

CLAIMS

1. A method of video processing, comprising:
 - deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which weights are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block; and
 - performing the conversion using the prediction block;
 - wherein the deriving of the prediction block comprises:
 - applying the weights to two intermediate prediction samples that respectively belong to the two initial prediction blocks to derive a final prediction sample; and
 - performing a right shift operation for the final prediction sample to convert a bit depth of the final prediction sample, wherein the right shift operation is $\text{pbSample} \gg (\text{shift1} + 3)$, wherein pbSample represents the final prediction sample, shift1 is set equal to $\text{Max}(2, 14 - \text{bitDepth})$; and
 - determining the prediction block based on the final prediction sample.
2. The method of claim 1, wherein, before the performing the right shift operation, an offset is added to the final prediction sample.
3. The method of claim 1, wherein, the offset is $1 \ll (\text{shift1} + 2)$.
4. The method of claim 3, wherein the final prediction sample, $\text{pbSamples}[x][y]$, is derived such that $\text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (w_0 * \text{predSamplesL0}[x][y] + w_1 * \text{predSamplesL1}[x][y] + \text{offset3}) \gg (\text{shift1} + 3))$, wherein w_0 and w_1 are the weights, and $\text{predSamplesL0}[x][y]$ and $\text{predSamplesL1}[x][y]$ are the two intermediate prediction samples.
5. The method of claim 1, wherein at least one of the weights belongs to a weight table, wherein entries of the weight table are non-monotonically increasing.

6. A method of video processing, comprising:
 - deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which weights are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block; and
 - performing the conversion using the prediction block;
 - wherein the deriving of the prediction block comprises:
 - applying the weights to two intermediate prediction sample that respectively belong to the two initial prediction blocks to derive a final prediction sample; and
 - adding an offset to the final prediction sample, wherein the offset is $1 \ll (\text{shift1} + 2)$, wherein shift1 is set equal to $\text{Max}(2, 14 - \text{bitDepth})$; and
 - determining the prediction block based on the final prediction sample.
7. The method of claim 6, wherein the determining the prediction block based on the final prediction sample comprises: performing a right shift operation for the final prediction sample to convert a bit depth of the final prediction sample.
8. The method of claim 1 or 7, wherein the right shift operation is $\text{pbSample} \gg (\text{shift1} + 3)$, wherein pbSample represents the final prediction sample.
9. The method of claim 8, wherein the final prediction sample, $\text{pbSamples}[x][y]$, is derived such that $\text{pbSamples}[x][y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (w0 * \text{predSamplesL0}[x][y] + w1 * \text{predSamplesL1}[x][y] + \text{offset3}) \gg (\text{shift1} + 3))$, wherein $w0$ and $w1$ are the weights, and $\text{predSamplesL0}[x][y]$ and $\text{predSamplesL1}[x][y]$ are the two intermediate prediction samples.
10. The method of claim 6, wherein at least one of the weights belongs to a weight table, wherein entries of the weight table are non-monotonically increasing.
11. A method of video processing, comprising:
 - deriving, for a conversion between a current block of a video and a coded representation of

the video, a motion vector difference for a merge mode motion vector predictor for the current block according to a rule; and

performing the conversion based on the deriving,

wherein the rule specifies whether to clip the motion vector difference to a same range as a range used for clipping motion vectors.

12. The method of claim 11, wherein the rule specifies that the motion vector difference is clipped to the same range as the range used for clipping motion vectors.

13. The method of claim 11, wherein the range used for clipping motion vectors is 18 bits.

14. The method of claim 13, wherein the range used for clipping motion vectors is $[-2^{17}, 2^{17} - 1]$.

15. The method of claim 11, wherein the rule specifies that the motion vector difference is clipped to $[-2^{17}+1, 2^{17} - 1]$.

16. The method of claim 11, wherein the rule specifies not to clip the motion vector difference.

17. The method of claim 16, wherein the rule specifies to clip a result obtained by adding the motion vector difference to a motion vector prediction candidate to 18 bits.

18. A method of video processing, comprising:

deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which weights are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block, wherein at least one of the weights belongs to a weight table, wherein the weight table is organized according to a rule; and

performing the conversion using the prediction block,

wherein the rule specifies that entries of the weight table are non-monotonically increasing.

19. The method of claim 18, wherein the weight table includes an (i+1)th entry whose value is smaller than that of i-th entry.

20. A method of video processing, comprising:

deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which weights are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block, wherein at least one of the weights belongs to a weight table, and the weight table is selected from multiple weight tables; and

performing the conversion using the prediction block.

21. The method of claim 20, wherein the multiple weight tables include {4, 5, 3, 10, 2}, {4, 3, 5, 10, 2}, {4, 5, 3, 10, 1}, {4, 3, 5, 10, 1}, or {4, 5, 3, 10, -1}.

22. A method of video processing, comprising:

deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which weights are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block; and

performing the conversion using the prediction block;

wherein the deriving of the prediction block comprises:

converting an intermediate prediction sample to a first bit depth;

applying the weights to the intermediate prediction sample to derive a final prediction sample; and

converting the final prediction sample to a second bit depth.

23. The method of claim 22, wherein the second bit depth is a bit depth of a color component of the current block.

24. The method of claim 22, wherein the first bit depth is not equal to a bit depth of the

intermediate prediction sample.

25. The method of claim 22, wherein the deriving of the prediction block comprises a right shift operation that converts a sample from a higher bit depth to a lower bit depth or left shift operation that converts the sample from the lower bit depth to the higher bit depth.

26. The method of claim 25, wherein an offset is added before the right shift operation or the left shift operation.

27. The method of claim 22, wherein the final prediction sample, $pbSamples[x][y]$, is derived such that $pbSamples[x][y] = Clip3(0, (1 \ll bitDepth) - 1, (w0 * ((predSamplesL0[x][y] + offset1) \gg shift1) + w1 * ((predSamplesL1[x][y] + offset1) \gg shift1) + 4) \gg 3)$, where $w0$ and $w1$ are the weights, $shift1$ is set equal to $Max(2, 14 - bitDepth)$ and $offset1$ is set equal to $1 \ll (shift1 - 1)$.

28. A method of video processing, comprising:

deriving, for a conversion between a current block of a video and a coded representation of the video, a prediction block for the current block that is determined according to a bi-directional coding unit level weighting (BCW) tool in which a first weight and a second weight are used to determine a weighted sum of two initial prediction blocks used to determine the prediction block, wherein at least one of the first weight and the second weight is included in a weight table, wherein the weight table is organized according to a rule; and

performing the conversion using the prediction block,

wherein the rule specifies that the weight table is asymmetric.

29. The method of claim 28, wherein the weight table includes an entry whose value corresponds to the first weight, W , and does not include any entry whose value corresponds to the second weight, $(1-W)$, whereby W is in the range of $[a, b]$ and a sum of a and b , $(a+b)$, is equal to 1.

30. The method of claim 28, wherein the weight table includes an entry whose value

corresponds to the first weight, W , and does not include any entry whose value corresponds to the second weight, $(2^N - W)$, in case that a prediction sample is generated by $(W_0 * P_0 + W_1 * P_1) \gg N$, whereby W_0 and W_1 are the first weight and the second weight applied to the initial two prediction blocks P_0 and P_1 , respectively; and $(W_0 + W_1)$ is equal to $(1 \ll N)$.

31. The method of any of claims 1 to 30, wherein the performing of the conversion includes generating the coded representation from the current video block.
32. The method of any of claims 1 to 30, wherein the performing of the conversion includes generating the current video block from the coded representation.
33. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1 to 32.
34. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of claims 1 to 32.
35. A computer readable medium that stores a coded representation or a bitstream representation generated according to any of the above described methods.