EBCOT coding passes explained on a detailed example

Xavier Delaunay
d.xav@free.fr

Contents

1 Introduction

JPEG2000 entropy coder is EBCOT (Embedded Block Coding with Optimal Truncation Points) contextual coder. It is a bit-plane coder. On each bit-plane, there are three coding passes: a pass of *Significance Propagation*, a pass of *Magnitude Refinement* and a *Cleanup* pass. Four coding primitives are used: the RL (Run-Length) primitive, the ZC (Zero Coding) primitive, the MR (Magnitude Refinement) and the SC (Sign Coding) primitive. In this document, we show on an example how those coding passes are jointly used with the coding primitives.

2 Example used

EBCOT coder is a block coder *i.e.* it codes the wavelet coefficients by blocks. In general, the blocks are 64×64. The part of block used in this example is outlined on figure 1. We made the hypothesis that this block is in a *LH* subband. Numbers represent wavelet coefficients values after quantization. These coefficients are scan in a column wise order by columns of four coefficients and

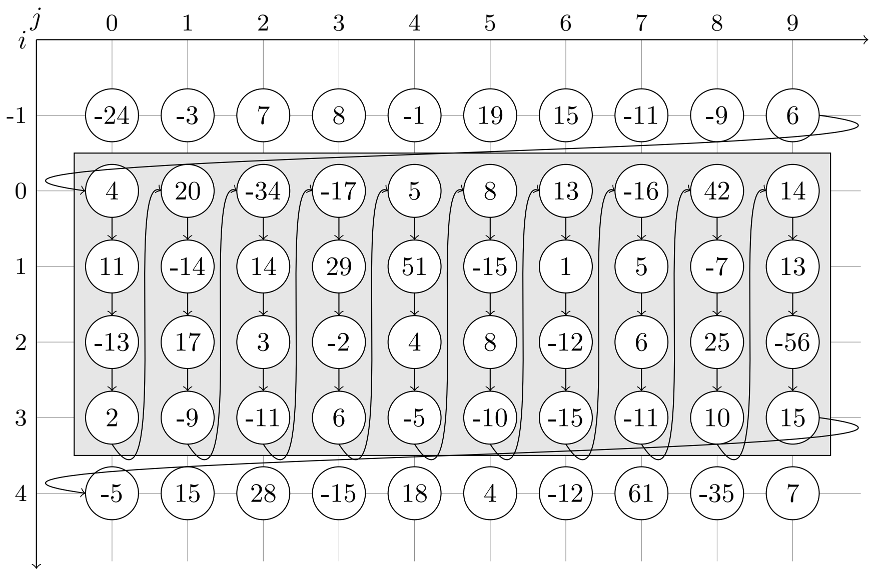

Figure 1: Code-block part used in this example.

strip by strip as illustrated on figure 1 on this figure, the first line of coefficient belongs to the previous strip. The last one belongs to the next strip.

The highest magnitude coefficient is 61 (at the position (4,7)). There are thus 6 magnitude bit-planes in this code-block ($2^6 = 64$). They are numbered from 0 to 5; 5 being the number of the most significant bit-plane. This number of bit-planes is included in the bit-stream in order to signal to the decoder at which bit-plane the coding begins.

3 Coding of the first bit-plane

The first bit-plane ($n = 5$) is displayed on figure 2. On this bit-plane, only the cleanup pass is performed. Indeed, the significance propagation pass cannot be used since there isn't any significant wavelet coefficient before this bit-plane. As no significant wavelet coefficient is known, significance cannot be spread around significant coefficients. The magnitude refinement pass is not used for the same reasons. Significant coefficients must be found before their magnitude can be refined. The only coding pass on the first bit-plane is then the cleanup pass. This pass is called cleanup because it is used to encode all bits that have not been encoded in the significance propagation pass or the magnitude refinement pass.

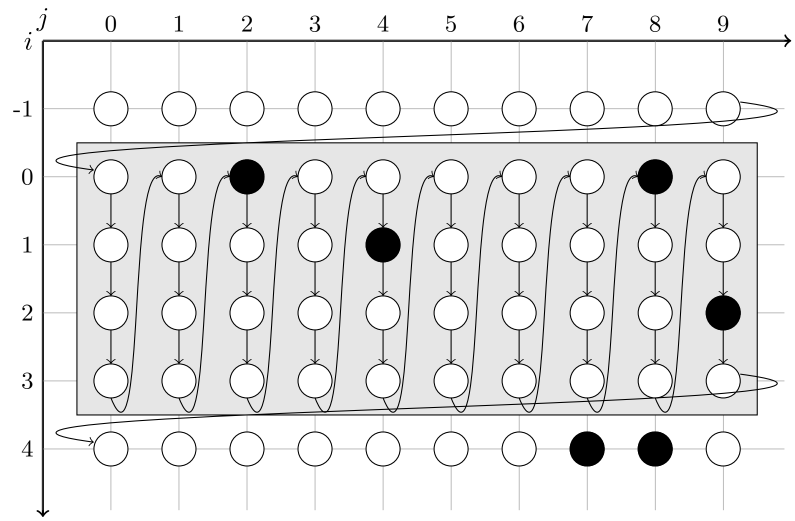

Figure 2: Bit-plane $n = 5$.

3.1 Cleanup pass

The coding process begins by the column $j = 0$. Before the coding of this column, no significant wavelet coefficient is known in this column. Moreover, there is no known significant coefficient in the neighborhood of this column. When those two conditions are satisfied, the coder switches to the *Run-Length* mode. The column $j = 0$ is scanned. At the bit-plane $n = 5$, there is no significant coefficient in this column. Therefore, the coder output the symbol 0 in the (RL) context in order to indicate a column of zeros.

In the same way, no significant coefficient is known in the column $j = 1$ and in its neighborhood. The coder stays in the *Run-Length* mode. as no coefficient of this column is significant at the bit-plane $n = 5$, a 0 symbol is output in the (RL) context.

For the coding the column $j = 2$, the coder is still in the *Run-Length* mode since the two conditions are still satisfied. However, the coefficient (0,2) is significant at the bit-plane $n = 5$. Thus, the coder output the symbol 1 in the (RL) context to indicate that it stops the Run-Length mode. Moreover, the coder indicates the position of the first significant coefficient of the column by the output of the symbols 00 in the (UNI) context. (In the UNIFORM context, symbols are considered equi-probable). This enable the decoder to know the exact position where the Run-Length mode stops. The sign of each wavelet coefficient is coded as soon as the coefficient is found to be significant. Thus, the sign of the coefficient (0,2) is now coded.

For the sign coding, EBCOT uses a prediction based on the signs of the four neighbors. The predictions are given in table 1 for the subbands LL, LH and HH. $\bar{\chi}^h$ and $\bar{\chi}^v$ are the sign of respectively the two horizontally aligned neighbors

$\bar{\chi}^h$	$\bar{\chi}^v$	κ^{SC}	$\hat{\chi}$
1	1	SC4	1
1	0	SC3	1
1	-1	SC2	1
0	1	SC1	1
0	0	SC0	1
0	-1	SC1	-1
-1	1	SC2	-1
-1	0	SC3	-1
-1	-1	SC4	-1

Table 1: Sign prediction $\hat{\chi}$ and contexts κ^{SC} of the SC primitive.

and the two vertically aligned neighbors. Those quantities are inverted in the table for the coding of HL subbands. $\bar{\chi}$ is 1 if the two neighbors are positive or if one neighbor is positive and the other one is not signed yet. $\bar{\chi}$ is 0 if the two neighbors are not signed yet or if they have opposite signs. $\bar{\chi}$ is 1 if the two neighbors are negative or if one neighbor is negative and the other one is not signed yet. $\hat{\chi}$ is the sign prediction of the encoded coefficient. If this prediction is correct, the symbol 0 is output in the respective SC context. Otherwise, the symbol 1 is output in the same context.

In the case of coefficient (0,2), as no neighbour is signed, the context is (SC0). The predicted sign is positive. This prediction is not correct thus the symbol 1 is output in the (SC0) context.

Symbols (1,2) to (3,2) remains in the column $j = 2$. As the coder is no more in the Run-Length mode, the ZC primitive is used. Nine contexts denoted from ZC0 to ZC8 are used by this primitive. Those contexts are based on the *known* significance of the 8-neighbors of the current coded coefficient. They are defined in table 2. κ^h, κ^v and κ^d are respectively the number of horizontal, vertical and diagonal neighbors that are already known significant.

LL and LH subbands			HL subband			HH subband		κ^{ZC}
κ^h	κ^v	κ^d	κ^h	κ^v	κ^d	κ^d	$\kappa^h + \kappa^v$	
0	0	0	0	0	0	0	0	ZC0
0	0	1	0	0	1	0	1	ZC1
0	0	≥ 2	0	0	≥ 2	0	≥ 2	ZC2
0	1	x	1	0	x	1	0	ZC3
0	2	x	2	0	x	1	1	ZC4
1	0	0	0	1	0	1	≥ 2	ZC5
1	0	≥ 1	0	1	≥ 1	2	0	ZC6
1	≥ 1	x	≥ 1	1	x	2	≥ 1	ZC7
2	x	x	x	2	x	≥ 3	x	ZC8

Table 2: κ^{ZC} context of ZC primitive. "x" means any value.

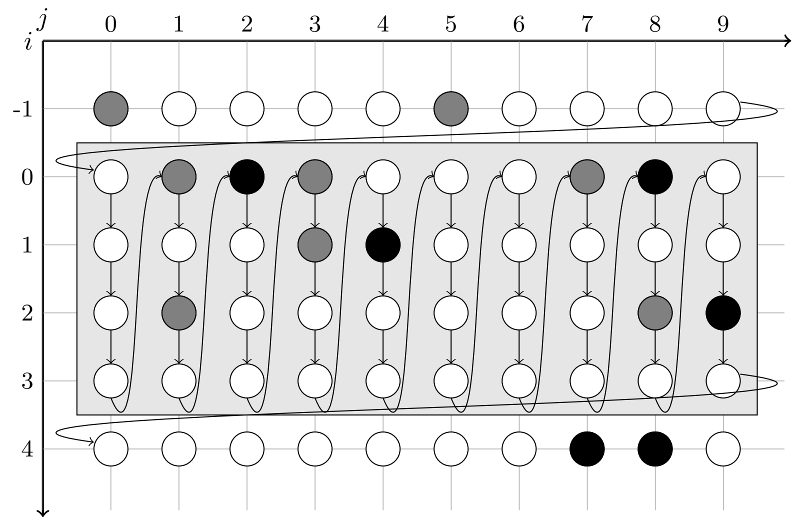

Figure 3: Bit-plane $n = 4$.

Coefficient (1,2) has only one significant neighbour: the coefficient (0,2). Moreover, coefficient (1,2) is not significant at the bit-plane $n = 5$. The symbol 0 is thus output in the (ZC3) context. Coefficients (2,2) and (3,2) do not have any significant neighbour and are not significant at the bit-plane $n = 5$. Two symbols 0 are thus output in the contexts (ZC0).

The coder is now coding the third column $j = 3$. As coefficient (0,2), next to this column, is known significant, the Run-Length mode is not activated. The ZC primitive is thus used to encode the four bits of this column. The coded symbols are 0(ZC5), 0(ZC1), 0(ZC0) and 0(ZC0).

There is no known significant coefficients in the column $j = 4$. Moreover, no neighbour of this column is known significant. The Run-Length mode is thus activated. As the coefficient (1,4) is significant at the bit-plane $n = 5$, the symbol 1 is output in the (RL) context. The symbols 01 are also output in the (UNI) context to signal the position of the first significant coefficient of this column. The sign of this coefficient is then coded. Coefficient (1,4) is positive. The sign context is (SC0). The predicted sign is thus correct. The symbol 0 is thus output in the (SC0) context. The next two bits are coded in the ZC primitive. Coded symbols are 0(ZC3) and 0(ZC0).

The remaining of the coding of bit-plane $n = 5$ by the cleanup pass is described in table 3.

4 Coding of the second bit-plane

The second bit-plane ($n = 4$) is displayed on figure 3. Significant coefficients at previously coded bit-plane (more significant bit-planes) are represented in black.

	Coded symbols and contexts	Comments
$j = 0$	0(RL)	Run-Length.
$j = 1$	0(RL)	Run-Length.
$j = 2$	1(RL) 00(UNI) 1(SC0) 0(ZC3) 0(ZC0) 0(ZC0)	Run-Length until coefficient (0,2). Position of the first significant coefficient. Predicted sign by the context (SC0) is not correct. Remaining of the column coded by the ZC primitive.
$j = 3$	0(ZC5) 0(ZC1) 0(ZC0) 0(ZC0)	No Run-Length since coefficient (0,2) is signifiant.
$j = 4$	1(RL) 01(UNI) 0(SC0) 0(ZC3) 0(ZC0)	Run-Length until coefficient (1,4). Position of the first significant coefficient. Predicted sign by the context (SC0) is correct. Remaining of the column coded by the ZC primitive.
$j = 5$	0(ZC1) 0(ZC5) 0(ZC1) 0(ZC0)	No Run-Length since coefficient (1,4) is signifiant.
$j = 6$	0(RL)	Run-Length.
$j = 7$	0(RL)	Run-Length.
$j = 8$	1(RL) 00(UNI) 0(SC0) 0(ZC3) 0(ZC0) 0(ZC0)	Run-Length until coefficient (0,8). Position of the first significant coefficient. Predicted sign by the context (SC0) is correct. Remaining of the column coded by the ZC primitive.
$j = 9$	0(ZC5) 0(ZC1) 1(ZC0) 1(SC0) 0(ZC3)	No Run-Length since coefficient (0,8) is signifiant. A significant coefficient in context (ZC0). Predicted sign by the context (SC0) is not correct.

Table 3: Coding of bit-plane $n = 5$ by the cleanup pass.

Coefficients found significant at the current bit-plane $n = 4$ are represented in gray. On this second bit-plane, the three coding passes are used: first the significance propagation pass encodes neighbors of coefficients already found significant at the current bit-plane $n = 4$ or at more significant bit-planes. Indeed, high magnitude coefficients are generally found in clusters. That is why significant coefficients are first search around the highest magnitude coefficients already known. Then, the magnitude refinement pass is used to refine the magnitude of coefficients which were already significant at previously encoded bit-planes. Finally, the cleanup pass encode the remaining bits.

4.1 Significance propagation pass

During the scan of bit-plane $n = 4$ begins with coefficient (0,0). Coefficient (-1,0) is a neighbour of this coefficient. It became significant at the bit-plane $n = 4$. The bit at the position (0,0) is thus coded in the significance propagation

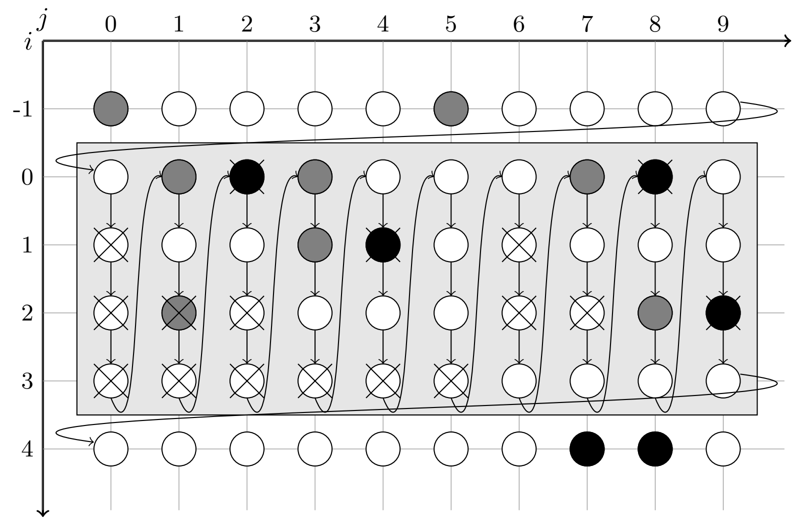

Figure 4: Significant propagation pass on the bit-plane $n = 4$.

pass. In this pass, only the ZC primitive is used. The output symbol to encode the (0,0) bit is thus 0 in the (ZC3) context.

Coefficients (1,0) to (3,0) do not have any *known* significant neighbour coefficients. They are not encoded in the significance propagation pass. Those coefficients are marked by a cross on figure 4.

As coefficient (0,2) is significant, bits at the positions (0,1) and (1,1) are coded. Bit (0,1) is significant at the bit-plane $n = 4$. Symbol 1(ZC6) is thus output. The sign of the wavelet coefficient is then coded by the symbol 1(SC3). Indeed, the predicted sign is negative whereas the coefficient is positive. Symbol 0(ZC3) is then output to encode the bit (1,1).

Coefficients (2,1) and (3,1) are not coded. Coefficient (0,2) is not coded either because it was already significant at the previous bit-plane. Symbol 0(ZC3) is output to encode the bit (1,2). Coefficients (2,2) and (3,2) are not coded.

The remaining of the coding of bit-plane $n = 4$ by the significance propagation pass is described in table 4.

4.2 Magnitude refinement pass

In the magnitude refinement pass, only the MR primitive is used. This primitive encode the bits of already significant coefficients in the previous bit-planes. The three contexts of this primitive are details in table 5. They are noted MR0 to MR2. $\tilde{\sigma}$ is 0 if it is the first time that the magnitude refinement pass is applied on the coded coefficient. Otherwise $\tilde{\sigma}$ is 1. κ^h and κ^v are defined the same way as for the ZC primitive.

The magnitude of coefficients that were significant in the previous bit-planes is refined in this pass. Symbol 0(MR1) is output to refine the magnitude of

	Coded symbols and contexts	Comments
$j = 0$	0(ZC3)	Coefficient (0,0) is coded.
$j = 1$	1(ZC6) 1(SC3) 0(ZC3)	Coefficient (0,1) is coded. The predicted signe in (SC3) context is not correct. Coefficient (1,1) is coded.
$j = 2$	0(ZC3)	Coefficient (1,2) is coded.
$j = 3$	1(ZC6) 0(SC3) 1(ZC7) 0(SC2) 0(ZC3)	Coefficient (0,3) is coded. Coefficient (1,3). Coefficient (2,3).
$j = 4$	0(ZC7) 0(ZC3)	Coefficients (0,4) and (2,4).
$j = 5$	0(ZC3) 0(ZC5) 0(ZC1)	Coefficients (0,5) and (2,5).
$j = 6$	0(ZC1) 0(ZC1)	Coefficients (0,6) and (3,6).
$j = 7$	1(ZC5) 1(SC3) 0(ZC3) 0(ZC3)	Coefficient (0,7). Coefficients (1,7) and (3,7).
$j = 8$	0(ZC3) 1(ZC5) 1(SC3) 0(ZC4)	Coefficient (1,8). Coefficient (2,8). Coefficient (3,8).
$j = 9$	0(ZC5) 0(ZC3) 0(ZC3)	Coefficients (0,9), (2,9), and (3,9).

Table 4: Coding of bit-plane $n = 4$ by the significance propagation pass.

$\tilde{\sigma}$	$\kappa^h + \kappa^v$	κ^{MR}
0	0	MR0
0	$\neq 0$	MR1
1	x	MR2

Table 5: The MR primitive κ^{MR} contexts.

coefficient (0,2). In the same way, symbols 1(MR1), 0(MR1), and 1(MR1) are output to refine the magnitude of coefficients (1,4), (0,8) and (2,9).

4.3 Cleanup pass

The cleanup pass is used to encode bits that were not coded by the previous two passes. It is the bits not marked by crosses on figure 5. The coding pass is described in table 6.

5 Continuation and end of the coding

In the following, the coding process is similar until bit-plane $n = 0$. The MQ arithmetic coder encode each output symbol using probability states of each context. Those probabilities are updated after each coded symbol. This coding process of code-blocks is the first part of EBCOT coder. It is the Tier1. In

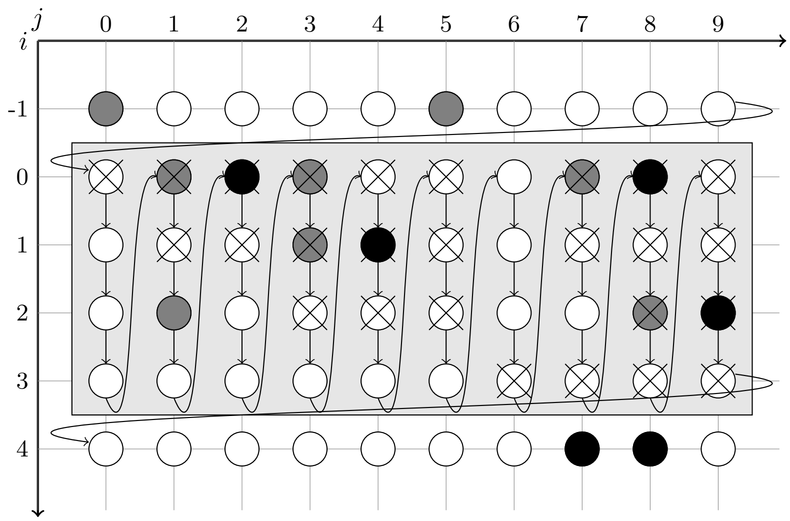

Figure 5: Cleanup pass on bit-plane $n = 4$.

Tier2, pieces of the bit-stream output by the bit-plane encoder are organized so as to optimize the final bit-stream in rate-distortion trade-off. This process is the PCRD-opt (*Post-Compression Rate-Distortion OPTimization*).

	Coded symbols and contexts	Comments
$j = 0$	0(ZC1) 0(ZC0) 0(ZC0)	
$j = 1$	1(ZC0) 0(SC0) 0(ZC3)	Coefficient (2,1).
$j = 2$	0(ZC6) 0(ZC1)	
$j = 3$	0(ZC0)	
$j = 4$	0(ZC0)	
$j = 5$	0(ZC0)	
$j = 6$	0(ZC6) 0(ZC1) 0(ZC0)	
$j = 7$	0(ZC5)	

Table 6: Coding of bit-plane $n = 4$ by the cleanup pass.