

CLAIMS

What is claimed is:

1. A visual media processing method, comprising:
performing a conversion between a current video block of a current picture of a visual media data and a bitstream representation of the visual media data,
wherein, the conversion is based on a reference region from the current picture comprising reference samples used for deriving a prediction block of the current video block, and
wherein a virtual buffer of a defined size is used for tracking availability of the reference samples for deriving the prediction block.
2. The method of claim 1, wherein the virtual buffer is maintained using a virtual pipeline data unit (VPDU), and wherein a size of the virtual buffer is $m \times W_{VPDU} \times n \times H_{VPDU}$, where W_{VPDU} and H_{VPDU} denote a width and a height of the VPDU.
3. The method of claim 2, wherein $m=4$ and $n=2$.
4. The method of claim 2, wherein m and/or n are based at least in part on a resolution of a picture associated with the current video block or a size of a coding tree unit including the current video block.
5. The method of claim 2, wherein m and/or n are predefined quantities.
6. The method of claim 2, wherein m and/or n are signaled as fields in the bitstream representation.
7. The method of claim 1, wherein a sample in the current video block is mapped to $(x \% (m \times W_{VPDU}), y \% (n \times H_{VPDU}))$ in the virtual buffer, where the sample in the current video block is located at (x, y) relative to a upper-left corner of a picture; " $x \% y$ " is defined as $y = x - y \times \text{floor}(x/y)$, where $\text{floor}(a)$ is the largest integer no greater than a , and W_{VPDU} and H_{VPDU} denote a width and a height of the VPDU.
8. The method of claim 1, further comprising:
using an array for tracking availabilities of samples stored in the virtual buffer.

9. The method of claim 8, wherein the array includes a flag to indicate if one or more samples stored in the buffer are used for prediction in an intra block copy mode.
10. The method of claim 8, wherein the array corresponds to one or more VPDU's of size 3×2 .
11. The method of claim 8, wherein the array corresponds to one or more VPDU's of size 4×2 .
12. The method of claim 1, wherein a subset of samples stored in the virtual buffer are flagged as unavailable for prediction.
13. The method of claim 12, wherein, the subset of samples flagged as unavailable for prediction are based on a position of a most-recently processed VPDU.
14. The method of claim 13, wherein the samples are flagged as unavailable at a beginning of processing a VPDU.
15. The method of claim 14, wherein, if $y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}})$ is equal to 0, then the subset of samples located at positions (x, y) are flagged as unavailable, wherein x lies in a first predetermined range and y lies in a second predetermined range, where $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$ denotes an upper-left corner of a coding tree unit of a most-recently processed VPDU, and W_{VPDU} and H_{VPDU} denote a width and a height of the VPDU.
16. The method of claim 15, wherein the first range is expressed as $[x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}} \% mW_{\text{VPDU}}, ((x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}} \% mW_{\text{VPDU}}) - 1 + W_{\text{VPDU}}]$ and the second range is expressed as $[y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}}), (y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}})) - 1 + H_{\text{VPDU}}]$.
17. The method of claim 15, wherein the first range is expressed as $[x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}} \% mW_{\text{VPDU}}, ((x_{\text{PrevVPDU}} - 2W_{\text{VPDU}} + 2mW_{\text{VPDU}} \% mW_{\text{VPDU}}) - 1 + W_{\text{VPDU}}]$ and the second range is expressed as $[y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}}), (y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}})) - 1 + H_{\text{VPDU}}]$.
18. The method of claim 14, wherein, if $y_{\text{PrevVPDU}} \% (n * H_{\text{VPDU}})$ is not equal to 0, then the subset of samples located at positions (x, y) are flagged as unavailable, wherein x lies in a first predetermined range and y lies in a second predetermined range, where $(x_{\text{PrevVPDU}}, y_{\text{PrevVPDU}})$

denotes an upper-left corner of a coding tree unit of a most-recently processed VPDU, and W_{VPDU} and H_{VPDU} denote a width and a height of the VPDU.

19. The method of claim 18, wherein the first range is expressed as $[xPrevVPDU - W_{VPDU} + 2*m*W_{VPDU} \% (m*W_{VPDU}), ((xPrevVPDU - W_{VPDU} + 2*m*W_{VPDU}) \% (m*W_{VPDU})) - 1 + W_{VPDU}]$ and the second range is expressed as $[yPrevVPDU \% (n*H_{VPDU}), (yPrevVPDU \% (n*H_{VPDU})) - 1 + H_{VPDU}]$.

20. The method of claim 18, wherein the first range is expressed as $[xPrevVPDU - W_{VPDU} + 2*m*W_{VPDU} \% mW_{VPDU}, ((xPrevVPDU - W_{VPDU} + 2*m*W_{VPDU}) \% (m*W_{VPDU})) - 1 + W_{VPDU}]$ and the second range is expressed as $[yPrevVPDU \% (n*H_{VPDU}), (yPrevVPDU \% (n*H_{VPDU})) - 1 + H_{VPDU}]$.

21. The method of claim 12, wherein, when the coding tree includes VPDUs, the subset of samples flagged as unavailable for prediction are based on a position of a most-recently processed coding tree unit.

22. The method of claim 21, wherein the samples are flagged as unavailable at a beginning of processing a coding tree unit.

23. The method of any one or more of claims 1-22, further comprising:
determining validity of a block vector corresponding to the current video block based on an upper-left position of the current video block, a bottom-left position of the current video block, and a bottom-right position of the current video block, wherein the determining excludes use of an upper-right position of the current video block.

24. The method of any one or more of claims 1-23, wherein the conversion is performed in an intra block copy mode.

25. The method of any of claims 1-24, wherein the conversion includes generating the bitstream representation from the current video block.

26. The method of any of claims 1-24, wherein the conversion includes generating pixel values of the current video block from the bitstream representation.

27. A video encoder apparatus comprising a processor configured to implement a method recited in any one or more of claims 1-24.

28. A video decoder apparatus comprising a processor configured to implement a method recited in any one or more of claims 1-24.

29. A computer readable medium having code stored thereon, the code embodying processor-executable instructions for implementing a method recited in any one or more of claims 1-24.