CAVLC in H.264

Encode the 16 integers

- VLC table selection for coef_token
- Coefficient Level (sign+magnitude) Coding
- Zero Coding

Number Coding

- Coef-token>>
- TotalCoeffs – number of non-zero coefficients
- T1 – Trailing Ones (0-3)

- Sign of each digit in T1

Coef-token

- Upper and left previously coded blocks
- N=(Nu+Nl)/2
- N=Nu
- N=Nl
- 3 VLC and 1 FLC tables

N	Table for coeff_token
0, 1	Num-VLC0
2, 3	Num-VLC1
4, 5, 6, 7	Num-VLC2
8 or above	FLC

Encode for coeff and T1s

trailing_ones	total_coeff	0 <= nC < 2	2 <= nC < 4	4 <= nC < 8	8 <= nC	nC == -1
0	0	1	11	1111	0000 11	01
0	1	0001 01	0010 11	0011 11	0000 00	0001 11
1	1	01	10	1110	0000 01	1
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00
1	2	0001 00	0011 1	0111 1	0001 01	0001 10
2	2	001	011	1101	0001 10	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011
2	3	0000 101	0010 01	0111 0	0010 10	0000 010
3	3	0001 1	0101	1100	0010 11	0001 01
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 0011
2	4	0000 0101	0001 01	0101 1	0011 10	0000 0010
3	4	0000 11	0100	1011	0011 11	0000 000
0	5	0000 0000 111	0000 0100	0001 011	0100 00	-
1	5	0000 0001 10	0000 110	0100 0	0100 01	-
2	5	0000 0010 1	0000 101	0100 1	0100 10	-
3	5	0000 100	0011 0	1010	0100 11	-
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	-
1	6	0000 0000 110	0000 0110	0011 10	0101 01	-
2	6	0000 0001 01	0000 0101	0011 01	0101 10	-
3	6	0000 0100	0010 00	1001	0101 11	-

Level coding

- The table to encode each label adapts depend on the magnitude of each successively encoded label
- Starting from level_VLC0
- If the magnitude is larger than a threshold move up to the next table

Total zero

- Total number of zero before the last non zero coefficients
- Number of zero preceding each non-zero coefficients (run of zero)— run_before
- The VLC table is dependent on both the run_before and zero_left

Level Coding

Each Coefficient Level is indicated by Quotent and Remainder

(levelprefix + levelsuffix)

levelCode = (levelPrefix << suffixLength) + levelSuffix

levelPrefix = levelCode / （1 << **suffixLength**）

levelSuffix = levelCode %（1 << **suffixLength**）

編碼步驟

步驟一：初始值設定。

suffixLength與**i**的初始值設爲**0**，**level [i]**爲Non Zero Coefficient值。

步驟二：將有號（signed）的**level [i]**轉換成無號（unsigned）的**levelCode**。

如果**level [i]**是正的，**levelCode =（level [i] << 1） - 2**;

如果**level [i]**是負的，**levelCode = -（level [i] << 1） - 1**

步驟三：計算**levelPrefix**。

levelPrefix = levelCode /（1 << **suffixLength**）。

步驟四：計算**levelSuffix**。

levelSuffix = levelCode %（1 << **suffixLength**）。

步驟五：**levelPrefix**查表A-3得到二進位代碼，然後輸入至位元流。

步驟六：**levelSuffix**的大小以**suffixLength**的位元長度表示。以**suffixLength**長度的位元數目將**levelSuffix**的大小輸入至位元流。

Level Scale Upgrading

步驟七：如果**suffixLength**等於0，**suffixLength**加一；如果 abs(**level(i)**)大於

3<<(**suffixLength**-1)並且**suffixLength**小於六，**suffixLength**再加一。

步驟八：如果**i**小於（total_coeff – T1s），**i**加一；回到步驟二。

For Example

4x4 block:

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0

- Scanned: 0,3,0,1,-1,-1,0,1,0……
- TotalCoeffs = 5
- TotalZeros = 3
- T1s = 3

Scanned: 0,3,0,1,-1,-1,0,1,0……

Element	Value	Code
coeff_token	TotalCoeffs=5, T1s=3	0000100
T1 sign (4)	+	0
T1 sign (3)	-	1
T1 sign (2)	-	1
Level (1)	+1 (use Level_VLC0)	1
Level (0)	+3 (use Level_VLC1)	0010
TotalZeros	3	111
run_before(4)	ZerosLeft=3; run_before=1	10
run_before(3)	ZerosLeft=2; run_before=0	1
run_before(2)	ZerosLeft=2; run_before=0	1
run_before(1)	ZerosLeft=2; run_before=1	01
run_before(0)	ZerosLeft=1; run_before=1	No code required; last coefficient.

- 1 →0 ,suffixlegth=0, 0 =(0,0) →1
- 3 →4 ,suffixlength=1, 4=(2,0) →0010

Prefix

Codeword table for levelPrefix

levelPrefix	bit string
0	1
1	01
2	001
3	0001
4	0000 1
5	0000 01
6	0000 001
7	0000 0001
8	0000 0000 1
9	0000 0000 01
10	0000 0000 001
11	0000 0000 0001
12	0000 0000 0000 1
13	0000 0000 0000 01
14	0000 0000 0000 001
15	0000 0000 0000 0001

Decoding

Code	Element	Value	Output array
0000100	coeff_token	TotalCoeffs=5, T1s=3	Empty
0	T1 sign	+	1
1	T1 sign	-	-1, 1
1	T1 sign	-	-1, -1, 1
1	Level	+1	1, -1, -1, 1
0010	Level	+3	3, 1, -1, -1, 1
111	TotalZeros	3	3, 1, -1, -1, 1
10	run_before	1	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
01	run_before	1	3, 0, 1, -1, -1, 0, 1

- The TotalZeros = 3, so another zero is inserted before the first coefficient

Another Example -2, 4, 3, -2, 0, 0, -1,0

Element	Value	Code
coeff_token	TotalCoeffs=5, T1s=1	0000000110
T1 sign (4)	-	1
Level (3)	Sent as –2 (**see note 1**) (use Level_VLC0)	0001
Level (2)	3 (use Level_VLC1)	0010
Level (1)	4 (use Level_VLC1)	00010
Level (0)	-2 (use Level_VLC2)	111
TotalZeros	2	0011
run_before(4)	ZerosLeft=2; run_before=2	00
run_before(3..0)	0	No code required

- Note 1:If there are less than 3 T1s, then the first non T1 level will not have a value of +/-1.Hence 3 is treated as 2 to save VLC bits.

- -2 →3 ,suffixlegth =0, 3=(3,0)→0001
- 3 →4, suffixlegth=1, 4=(2,0) →001 0
- 4 →6 ,suffixlength=1, 6=(3,0) →0001 0
- -2 →3 ,suffixlength=2, 3=(0,3) →1 11

Decoding

Code	Element	Value	Output array
0000000110	coeff_token	TotalCoeffs=5, T1s=1	Empty
1	T1 sign	-	-1
0001	Level	-2 decoded as -3	-3, -1
0010	Level	+3	+3, -3, -1
00010	Level	+4	+4, 3, -3, -1
111	Level	-2	-2, 4, 3, -3, -1
011	TotalZeros	2	-2, 4, 3, -3, -1
00	run_before	2	-2, 4, 3, -3, 0, 0, -1

Encode for level

Current VLC table	Threshold to increment table
VLC0	0
VLC1	3
VLC2	6
VLC3	12
VLC4	24
VLC5	48
VLC6	N/A (highest table)

Table for run_before

run_before	zerosLeft						
	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	-	00	01	01	011	001	101
3	-	-	00	001	010	011	100
4	-	-	-	000	001	010	011
5	-	-	-	-	000	101	010
6	-	-	-	-	-	100	001
7	-	-	-	-	-	-	0001
8		-	-	-	-	-	00001
9	-	-	-	-	-	-	000001
10	-	-	-	-	-	-	0000001
11	-	-	-	-	-	-	00000001
12	-	-	-	-	-	-	000000001
13	-	-	-	-	-	-	0000000001
14	-	-	-	-	-	-	00000000001

Level decoding process

- If total_coeff is larger than 10 and trailing_ones is smaller than 3, suffixLength is set to 1.
- Otherwise suffixLength is set to 0.
- The following procedure is then applied iteratively (total_coeff – trailing_ones) times to decode the remaining levels, if any:
- The syntax element coeff_level is decoded in two steps. In a first step a value of levelPrefix is decoded using the VLC specified in Table 9-6. In a second step an unsigned integer levelSuffix is read from the slice data. The size in bits of the unsigned integer is equal to suffixLength with the exception of the following two cases:
  - if levelPrefix is equal to 14 and suffixLength is 0, the size is 4 bits
  - if levelPrefix is equal to 15, the size is 12 bits.
- A variable levelCode is set to (levelPrefix << suffixLength) + levelSuffix.
- If levelPrefix is equal to 15 and suffixLength is equal to 0, levelCode is incremented by 15.
- If the index i is equal to trailing_ones and trailing_ones is smaller than 3, levelCode is incremented by 2.
- If levelCode is an even number the value (levelCode + 2) >> 1 is assigned to level[i]. Otherwise, the value (-levelCode – 1) >> 1 is assigned to level[i].
- If suffixLength is equal to zero, suffixLength is set to 1.
- If the absolute value of level[i] is larger than (3 << (suffixLength – 1)) and suffixLength is smaller than 6, suffixLength is incremented by 1.
- The index i is incremented by 1.