

## CLAIMS

1. A method for inverse quantization of a current block of a picture, wherein the method is performed by a decoder, and the method comprising:
  - 5 receiving a bitstream;
  - obtaining a joint chrominance component residual, JCCR, control flag from the bitstream;
  - obtaining a chrominance mapping information from the bitstream based on the JCCR control flag;
  - 10 obtaining at least one chrominance quantization parameter, QP, offset from the bitstream based on the JCCR control flag;
  - obtaining a QP value for the current chrominance block based on the obtained chrominance mapping information and the at least one obtained chrominance QP offset;
  - performing inverse quantization on the current chrominance block by using the
  - 15 determined QP value.
2. The method according to claim 1, wherein the bitstream comprises an SPS level syntax, and the JCCR control flag is obtained from the SPS level syntax.
- 20 3. The method according to claim 1 or 2, wherein the JCCR control flag is the `sps_joint_cbr_enabled_flag`.
4. The method according to claim 3, wherein if the value of the `sps_joint_cbr_enabled_flag` is 1, the at least one obtained chrominance QP offset is specified by `slice_joint_cbr_qp_offset`.
- 25 5. The method according to any one of claims 1 to 4, wherein the chrominance mapping information comprises `delta_qp_in_val_minus1[i][j]` and `delta_qp_out_val[i][j]`, and the chrominance mapping information is obtained from an SPS level syntax comprised in the bitstream.
- 30 6. The method according to any one of claims 2 to 5, wherein the SPS level syntax comprises the following structure:

|                                         |                   |
|-----------------------------------------|-------------------|
| <code>seq_parameter_set_rbsp() {</code> | <b>Descriptor</b> |
| <code>...</code>                        |                   |

|                                                                                            |              |
|--------------------------------------------------------------------------------------------|--------------|
|                                                                                            |              |
| <b>same_qp_table_for_chroma</b>                                                            | <b>u(1)</b>  |
| for( i = 0; i < same_qp_table_for_chroma ? 1 : sps_joint_cbr_enabled_flag ? 3 : 2; i++ ) { |              |
| <b>num_points_in_qp_table_minus1[ i ]</b>                                                  | <b>ue(v)</b> |
| for( j = 0; j <= num_points_in_qp_table_minus1[ i ]; j++ ) {                               |              |
| <b>delta_qp_in_val_minus1[ i ][ j ]</b>                                                    | <b>ue(v)</b> |
| ...                                                                                        | <b>ue(v)</b> |
| }                                                                                          |              |
| }                                                                                          |              |
| }                                                                                          |              |
| ...                                                                                        |              |
| }                                                                                          |              |

7. The method according to any one of claims 1 to 6, wherein the obtaining the at least one chrominance QP offset from the bitstream based on the JCCR control flag comprises:  
obtaining, based on the JCCR control flag, the at least one chrominance QP offset from a  
5 picture parameter set, PPS, level syntax of the bitstream.

8. The method according to claim 7, wherein the PPS level syntax comprises the following structure:

|                                                            |              |
|------------------------------------------------------------|--------------|
| pic_parameter_set_rbsp( ) {                                | Descriptor   |
| ...                                                        |              |
| <b>pps_cb_qp_offset</b>                                    | <b>se(v)</b> |
| <b>pps_cr_qp_offset</b>                                    | <b>se(v)</b> |
| <b>if( sps_joint_cbr_enabled_flag )</b>                    |              |
| <b>pps_joint_cbr_qp_offset</b>                             | <b>se(v)</b> |
| <b>cu_chroma_qp_offset_enabled_flag</b>                    | <b>u(1)</b>  |
| <b>if( pps_cu_chroma_qp_offset_list_enabled_flag ) {</b>   |              |
| <b>cu_chroma_qp_offset_subdiv</b>                          | <b>ue(v)</b> |
| <b>chroma_qp_offset_list_len_minus1</b>                    | <b>ue(v)</b> |
| for( i = 0; i <= chroma_qp_offset_list_len_minus1; i++ ) { |              |
| <b>cb_qp_offset_list[ i ]</b>                              | <b>se(v)</b> |
| <b>cr_qp_offset_list[ i ]</b>                              | <b>se(v)</b> |
| <b>if( sps_joint_cbr_enabled_flag )</b>                    |              |
| <b>joint_cbr_qp_offset_list[ i ]</b>                       | <b>se(v)</b> |
| }                                                          |              |
| }                                                          |              |
| ...                                                        |              |
| }                                                          |              |

9. A method for inverse quantization of a current block of a picture, wherein the method is performed by a decoder, and the method comprising:
- receiving a bitstream, wherein the bitstream comprises a Slice Header syntax and a PPS syntax;
  - 5 obtaining syntax elements from the PPS syntax, wherein the obtained syntax elements comprise at least one chrominance quantization parameter, QP, offset;
  - obtaining chrominance QP offset information from the Slice Header, wherein the QP offset information is obtained independently of any PPS syntax elements in the PPS syntax;
  - determining a QP value for the current chrominance block depending on the at least one
  - 10 chrominance QP offset obtained from the PPS syntax and the chrominance QP offset information obtained from the Slice Header syntax;
  - performing inverse quantization on the current chrominance block by using the determined QP value.
- 15 10. A method for encoding a current block of a picture, wherein the method is performed by an encoder, the method comprising:
- encoding a joint chrominance component residual, JCCR, control flag into a bitstream;
  - encoding a chrominance mapping information into the bitstream based on the JCCR control flag;
  - 20 encoding at least one chrominance quantization parameter, QP, offset into the bitstream based on the JCCR control flag;
  - providing the bitstream.
11. The method according to claim 10, wherein the bitstream comprises an SPS level syntax,
- 25 and the JCCR control flag is encoded into the SPS level syntax.
12. The method according to claim 10 or 11, wherein the JCCR control flag is the `sps_joint_cbr_enabled_flag`.
- 30 13. The method according to claim 12, wherein if the value of the `sps_joint_cbr_enabled_flag` is 1, the at least one encoded chrominance QP offset is specified by `slice_joint_cbr_qp_offset`.
14. The method according to any one of claims 10 to 13, wherein the chrominance mapping information comprises `delta_qp_in_val_minus1[i][j]` and `delta_qp_out_val[i][j]`, and the

chrominance mapping information is encoded into an SPS level syntax comprised in the bitstream.

15. The method according to any one of claims 10 to 14, wherein the SPS level syntax  
5 comprises the following structure:

|                                                                                            |                   |
|--------------------------------------------------------------------------------------------|-------------------|
| seq_parameter_set_rbsp( ) {                                                                | <b>Descriptor</b> |
| ...                                                                                        |                   |
|                                                                                            |                   |
| <b>same_qp_table_for_chroma</b>                                                            | <b>u(1)</b>       |
| for( i = 0; i < same_qp_table_for_chroma ? 1 : sps_joint_cbr_enabled_flag ? 3 : 2; i++ ) { |                   |
| <b>num_points_in_qp_table_minus1[ i ]</b>                                                  | <b>ue(v)</b>      |
| for( j = 0; j <= num_points_in_qp_table_minus1[ i ]; j++ ) {                               |                   |
| <b>delta_qp_in_val_minus1[ i ][ j ]</b>                                                    | <b>ue(v)</b>      |
| ...                                                                                        | <b>ue(v)</b>      |
| }                                                                                          |                   |
| }                                                                                          |                   |
| }                                                                                          |                   |
| ...                                                                                        |                   |
| }                                                                                          |                   |

16. The method according to any one of claims 10 to 15, wherein the encoding the at least one chrominance QP offset into the bitstream based on the JCCR control flag comprises:

- encoding, based on the JCCR control flag, the at least one chrominance QP offset into  
10 a picture parameter set, PPS, level syntax of the bitstream.

17. The method according to claim 16, wherein the PPS level syntax comprises the following structure:

|                                                            |                   |
|------------------------------------------------------------|-------------------|
| pic_parameter_set_rbsp( ) {                                | <b>Descriptor</b> |
| ...                                                        |                   |
| <b>pps_cb_qp_offset</b>                                    | <b>se(v)</b>      |
| <b>pps_cr_qp_offset</b>                                    | <b>se(v)</b>      |
| <b>if( sps_joint_cbr_enabled_flag )</b>                    |                   |
| <b>pps_joint_cbr_qp_offset</b>                             | <b>se(v)</b>      |
| <b>cu_chroma_qp_offset_enabled_flag</b>                    | <b>u(1)</b>       |
| <b>if( pps_cu_chroma_qp_offset_list_enabled_flag ) {</b>   |                   |
| <b>cu_chroma_qp_offset_subdiv</b>                          | <b>ue(v)</b>      |
| <b>chroma_qp_offset_list_len_minus1</b>                    | <b>ue(v)</b>      |
| for( i = 0; i <= chroma_qp_offset_list_len_minus1; i++ ) { |                   |
| <b>cb_qp_offset_list[ i ]</b>                              | <b>se(v)</b>      |
| <b>cr_qp_offset_list[ i ]</b>                              | <b>se(v)</b>      |
| <b>if( sps_joint_cbr_enabled_flag )</b>                    |                   |

| <code>joint_cbr_qp_offset_list[ i ]</code> | <code>se(v)</code> |
|--------------------------------------------|--------------------|
| <code>}</code>                             |                    |
| <code>}</code>                             |                    |
| <code>...</code>                           |                    |
| <code>}</code>                             |                    |

18. A method for encoding a current block of a picture, wherein the method is performed by an encoder, the method comprising:

- encoding syntax elements of a PPS syntax into a bitstream, wherein the syntax elements
- 5 comprise at least one chrominance quantization parameter, QP, offset;
- encoding chrominance QP offset information of a Slice Header syntax into the bitstream, wherein the QP offset information is encoded independently of any PPS syntax elements in the PPS syntax;
- providing the bitstream.

10

19. A decoder (30) comprising processing circuitry for carrying out the method according to any one of claims 1 to 9.

20. An encoder (20) comprising processing circuitry for carrying out the method according to

15 any one of claims 10 to 18.

21. A computer program product comprising program code for performing the method according to any one of the preceding claims 1 – 18 when executed on a computer or a processor.

20

22. A decoder, comprising:

- one or more processors; and
- a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when
- 25 executed by the processors, configures the decoder to carry out the method according to any one of the preceding claims 1 - 9.

23. An encoder, comprising:

- one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the encoder to carry out the method according to any one of the preceding claims 10 – 18.

5

24. 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 the preceding claims 1 - 18.

10

25. A decoder (30), comprising  
a receiving unit configured to receive a bitstream;  
a first obtaining unit configured to obtain a joint chrominance component residual, JCCR, control flag from the bitstream;  
a second obtaining unit configured to obtain a chrominance mapping information from  
15 the bitstream based on the JCCR control flag;  
a third obtaining unit configured to obtain at least one chrominance quantization parameter, QP, offset from the bitstream based on the JCCR control flag;  
a fourth obtaining unit configured to obtain a QP value for the current chrominance block based on the obtained chrominance mapping information and the at least one obtained  
20 chrominance QP offset;  
an inverse quantizing unit configured to perform inverse quantization on the current chrominance block by using the determined QP value.

25

26. The decoder according to claim 25, wherein the bitstream comprises an SPS level syntax, and the JCCR control flag is obtained from the SPS level syntax.

27. The decoder according to claim 25 or 26, wherein the JCCR control flag is the `sps_joint_cbr_enabled_flag`.

30

28. The decoder according to claim 27, wherein if the value of the `sps_joint_cbr_enabled_flag` is 1, the at least one obtained chrominance QP offset is specified by `slice_joint_cbr_qp_offset`.

29. The decoder according to any one of claims 25 to 28, wherein the chrominance mapping information comprises  $\text{delta\_qp\_in\_val\_minus1}[i][j]$  and  $\text{delta\_qp\_out\_val}[i][j]$ , and the chrominance mapping information is obtained from an SPS level syntax comprised in the bitstream.

5

30. The decoder according to any one of claims 26 to 29, wherein the SPS level syntax comprises the following structure:

|                                                                                                            |                   |
|------------------------------------------------------------------------------------------------------------|-------------------|
| <code>seq_parameter_set_rbsp() {</code>                                                                    | <b>Descriptor</b> |
| <code>...</code>                                                                                           |                   |
| <code>same_qp_table_for_chroma</code>                                                                      | <b>u(1)</b>       |
| <code>for( i = 0; i &lt; same_qp_table_for_chroma ? 1 : sps_joint_cbr_enabled_flag ? 3 : 2; i++ ) {</code> |                   |
| <code>num_points_in_qp_table_minus1[ i ]</code>                                                            | <b>ue(v)</b>      |
| <code>for( j = 0; j &lt;= num_points_in_qp_table_minus1[ i ]; j++ ) {</code>                               |                   |
| <code>delta_qp_in_val_minus1[ i ][ j ]</code>                                                              | <b>ue(v)</b>      |
| <code>...</code>                                                                                           | <b>ue(v)</b>      |
| <code>}</code>                                                                                             |                   |
| <code>}</code>                                                                                             |                   |
| <code>}</code>                                                                                             |                   |
| <code>...</code>                                                                                           |                   |
| <code>}</code>                                                                                             |                   |

31. The decoder according to any one of claims 25 to 30, wherein the obtaining the at least one chrominance QP offset from the bitstream based on the JCCR control flag comprises: obtaining, based on the JCCR control flag, the at least one chrominance QP offset from a picture parameter set, PPS, level syntax of the bitstream.

10

32. The decoder according to claim 31, wherein the PPS level syntax comprises the following structure:

15

|                                                                            |                   |
|----------------------------------------------------------------------------|-------------------|
| <code>pic_parameter_set_rbsp() {</code>                                    | <b>Descriptor</b> |
| <code>...</code>                                                           |                   |
| <code>pps_cb_qp_offset</code>                                              | <b>se(v)</b>      |
| <code>pps_cr_qp_offset</code>                                              | <b>se(v)</b>      |
| <code>if( sps_joint_cbr_enabled_flag )</code>                              |                   |
| <code>pps_joint_cbr_qp_offset</code>                                       | <b>se(v)</b>      |
| <code>cu_chroma_qp_offset_enabled_flag</code>                              | <b>u(1)</b>       |
| <code>if( pps_cu_chroma_qp_offset_list_enabled_flag ) {</code>             |                   |
| <code>cu_chroma_qp_offset_subdiv</code>                                    | <b>ue(v)</b>      |
| <code>chroma_qp_offset_list_len_minus1</code>                              | <b>ue(v)</b>      |
| <code>for( i = 0; i &lt;= chroma_qp_offset_list_len_minus1; i++ ) {</code> |                   |

|                                               |                    |
|-----------------------------------------------|--------------------|
| <code>cb_qp_offset_list[ i ]</code>           | <code>se(v)</code> |
| <code>cr_qp_offset_list[ i ]</code>           | <code>se(v)</code> |
| <code>if( sps_joint_cbr_enabled_flag )</code> |                    |
| <code>joint_cbr_qp_offset_list[ i ]</code>    | <code>se(v)</code> |
| <code>}</code>                                |                    |
| <code>}</code>                                |                    |
| <code>...</code>                              |                    |
| <code>}</code>                                |                    |

33. A decoder (30), comprising:

5 a receiving unit configured to receive a bitstream, wherein the bitstream comprises a Slice Header syntax and a PPS syntax;

a first obtaining unit configured to obtain syntax elements from the PPS syntax, wherein the obtained syntax elements comprises chrominance quantization parameter, QP, offsets;

10 a second obtaining unit configured to obtain chrominance QP offset information from the Slice Header, wherein the QP offset information is obtained independently of any PPS syntax elements in the PPS syntax;

a determining unit configured to determining a QP value for the current chrominance block depending on the chrominance QP offset obtained from the PPS syntax and the chrominance QP offset information obtained from the Slice Header syntax;

15 an inverse quantization unit configured to perform inverse quantization on the current chrominance block by using the determined QP value.

34. An encoder (20), comprising:

20 a first encoding unit configured to encode a joint chrominance component residual, JCCR, control flag into a bitstream;

a second encoding unit configured to encode a chrominance mapping information into the bitstream based on the JCCR control flag;

a third encoding unit configured to encode at least one chrominance quantization parameter, QP, offset into the bitstream based on the JCCR control flag;

25 a providing unit configured to provide the bitstream.

35. The encoder according to claim 34, wherein the bitstream comprises an SPS level syntax, and the JCCR control flag is encoded into the SPS level syntax.

36. The encoder according to claim 33 or 35, wherein the JCCR control flag is the `sps_joint_cbr_enabled_flag`.

- 5 37. The encoder according to claim 36, wherein if the value of the `sps_joint_cbr_enabled_flag` is 1, the at least one encoded chrominance QP offset is specified by `slice_joint_cbr_qp_offset`.

38. The encoder according to any one of claims 33 to 37, wherein the chrominance mapping  
10 information comprises `delta_qp_in_val_minus1[ i ][ j ]` and `delta_qp_out_val[ i ][ j ]`, and the chrominance mapping information is encoded into an SPS level syntax comprised in the bitstream.

39. The encoder according to any one of claims 33 to 38, wherein the SPS level syntax  
15 comprises the following structure:

|                                                                                                            |                   |
|------------------------------------------------------------------------------------------------------------|-------------------|
| <code>seq_parameter_set_rbsp( ) {</code>                                                                   | <b>Descriptor</b> |
| <code>...</code>                                                                                           |                   |
| <code>same_qp_table_for_chroma</code>                                                                      | <b>u(1)</b>       |
| <code>for( i = 0; i &lt; same_qp_table_for_chroma ? 1 : sps_joint_cbr_enabled_flag ? 3 : 2; i++ ) {</code> |                   |
| <code>num_points_in_qp_table_minus1[ i ]</code>                                                            | <b>ue(v)</b>      |
| <code>for( j = 0; j &lt;= num_points_in_qp_table_minus1[ i ]; j++ ) {</code>                               |                   |
| <code>delta_qp_in_val_minus1[ i ][ j ]</code>                                                              | <b>ue(v)</b>      |
| <code>...</code>                                                                                           | <b>ue(v)</b>      |
| <code>}</code>                                                                                             |                   |
| <code>}</code>                                                                                             |                   |
| <code>}</code>                                                                                             |                   |
| <code>...</code>                                                                                           |                   |
| <code>}</code>                                                                                             |                   |

40. The encoder according to any one of claims 33 to 39, wherein the encoding the at least one chrominance QP offset into the bitstream based on the JCCR control flag comprises:

- encoding, based on the JCCR control flag, the at least one chrominance QP offset into  
20 a picture parameter set, PPS, level syntax of the bitstream.

41. The encoder according to claim 40, wherein the PPS level syntax comprises the following structure:

|                                                            |            |
|------------------------------------------------------------|------------|
| pic_parameter_set_rbsp( ) {                                | Descriptor |
| ...                                                        |            |
| pps_cb_qp_offset                                           | se(v)      |
| pps_cr_qp_offset                                           | se(v)      |
| if( sps_joint_cbr_enabled_flag )                           |            |
| pps_joint_cbr_qp_offset                                    | se(v)      |
| cu_chroma_qp_offset_enabled_flag                           | u(1)       |
| if( pps_cu_chroma_qp_offset_list_enabled_flag ) {          |            |
| cu_chroma_qp_offset_subdiv                                 | ue(v)      |
| chroma_qp_offset_list_len_minus1                           | ue(v)      |
| for( i = 0; i <= chroma_qp_offset_list_len_minus1; i++ ) { |            |
| cb_qp_offset_list[ i ]                                     | se(v)      |
| cr_qp_offset_list[ i ]                                     | se(v)      |
| if( sps_joint_cbr_enabled_flag )                           |            |
| joint_cbr_qp_offset_list[ i ]                              | se(v)      |
| }                                                          |            |
| }                                                          |            |
| ...                                                        |            |
| }                                                          |            |

- 5 42. An encoder (20), comprising:
- a first encoding unit configured to encode syntax elements from Slice Header and PPS syntax into the bitstream, wherein the syntax elements comprise chrominance quantization parameter, QP, offsets;
- a second encoding unit configured to encode chrominance QP offset information from
- 10 the Slice Header into the bitstream, wherein the QP offset information is obtained independently of any PPS syntax elements in the PPS syntax;
- a providing unit configured to provide the bitstream.