CLAIMS

1. A method of coding video data, the method comprising: determining that a block of video data has one of: 1) a width of 4 pixels and a height of more than 8 pixels or 2) a height of 4 pixels and a width of more than 8 pixels; in response to determining that the block of video data has the one of 1) the width of 4 pixels and the height of more than 8 pixels or 2) the height of 4 pixels and the width of more than 8 pixels, determining that decoder-side motion vector refinement (DMVR) is not available to be used for the block; and in response to determining that DMVR is not available to be used for the block, coding the block without performing DMVR for the block.

2. The method of claim 1, wherein the block comprises a first block, the method further comprising: determining that a second block of video data has a size of at least 8×N or N×8, wherein N is an integer value greater than 8; in response to determining that the second block of video data has the size of at least 8×N or N×8, determining whether to code the second block using DMVR; in response to determining to code the second block using DMVR, coding the second block using DMVR.

3. The method of claim 2, wherein coding the second block comprises: retrieving padded filter input samples of a reference picture for the second block when the second block is to be predicted using bi-directional optical flow (BIO); calculating one or more gradient values for one or more samples of the second block using the padded filter input samples; calculating one or more refined motion vectors for the one or more samples of the second block using the gradient values; and generating a prediction block for the second block using the one or more refined motion vectors.

4. The method of claim 3, wherein the second block comprises a width of w samples and a height of h samples, and wherein retrieving the padded filter input samples comprises retrieving (w+7)*(h+7) samples from the reference picture and padding the retrieved samples to a size of (w+7+2d)*(h+7+2d), wherein d represents a pre-defined maximum displacement vector.

5. The method of claim 3, wherein calculating the one or more gradient values comprises calculating the one or more gradient values using a gradient filter having a length of L, wherein the second block comprises a width of w samples and a height of h samples, and wherein retrieving the padded filter input samples comprises retrieving (w+7)*(h+7) samples from the reference picture and padding the retrieved samples to a size of (w+7+2d+2s)*(h+7+2d+2s), wherein d represents a pre-defined maximum displacement vector, and wherein s represents one-half of L.

6. The method of claim 3, wherein the second block comprises a width of w samples and a height of h samples, and wherein retrieving the padded filter input samples comprises: retrieving (w+7)*(h+7) samples from the reference picture; horizontally padding the retrieved samples to a size of (w+11)*(h+7); horizontally interpolating the padded samples to a size of (w+11−t)*(h+7); and vertically padding the horizontally interpolated padded samples to a size of (w+11−t)*(h+11).

7. The method of claim 3, further comprising decoding an initial motion vector for the second block, wherein calculating the one or more refined motion vectors comprises calculating the one or more refined motion vectors using the initial motion vector.

8. The method of claim 1, wherein coding the block comprises: decoding a motion vector for the block; calculating a prediction block for the block using the motion vector; decoding a residual block for the block; and combining the prediction block with the residual block to decode the block.

9. The method of claim 1, wherein coding the block comprises: generating a motion vector for the block; calculating a prediction block for the block using the motion vector; generating a residual block for the block representing differences between the block and the prediction block; and encoding the residual block and the motion vector to encode the block.

10. The method of claim 1, wherein coding the block comprises: retrieving a refined motion vector of an above-right neighboring block to the block as a motion vector predictor for a motion vector of the block; generating the motion vector for the current block using the motion vector predictor; and generating a prediction block for the block using the motion vector.

11. The method of claim 1, wherein coding the block comprises: retrieving a refined motion vector of a virtual pipeline data unit (VPDU) neighboring the block as a motion vector predictor for a motion vector of the block, wherein the VPDU comprises one of four VPDUs of a coding tree unit (CTU), wherein the four VPDUs are evenly split quarter blocks of the CTU; generating the motion vector for the current block using the motion vector predictor; and generating a prediction block for the block using the motion vector.

12. The method of claim 1, wherein coding the block comprises: generating a history-based motion vector prediction (HMVP) list including one or more motion vector predictor candidates for a motion vector of the block; in response to decoding a previous block using a refined motion vector, wherein an affine flag for the previous block indicates that affine motion is disabled for the previous block, preventing insertion of the refined motion vector into the HMVP list; coding the motion vector for the block using one of the motion vector predictor candidates of the HMVP list; and generating a prediction block for the block using the motion vector.

13. The method of claim 1, wherein coding the block comprises: generating a history-based motion vector prediction (HMVP) list including one or more motion vector predictor candidates for a motion vector of the block; in response to decoding a previous block using a refined control point motion vector, wherein an affine flag for the previous block indicates that affine motion is enabled for the previous block, preventing insertion of the refined control point motion vector into the HMVP list; coding the motion vector for the block using one of the motion vector predictor candidates of the HMVP list; and generating a prediction block for the block using the motion vector.

14. The method of claim 1, wherein coding the block comprises: determining an affine merge candidate for a motion vector of the block from a causal neighboring sub-block to the block, wherein determining the affine merge candidate comprises: when the causal neighboring sub-block is in an above-neighboring coding tree unit (CTU) row to a CTU row including the block, selecting a refined motion vector of the causal neighboring sub-block as the affine merge candidate; or when the causal neighboring sub-block is in the CTU row including the block, selecting an unrefined motion vector used to generate the refined motion vector as the affine merge candidate; coding the motion vector using the affine merge candidate; and generating a prediction block for the block using the motion vector.

15. The method of claim 1, wherein coding the block comprises: determining a positive integer scalar (s) value to be applied to weight values used to calculate a cost value using a cost function; calculating the cost value using the cost function and the s value for a sample of a prediction block of the block; and generating the prediction block using the cost value according to at least one of generalized bi-prediction (GBi) or weighted bi-prediction (WP).

16. The method of claim 15, wherein determining the s value comprises:

determining s = 8 for GBi;

determining $s = 2^{luma_log2_weight_denom+Max(2, 14-LumaBitDepth)}$ for luma WP; or

determining $s = 2^{luma_log2_weight_denom+delta_chroma_log2_weight_denom+Max(2, 14-ChromaBitDepth)}$ for chroma WP.

17. 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: determine that a block of video data has one of: 1) a width of 4 pixels and a height of more than 8 pixels or 2) a height of 4 pixels and a width of more than 8 pixels; in response to determining that the block of video data has the one of 1) the width of 4 pixels and the height of more than 8 pixels or 2) the height of 4 pixels and the width of more than 8 pixels, determine that decoder-side motion vector refinement (DMVR) is not available to be used for the block; and in response to determining that DMVR is not available to be used for the block, code the block without performing DMVR for the block.

18. The device of claim 17, wherein the block comprises a first block, and wherein the one or more processors are further configured to: determine that a second block of video data has a size of at least 8×N or N×8, wherein N is an integer value greater than 8; in response to determining that the second block of video data has the size of at least 8×N or N×8, determine whether to code the second block using DMVR; in response to determining to code the second block using DMVR, code the second block using DMVR.

19. The device of claim 18, wherein to code the second block, the one or more processors are configured to: retrieve padded filter input samples of a reference picture for the second block when the second block is to be predicted using bi-directional optical flow (BIO); calculate one or more gradient values for one or more samples of the second block using the padded filter input samples; calculate one or more refined motion vectors for the one or more samples of the second block using the gradient values; and generate a prediction block for the second block using the one or more refined motion vectors.

20. The device of claim 17, wherein to code the block, the one or more processors are configured to: decode a motion vector for the block; calculate a prediction block for the block using the motion vector; decode a residual block for the block; and combine the prediction block with the residual block to decode the block.

21. The device of claim 17, wherein to code the block, the one or more processors are configured to: generate a motion vector for the block; calculate a prediction block for the block using the motion vector; generate a residual block for the block representing differences between the block and the prediction block; and encode the residual block and the motion vector to encode the block.

22. The device of claim 17, wherein to code the block, the one or more processors are configured to: retrieve a refined motion vector of an above-right neighboring block to the block as a motion vector predictor for a motion vector of the block; generate the motion vector for the current block using the motion vector predictor; and generate a prediction block for the block using the motion vector.

23. The device of claim 17, wherein to code the block, the one or more processors are configured to: retrieve a refined motion vector of a virtual pipeline data unit (VPDU) neighboring the block as a motion vector predictor for a motion vector of the block, wherein the VPDU comprises one of four VPDUs of a coding tree unit (CTU), wherein the four VPDUs are evenly split quarter blocks of the CTU; generate the motion vector for the current block using the motion vector predictor; and generate a prediction block for the block using the motion vector.

24. The device of claim 17, wherein to code the block, the one or more processors are configured to: generate a history-based motion vector prediction (HMVP) list including one or more motion vector predictor candidates for a motion vector of the block; in response to decoding a previous block using a refined motion vector, wherein an affine flag for the previous block indicates that affine motion is disabled for the previous block, prevent insertion of the refined motion vector into the HMVP list; code the motion vector for the block using one of the motion vector predictor candidates of the HMVP list; and generate a prediction block for the block using the motion vector.

25. The device of claim 17, wherein to code the block, the one or more processors are configured to: generate a history-based motion vector prediction (HMVP) list including one or more motion vector predictor candidates for a motion vector of the block; in response to decoding a previous block using a refined control point motion vector, wherein an affine flag for the previous block indicates that affine motion is enabled for the previous block, prevent insertion of the refined control point motion vector into the HMVP list; code the motion vector for the block using one of the motion vector predictor candidates of the HMVP list; and generate a prediction block for the block using the motion vector.

26. The device of claim 17, wherein to code the block, the one or more processors are configured to: determine an affine merge candidate for a motion vector of the block from a causal neighboring sub-block to the block, wherein to determine the affine merge candidate, the one or more processors are configured to: when the causal neighboring sub-block is in an above-neighboring coding tree unit (CTU) row to a CTU row including the block, select a refined motion vector of the causal neighboring sub-block as the affine merge candidate; or when the causal neighboring sub-block is in the CTU row including the block, select an unrefined motion vector used to generate the refined motion vector as the affine merge candidate; code the motion vector using the affine merge candidate; and generate a prediction block for the block using the motion vector.

27. The device of claim 17, wherein to code the block, the one or more processors are configured to: determine a positive integer scalar (s) value to be applied to weight values used to calculate a cost value using a cost function; calculate the cost value using the cost function and the s value for a sample of a prediction block of the block; and generate the prediction block using the cost value according to at least one of generalized bi-prediction (GBi) or weighted bi-prediction (WP).

28. The device of claim 17, further comprising a display configured to display the video data.

29. The device of claim 17, wherein the device comprises one or more of a camera, a computer, a mobile device, a broadcast receiver device, or a set-top box.

30. A non-transitory computer-readable storage medium having stored thereon instructions that, when executed, cause a processor to: determine that a block of video data has one of: 1) a width of 4 pixels and a height of more than 8 pixels or 2) a height of 4 pixels and a width of more than 8 pixels; in response to determining that the block of video data has the one of 1) the width of 4 pixels and the height of more than 8 pixels or 2) the height of 4 pixels and the width of more than 8 pixels, determine that decoder-side motion vector refinement (DMVR) is not available to be used for the block; and in response to determining that DMVR is not available to be used for the block, code the block without performing DMVR for the block.

31. The non-transitory computer-readable storage medium of claim 30, wherein the block comprises a first block, further comprising instructions that cause the processor to: determine that a second block of video data has a size of at least 8×N or N×8, wherein N is an integer value greater than 8; in response to determining that the second block of video data has the size of at least 8×N or N×8, determine whether to code the second block using DMVR; in response to determining to code the second block using DMVR, code the second block using DMVR.

32. The non-transitory computer-readable storage medium of claim 30, wherein the instructions that cause the processor to code the block comprise instructions that cause the processor to: decode a motion vector for the block; calculate a prediction block for the block using the motion vector; decode a residual block for the block; and combine the prediction block with the residual block to decode the block.

33. The non-transitory computer-readable storage medium of claim 30, wherein the instructions that cause the processor to code the block comprise instructions that cause the processor to: generate a motion vector for the block; calculate a prediction block for the block using the motion vector; generate a residual block for the block representing differences between the block and the prediction block; and encode the residual block and the motion vector to encode the block.

34. A device for coding video data, the device comprising: means for determining that a block of video data has one of: 1) a width of 4 pixels and a height of more than 8 pixels or 2) a height of 4 pixels and a width of more than 8 pixels; means for determining that decoder-side motion vector refinement (DMVR) is not available to be used for the block in response to determining that the block of video data has the one of 1) the width of 4 pixels and the height of more than 8 pixels or 2) the height of 4 pixels and the width of more than 8 pixels; and means for coding the block without performing DMVR for the block in response to determining that DMVR is not available to be used for the block.

35. The device of claim 34, wherein the block comprises a first block, the device further comprising: means for determining that a second block of video data has a size of at least 8×N or N×8, wherein N is an integer value greater than 8; means for determining whether to code the second block using DMVR in response to determining that the second block of

video data has the size of at least 8×N or N×8; means for coding the second block using DMVR in response to a determination to code the second block using DMVR.

36. The device of claim 34, wherein the means for coding the block comprises: means for decoding a motion vector for the block; means for calculating a prediction block for the block using the motion vector; means for decoding a residual block for the block; and means for combining the prediction block with the residual block to decode the block.

37. The device of claim 34, wherein the means for coding the block comprises: means for generating a motion vector for the block; means for calculating a prediction block for the block using the motion vector; means for generating a residual block for the block representing differences between the block and the prediction block; and means for encoding the residual block and the motion vector to encode the block.