MVP) for the first motion vector;

predicting a second MVD from the first MVD for a second motion vector of the current block, the second MVD representing a second difference between the second motion vector and a second MVP for the second motion vector, the second motion vector being different than the first motion vector; and

coding the current block using affine prediction according to the first motion vector and the second motion vector.

2. The method of claim 1, wherein the first MVD includes a horizontal component (MVDx1) and a vertical component (MVDy1), the first MVP includes a horizontal component (MVPx1) and a vertical component (MVPy1), the first motion vector includes a horizontal component (MVx1) and a vertical component (MVy1), MVDx1=MVx1-MVPx1, and MVDy1 = MVy1-MVPy1.

3. The method of claim 1, wherein coding the current block comprises coding the current block according to a four-parameter affine model.

4. The method of claim 1, further comprising predicting a third MVD from at least one of the first MVD or the second MVD for a third motion vector of the current block, wherein coding the current block comprises coding the current block using affine prediction according to the first motion vector, the second motion vector, and the third motion vector.

5. The method of claim 4, wherein coding the current block comprises coding the current block according to a six-parameter affine model.

6. The method of claim 1, wherein the first motion vector originates from an upper-left

corner of the current block, and wherein the second motion vector originates from an upper-right corner of the current block.

7. The method of claim 1, further comprising coding data defining control points for the first motion vector and the second motion vector.

8. The method of claim 1, further comprising determining control points for the first motion vector and the second motion vector based on a shape of the current block.

9. The method of claim 1, further comprising implicitly deriving control points for the first motion vector and the second motion vector.

10. The method of claim 1, further comprising coding data representing MVD’2 for the second MVD, wherein MVD’2 represents a residual value for the second MVD relative to the first MVD.

11. The method of claim 10, wherein the first MVD is MVD1, the second MVD is MVD2, w is a weighting value, and MVD’2 = MVD1-w*MVD2.

12. The method of claim 10, wherein the first MVD is MVD1, the second MVD is MVD2, and MVD’2 = MVD2-((MVD1+1)>>1) for a weighting value of 0.5, wherein “>>” is a bitwise right-shift operator.

13. The method of claim 10, wherein the first MVD is MVD1, the second MVD is MVD2, and MVD’2 = MVD2-((MVD1+2)>>2) for a weighting value of 0.25, wherein “>>” is a bitwise right-shift operator.

14. The method of claim 1, further comprising determining to predict the second MVD from the first MVD, wherein predicting the second MVD from the first MVD comprises predicting the second MVD from the first MVD in response to determining to predict the second MVD from the first MVD.

15. The method of claim 14, wherein determining to predict the second MVD from the first MVD comprises determining to predict the second MVD from the first MVD based on a shape of the current block.

16. The method of claim 14, wherein determining to predict the second MVD from the first MVD comprises determining to predict the second MVD from the first MVD based on a motion prediction method for the current block.

17. The method of claim 16, wherein determining to predict the second MVD from the first MVD comprises determining that the motion prediction method is an affine merge mode.

18. The method of claim 1, wherein the second MVD includes a horizontal component ($MVD2^x$) and a vertical component ($MVD2^y$), and wherein predicting the second MVD comprises predicting $MVD2^x$ differently than predicting $MVD2^y$.

19. The method of claim 1, further comprising generating a third MVP for a third motion vector of the current block from at least one of the first MVD or the second MVD.

20. The method of claim 1, wherein the first MVD is MVD1, the method further comprising:

determining a second intermediate MVP (MVP'2) for the second motion vector from a motion vector of one or more neighboring blocks to the current block; and

generating the second MVP (MVP2) for the second motion vector from MVP'2 and MVD1.

21. The method of claim 1, further comprising generating the second MVP for the second motion vector from the first MVD and a third MVD for a third motion vector of the current block, wherein the second MVP is MVP2, the first MVD is MVD1, the third MVD is MVD3, the method further comprising determining a second intermediate MVP (MVP'2) for the second motion vector from a motion vector of one or more neighboring blocks to the current block, wherein generating MVP2 comprises generating MVP2 as

MVP2=MVP’2+((MVD1+MVD3)>>1), wherein “>>” is a bitwise right-shift operator.

22. The method of claim 1, wherein coding the current block comprises decoding the current block, comprising:

adding the first MVD to the first MVP to reconstruct the first motion vector;

determining the second MVP for the second motion vector;

reconstructing the second MVD using the prediction from the first MVD;

adding the second MVD to the second MVP to reconstruct the second motion vector;

forming a prediction block for the current block using the first motion vector and the second motion vector;

decoding a residual block for the current block; and

adding the residual block and the prediction block to reconstruct the current block.

23. The method of claim 1, wherein coding the current block comprises encoding the current block, comprising:

subtracting the first MVP from the first motion vector to generate the first MVD;

determining the second MVP for the second motion vector;

subtracting the second motion vector from the second MVP to generate the second MVD;

encoding the first MVD;

encoding data representative of the second MVD predicted from the first MVD;

forming a prediction block for the current block using the first motion vector and the second motion vector;

subtracting the prediction block from the current block to generate a residual block; and

encoding the residual block.

24. A device for coding video data, the device comprising:

a memory configured to store video data; and

one or more processors implemented in circuitry and configured to:

code a first motion vector difference (MVD) representing a difference between a first motion vector of a current block of video data predicted using affine prediction and a first motion vector predictor (MVP) for the first motion vector;

predict a second MVD from the first MVD for a second motion vector of the current block, the second MVD representing a second difference between the second motion vector and a second MVP for the second motion vector, the second motion vector being different than the first motion vector; and

code the current block using affine prediction according to the first motion vector and the second motion vector.

25. The device of claim 24, wherein the one or more processors are configured to:

add the first MVD to the first MVP to reconstruct the first motion vector;

determine the second MVP for the second motion vector;

reconstruct the second MVD using the prediction from the first MVD;

add the second MVD to the second MVP to reconstruct the second motion vector;

form a prediction block for the current block using the first motion vector and the second motion vector;

decode a residual block for the current block; and

add the residual block and the prediction block to reconstruct the current block.

26. The device of claim 24, wherein the one or more processors are configured to:

subtract the first MVP from the first motion vector to generate the first MVD;

determine the second MVP for the second motion vector;

subtract the second motion vector from the second MVP to generate the second MVD;

encode the first MVD;

encode data representative of the second MVD predicted from the first MVD;

form a prediction block for the current block using the first motion vector and the second motion vector;

subtract the prediction block from the current block to generate a residual block; and

encode the residual block.

27. The device of claim 24, further comprising a display configured to display decoded video data.

28. The device of claim 24, wherein the device comprises one or more of a camera, a computer, a mobile device, a broadcast receiver device, or a set-top box.

29. A non-transitory computer-readable storage medium having stored thereon instructions that, when executed, cause a processor of a device for coding video data to:

code a first motion vector difference (MVD) representing a difference between a first motion vector of a current block of video data predicted using affine prediction and a first motion vector predictor (MVP) for the first motion vector;

predict a second MVD from the first MVD for a second motion vector of the current block, the second MVD representing a second difference between the second motion vector and a second MVP for the second motion vector, the second motion vector being different than the first motion vector; and

code the current block using affine prediction according to the first motion vector and the second motion vector.

30. A device for coding video data, the device comprising:

means for coding a first motion vector difference (MVD) representing a difference between a first motion vector of a current block of video data predicted using affine prediction and a first motion vector predictor (MVP) for the first motion vector;

means for predicting a second MVD from the first MVD for a second motion vector of the current block, the second MVD representing a second difference between the second motion vector and a second MVP for the second motion vector, the second motion vector being different than the first motion vector; and

means for coding the current block using affine prediction according to the first motion vector and the second motion vector.