

AN ENCODER, A DECODER AND CORRESPONDING METHODS

TECHNICAL FIELD

[0002] The present disclosure is generally related to video coding, and is specifically related to efficient signaling of coding tool parameters used to compress video data in video coding.

BACKGROUND

[0003] The amount of video data needed to depict even a relatively short video can be substantial, which may result in difficulties when the data is to be streamed or otherwise communicated across a communications network with limited bandwidth capacity. Thus, video data is generally compressed before being communicated across modern day telecommunications networks. The size of a video could also be an issue when the video is stored on a storage device because memory resources may be limited. Video compression devices often use software and/or hardware at the source to code the video data prior to transmission or storage, thereby decreasing the quantity of data needed to represent digital video images. The compressed data is then received at the destination by a video decompression device that decodes the video data. With limited network resources and ever increasing demands of higher video quality, improved compression and decompression techniques that improve compression ratio with little to no sacrifice in image quality are desirable.

CLAIMS

What is claimed is:

1. A method implemented in a decoder, the method comprising:
receiving, by a receiver of the decoder, a bitstream comprising a plurality of adaptation parameter sets (APSs) including a plurality of APS types associated with a coded slice, wherein each APS includes an APS identifier (ID), and wherein each of the APS types uses a separate value space for the APS ID; and
decoding, by the processor, the coded slice using parameters from the plurality of APSs obtained based on the APS IDs.
2. The method of claim 1, wherein the separate value spaces are overlapping.
3. The method of any of claims 1-2, wherein the plurality of APS types include an adaptive loop filter (ALF) type containing ALF parameters, a scaling list type containing scaling list parameters, and a luma mapping with chroma scaling (LMCS) type containing LMCS parameters.
4. The method of any of claims 1-3, wherein each APS includes an APS parameter type (`aps_params_type`) code set to a predefined value indicating a type of parameters included in the each APS.
5. The method of any of claims 1-4, wherein a current APS includes a current APS ID selected from a predefined range across a current value space, wherein the current APS ID is related to a previous APS ID associated with a previous APS of a same type as the current APS, and wherein the current APS ID is not related to another previous APS ID associated with another previous APS of a different type than the current APS.
6. The method of any of claims 1-5, wherein each of the separate value spaces extend over a predefined range, and wherein the predefined ranges are determined based on the APS types.
7. The method of any of claims 1-6, wherein each APS is identified by a combination of a current APS type and a current APS ID.
8. A method implemented in an encoder, the method comprising:

determining, by the processor, a plurality of types of parameters employed for encoding the coded slice;
encoding, by a processor of the encoder, a slice into a bitstream as a coded slice;
encoding, by the processor, the plurality of types of parameters into a plurality of adaptation parameter sets (APSs) of the bitstream;
assigning, by the processor, an APS identifier(ID) to each of the APSs, wherein each of the APS types uses a separate value space for the APS ID;
encoding, by the processor, each APS ID into the plurality of APSs.
9. The method of claim 8, wherein the separate value spaces are overlapping.

10. The method of any of claims 8-9, wherein the plurality of APS types include an adaptive loop filter (ALF) type containing ALF parameters, a scaling list type containing scaling list parameters, and a luma mapping with chroma scaling (LMCS) type containing LMCS parameters.
11. The method of any of claims 8-10, wherein each APS includes an APS parameter type (aps_params_type) code set to a predefined value indicating a type of parameters included in the each APS.
12. The method of any of claims 8-11, wherein a current APS includes a current APS ID selected from a predefined range across a current value space, wherein the current APS ID is related to a previous APS ID associated with a previous APS of a same type as the current APS, and wherein the current APS ID is not related to another previous APS ID associated with another previous APS of a different type than the current APS.
13. The method of any of claims 8-12, wherein each of the separate value spaces extend over a predefined range, and wherein the predefined ranges are determined based on the APS types.
14. The method of any of claims 8-13, wherein each APS is identified by a combination of a current APS type and a current APS ID.
15. A video coding device comprising:
a processor, a receiver coupled to the processor, a memory coupled to the processor, and a transmitter coupled to the processor, wherein the processor, receiver, memory, and transmitter are configured to perform the method of any of claims 1-14.
16. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of claims 1-14.
17. A decoder comprising:
a receiving unit, configured to receive a bitstream comprising a plurality of adaptation parameter sets (APSs) including a plurality of APS types associated with a coded slice, wherein each APS includes an APS identifier (ID), and wherein each of the APS types uses a separate value space for the APS ID; and
a decoding unit, configured to decode the coded slice using parameters from the plurality of APSs obtained based on the APS IDs.
18. The decoder of claim 17, wherein the decoder is further configured to perform the method of any of claims 1-7.
19. An encoder comprising:
a determining unit, configured to determine a plurality of types of parameters employed for encoding a slice;
an encoding unit, configured to:
 encode the slice into a bitstream as a coded slice;

encode the plurality of types of parameters into a plurality of adaptation parameter sets (APSs) of the bitstream; and

encode each APS identifier(ID) into the plurality of APSs; and

an assigning unit, configured to assign an APS ID to each of the APSs, wherein each of the APS types uses a separate value space for the APS ID.

20. The encoder of claim 19, wherein the encoder is further configured to perform the method of any of claims 8-14.

21. A non-transitory storage medium which includes an encoded bitstream, the bitstream comprising a plurality of adaptation parameter sets (APSs) including a plurality of APS types associated with a coded slice, wherein each APS includes an APS identifier (ID), and wherein each of the APS types uses a separate value space for the APS ID.

22. An decoder comprising processing circuitry for carrying out the method according to any one of claims 1 to 7.

23. A encoder comprising processing circuitry for carrying out the method according to any of claims 8 to 14.

24. A computer program product comprising program code for performing the method according to any of claims 1 to 14 when executed on a computer or a processor.

25. A non-transitory computer-readable medium carrying a program code which, when executed by a computer device, causes the computer device to perform the method of any one of claims 1 to 14.