

CLAIMS

1. A method for processing a reconstructed block, wherein the reconstructed block comprises a plurality of pixels, wherein the method comprises:
 - obtaining a quantization parameter (QP) of the reconstructed block;
 - obtaining a threshold (THR) based on the QP; and
 - obtaining a look up table based on QP, so as to generate a filtered reconstructed block based on the threshold and the look up table.
2. The method of claim 1, wherein the method further comprises:
 - scanning a current pixel of the reconstructed block and its neighboring pixels of the current pixel according to a predefined scan template;
 - obtaining spectrum components by performing transform for the current pixel and its neighboring pixels;
 - obtaining filtered spectrum components by using the look up table and the spectrum components.
3. The method of claim 2, a filtered spectrum component $F(i, \sigma)$ is derived by:

$$F(i, \sigma) = \begin{cases} R(i) & , \text{ } Abs(R(i)) \geq THR \\ LUT(R(i), \sigma), & R(i) > 0 \\ -LUT(-R(i), \sigma), & R(i) \leq 0 \end{cases}$$

wherein (i) is an index of a spectrum component, $R(i)$ is the spectrum component corresponding to (i) index, σ is a filtering parameter, $LUT(R(i), \sigma)$ is an array in the look up table, and THR is the threshold being derived from the QP.
4. The method of claim 3, wherein $LUT(R(i), \sigma) = \frac{R(i)^3}{R(i)^2 + m \cdot \sigma^2}$, m is normalization constant equal to number of spectrum components.
5. The method of claim 3 or 4, wherein the filtering parameter is derived based on the QP.
6. The method of claim 5, wherein the filtering parameter σ is derived by

$$\sigma = k * 2^{(n*(QP-s))}$$

, wherein QP is the codec quantization parameter, k, n and s are constants.

7. The method of claim 6, wherein $k = 2.64, n = 0.1296$ and $s = 11$.

8. The method of any one of claims 1-7, wherein the threshold is derived by:

Wherein C is a value close to 1, σ is a filtering parameter, and m is normalization constant equal to number of spectrum components.

$$\frac{THR^2}{THR^2 + m * \sigma^2} = C$$

9. The method of any one of claims 1-8, wherein the method further comprises:

choose a set of QPs to obtain the lookup table (LUT), wherein the set of QPs include a first QP corresponding to (i) index and a second QP corresponding to (i+1) index, and wherein the first QP and the second QP have an interval more than 1.

10. The method of claim 9, wherein the interval is constant and equal to 8, 10 or 16.

11. The method of any one of claims 1-10, wherein the method further comprises:

dropping N bits from table values of the LUT, N is an integer.

12. The method of claim 11, wherein access to element of LUT is as follows

$tbl[(z + HTDF_TBL_RND) \gg HTDF_TBL_SH]$

where $HTDF_TBL_SH$ is alias of N , and $HTDF_TBL_RND = 1 \ll (HTDF_TBL_SH - 1)$

13. The method of claim 11, wherein access to element of LUT is as follows

$LUT[(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift]$

for positive values of $fHad[i]$, and

$-LUT[(-(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift)]$

for negative values of $fHad[i]$

where $tblShift$ is alias of N , and $fHad[i]$ is spectrum components to be filtered.

14. The method of any one of claims 11-13, wherein N is 3.

15. The method of any one of claims 11-13, wherein for the first QP in a set of QPs, N is equal to 2, and wherein the set of QPs is used to obtain the lookup table (LUT);

wherein for the last QP or for the last two QPs in the set of QPs, N is equal to 4; and wherein for the rest QPs from the set of QPs, N is equal to 3.

16. The method of any one of claims 11-13, N is defined as follows;

$$tblShift = tblThrLog2[qpIdx] - 4$$

$$tblThrLog2[5] = \{ 6, 7, 7, 8, 8 \}$$

where tblShift is alias of N, and qpIdx is derived based on QP.

17. The method of claim 16, qpIdx is derived based on QP as follows;

$$\text{if}(\text{pred_mode_flag}[xCb][yCb] = 0 \ \&\& \ nCbW == nCbH \ \&\& \ \min(nCbW, nCbH) \geq 32)$$

$$qpIdx = \text{Clip3}(0, 4, (QpY - 28 + (1 \ll 2)) \gg 3)$$

else

$$qpIdx = \text{Clip3}(0, 4, (QpY - 20 + (1 \ll 2)) \gg 3)$$

where QpY is the current block QP, xCb, yCb defines the position of the current block on the picture (x, y), $\text{pred_mode_flag}[xCb][yCb]$ defines prediction mode of the current block, if equal to 0 the prediction is inter, otherwise intra, $nCbW, nCbH$ is the width and height of the current block correspondingly.

18. The method of any one of claims 11-13, wherein N is dependent on QP value.

19. The method of any one of claims 1-18, wherein LUT generation is based on an auxiliary function for at least some quantization parameters (QPs).

20. The method of claim 19, wherein the LUT generation $LUT(R_i, \sigma)$ is derived by:

$$LUT(R_i, \sigma) = \max \left(\frac{R_i^3}{R_i^2 + m * \sigma^2}, \text{AuxiliaryFunc}_\sigma(i) \right),$$

wherein $\text{AuxiliaryFunc}_\sigma(i)$ represents the auxiliary function, and the auxiliary function has a value equal to THR at i corresponding to an argument of last LUT + 1.

21. The method of claim 19 or 20, wherein the auxiliary function is a straight line equation comming over points (i,THR) and (a, 0), where $a > 0$ and a depends on filtering parameter σ or QP value.

22. The method of claim 21, wherein the LUT is defined as follows

```

setOfLUT[5][16]=
{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 3, 9, 19, 32, 47, 64, 81, 99, 117, 135, 154, 179, 205, 230, },
{ 0, 0, 0, 2, 6, 11, 18, 27, 38, 51, 64, 96, 128, 160, 192, 224, },

```

23. The method of claim 21, wherein the LUT is defined as follows

```

setOfLUT[5][16]=
{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, },
{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, },

```

24. The method of claim 21, wherein the LUT is defined as follows

```

setOfLUT[5][16]=
{ 0, 2, 10, 19, 28, 36, 45, 53, 61, 70, 78, 86, 94, 102, 110, 118, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, },
{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, },

```

25. The method of any one of claims 1-24, wherein the transform is a Hadamard transform.

26. The method of any one of claims 1-25, wherein the transform is a 1D transform.

27. A method for processing a predicted block, wherein the predicted block comprises a plurality of pixels, wherein the method comprises:

- obtaining a quantization parameter (QP) of the predicted block;
- obtaining a threshold (THR) based on the QP;
- obtaining a look up table based on QP, so as to generate a filtered predicted block based on the threshold and the look up table.

28. The method of claim 27, wherein the method further comprises:

scanning a current pixel of the predicted block and its neighboring pixels of the current pixel according to a predefined scan template;
 obtaining spectrum components by performing transform for the current pixel and its neighboring pixels;
 obtaining filtered spectrum components by using the look up table and the spectrum components.

29. The method of claim 28, a filtered spectrum component $F(i, \sigma)$ is derived by:

$$F(i, \sigma) = \begin{cases} R(i) & , \text{Abs}(R(i)) \geq THR \\ LUT(R(i), \sigma), & R(i) > 0 \\ -LUT(-R(i), \sigma), & R(i) \leq 0 \end{cases}$$

wherein (i) is an index of a spectrum component, $R(i)$ is the spectrum component corresponding to (i) index, σ is a filtering parameter, $LUT(R(i), \sigma)$ is an array in the look up table, and THR is the threshold being derived from the QP.

30. The method of claim 29, wherein $LUT(R(i), \sigma) = \frac{R(i)^3}{R(i)^2 + m \cdot \sigma^2}$, m is normalization constant equal to number of spectrum components.

31. The method of claim 29 or 30, wherein the filtering parameter is derived from the QP.

32. The method of claim 29, wherein the filtering parameter σ is derived by

$$\sigma = k * 2^{(n * (QP - s))}$$

, wherein QP is the codec quantization parameter, k, n and s are constants.

33. The method of any one of claims 27-32, wherein the threshold is derived by:
 Wherein C is a value close to 1, σ is a filtering parameter, and m is normalization

$$\frac{THR^2}{THR^2 + m \cdot \sigma^2} = C$$

constant equal to number of spectrum components.

34. The method of any one of claims 27-33, wherein the method further comprises:

choose a set of QPs to obtain the lookup table (LUT), wherein the set of QPs include a first QP corresponding to (i) index and a second QP corresponding to (i+1) index, and wherein the first QP and the second QP have an interval more than 1.

35. The method of any one of claims 27-34, wherein the method further comprises:
dropping N bits from table values of the LUT, N is an integer.

36. The method of claim 35, where access to element of LUT is as follows

$tbl[(z + HTDF_TBL_RND) \gg HTDF_TBL_SH]$
where HTDF_TBL_SH is alias of N, and $HTDF_TBL_RND = 1 \ll (HTDF_TBL_SH - 1)$

37. The method of claim 35, where access to element of LUT is as follows

$LUT[(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift]$
for positive values of fHad[i], and
 $-LUT[(-(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift)]$
for negative values of fHad[i]
where tblShift is alias of N, and fHad[i] is spectrum components to be filtered.

38. The method of any one of claims 35-37, N is defined as follows;

$tblShift = tblThrLog2[qIdx] - 4$
 $tblThrLog2[5] = \{6, 7, 7, 8, 8\}$
where tblShift is alias of N, and qIdx is derived based on QP.

39. The method of claim 38, qIdx is derived based on QP as follows;

if (pred_mode_flag[xCb][yCb] == 0 && nCbW == nCbH && min(nCbW, nCbH) >= 32)
 $qIdx = Clip3(0, 4, (QpY - 28 + (1 \ll 2)) \gg 3)$
else
 $qIdx = Clip3(0, 4, (QpY - 20 + (1 \ll 2)) \gg 3)$
where QpY is the current block QP, xCb, yCb defines the position of the current block on the picture (x, y), pred_mode_flag[xCb][yCb] defines prediction mode of the current block, if equal to 0 the prediction is inter, otherwise intra, nCbW, nCbH is the width and height of the current block correspondingly.

40. The method of any one of claims 35-39, wherein N is dependent on QP value.

41. The method of any one of claims 35-40, wherein LUT generation is based on an auxiliary function for at least some quantization parameters (QPs).

42. The method of claim 41, wherein the LUT generation $LUT(R_i, \sigma)$ is derived by:

$$LUT(R_i, \sigma) = \max\left(\frac{R_i^3}{R_i^2 + m * \sigma^2}, AuxiliaryFunc_{\sigma}(i)\right),$$

wherein $AuxiliaryFunc_{\sigma}(i)$ represents the auxiliary function, and the auxiliary function has a value equal to THR at i corresponding to an argument of last LUT + 1.

43. The method of claim 41 or 42, wherein the auxiliary function is a straight line equation comming over points (i, THR) and $(a, 0)$, where $a > 0$ and a depends on filtering parameter σ or QP value.

44. The method of claim 43, wherein the LUT is defined as follows

```
setOfLUT[ 5 ][ 16 ] =
{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 3, 9, 19, 32, 47, 64, 81, 99, 117, 135, 154, 179, 205, 230, },
{ 0, 0, 0, 2, 6, 11, 18, 27, 38, 51, 64, 96, 128, 160, 192, 224, },
```

45. The method of claim 43, wherein the LUT is defined as follows

```
setOfLUT[ 5 ][ 16 ] =
{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, },
{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, },
```

46. The method of claim 43, wherein the LUT is defined as follows

```
setOfLUT[ 5 ][ 16 ] =
{ 0, 2, 10, 19, 28, 36, 45, 53, 61, 70, 78, 86, 94, 102, 110, 118, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },
{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, },
{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, },
{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, }.
```

47. A decoder comprising processing circuitry for carrying out the method according to any one of claims 1-46.
48. An encoder comprising processing circuitry for carrying out the method according to any one of claims 1-46.
49. A decoder comprising a filter for carrying out the method according to any one of claims 1-46.
50. An encoder comprising a filter for carrying out the method according to any one of claims 1-46.
51. A decoder, comprising:
a memory storage comprising instructions; and
one or more processors in communication with the memory, wherein the one or more processors execute the instructions to carry out the method according to any one of claims 1-46.
52. An encoder, comprising:
a memory storage comprising instructions; and
one or more processors in communication with the memory, wherein the one or more processors execute the instructions to carry out the method according to any one of claims 1-46.
53. A computer program product comprising program code for performing the method of claims 1-46 when executed on a computer or a processor.