

CLAIMS

What is claimed is:

1. A method implemented by a decoder, the method comprising:
receiving, by a receiver of the decoder, a bitstream comprising an output layer set (OLS) and a video parameter set (VPS), wherein the OLS includes one or more layers of coded pictures and the VPS includes an OLS mode identification code (ols_mode_idc) specifying that, for each OLS, all layers in the each OLS are output layers;
determining, by a processor of the decoder, the output layers based on the ols_mode_idc in the VPS; and
decoding, by a processor of the decoder, a coded picture from the output layers to produce a decoded picture.
2. The method of claim 1, wherein the ols_mode_idc specifies that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS.
3. The method any of claims 1-2, wherein the ols_mode_idc specifies that an i-th OLS includes layers with layer indices from zero to i, inclusive.
4. The method of any of claims 1-3, wherein the ols_mode_idc is equal to one.
5. The method of any of claims 1-4, wherein the VPS includes a VPS maximum layers minus one (vps_max_layers_minus1) that specifies the number of layers specified by the VPS, which is a maximum allowed number of layers in each coded video sequence (CVS) referring to the VPS.
6. The method of any of claims 1-5, wherein the total number of OLSs (TotalNumOlss) is equal to vps_max_layers_minus1 plus one when the ols_mode_idc is equal to zero or when the ols_mode_idc is equal to one.
7. The method of any of claims 1-6, wherein a number of layers in an i-th OLS (NumLayersInOlss[i]) and a network abstraction layer (NAL) unit header layer identifier (nuh_layer_id) value of a j-th layer in the i-th OLS (LayerIdInOLS[i][j]) are derived as follows:

```

NumLayersInOls[ 0 ] = 1
LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
for( i = 1, i < TotalNumOlss; i++ ) {
    if( each_layer_is_an_ols_flag ) {
        NumLayersInOls[ i ] = 1
        LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
    } else if( ols_mode_idc == 0 || ols_mode_idc == 1 ) {
        NumLayersInOls[ i ] = i + 1
        for( j = 0; j < NumLayersInOls[ i ]; j++ )
            LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
    }
}

```

where `vps_layer_id[i]` is an *i*-th VPS layer identifier, `TotalNumOlss` is a total number of OLSs specified by the VPS, and an `each_layer_is_an_ols_flag` is an each layer is an OLS flag that specifies whether at least one OLS contains more than one layer.

8. A method implemented by an encoder, the method comprising:
 - encoding, by a processor of the encoder, a bitstream comprising one or more output layer sets (OLSs) including one or more layers of coded pictures;
 - encoding into the bitstream, by the processor, a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (`ols_mode_idc`) specifying that for each OLS, all layers in the each OLS are output layers; and
 - storing, by a memory coupled to the processor, the bitstream for communication toward a decoder.
9. The method of claim 8, wherein the `ols_mode_idc` specifies that a total number of OLSs specified by the VPS is equal to a number of layers specified by the VPS.
10. The method of any of claims 8-9, wherein the `ols_mode_idc` specifies that an *i*-th OLS includes layers with layer indices from zero to *i*, inclusive
11. The method of any of claims 8-10, wherein the `ols_mode_idc` is equal to one.

12. The method of any of claims 8-11, wherein the VPS includes a VPS maximum layers minus one (`vps_max_layers_minus1`) that specifies the number of layers specified by the VPS, which is a maximum allowed number of layers in each coded video sequence (CVS) referring to the VPS.

13. The method of any of claims 8-12, wherein the total number of OLSs (`TotalNumOlss`) is equal to `vps_max_layers_minus1` plus one when the `ols_mode_idc` is equal to zero or when the `ols_mode_idc` is equal to one.

14. The method of any of claims 8-13, wherein a number of layers in an *i*-th OLS (`NumLayersInOls[i]`) and a network abstraction layer (NAL) unit header layer identifier (`nuh_layer_id`) value of a *j*-th layer in the *i*-th OLS (`LayerIdInOLS[i][j]`) are derived as follows:

```

NumLayersInOls[ 0 ] = 1
LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
for( i = 1, i < TotalNumOlss; i++ ) {
    if( each_layer_is_an_ols_flag ) {
        NumLayersInOls[ i ] = 1
        LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
    } else if( ols_mode_idc == 0 || ols_mode_idc == 1 ) {
        NumLayersInOls[ i ] = i + 1
        for( j = 0; j < NumLayersInOls[ i ]; j++ )
            LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
    }
}

```

where `vps_layer_id[i]` is an *i*-th VPS layer identifier, `TotalNumOlss` is a total number of OLSs specified by the VPS, and an `each_layer_is_an_ols_flag` is an each layer is an OLS flag that specifies whether at least one OLS contains more than one layer.

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 means for receiving a bitstream comprising an output layer set (OLS) and a video parameter set (VPS), wherein the OLS includes one or more layers of coded pictures and the VPS includes an OLS mode identification code (ols_mode_idc) specifying that for each OLS, all layers in the each OLS are output layers;

- a determining means for determining the output layers based on the ols_mode_idc in the VPS;

- a decoding means for decoding a coded picture from the output layers to produce a decoded picture; and

- a forwarding means for forwarding the decoded picture for display as part of a decoded video sequence.

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:

- an encoding means for:

- encoding a bitstream comprising one or more output layer sets (OLSs) including one or more layers of coded pictures; and

- encoding into the bitstream a video parameter set (VPS), wherein the VPS includes an output layer set (OLS) mode identification code (ols_mode_idc) specifying that for each OLS, all layers in the each OLS are output layers; and

- a storing means for storing the bitstream for communication toward a decoder.

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