7.3.8.8 Palette syntax

|  |  |
| --- | --- |
| palette\_coding( x0, y0, nCbS ) { | **Descriptor** |
| …… |  |
| if( MaxPaletteIndex > 0) { |  |
| **palette\_transpose\_flag** | ae(v) |
| **palette\_rotated\_scan\_flag** | ae(v) |
| **num\_palette\_indices\_idc** | ae(v) |
| for( i=0; i < NumPaletteIndices; i++ ) { |  |
| **palette\_index\_idc** | ae(v) |
| PaletteIndexIdc[ i ] = palette\_index\_idc |  |
| } |  |
| **last\_palette\_run\_type\_flag** | ae(v) |
| } |  |
| …… |  |
| } |  |

7.4.9.6 Palette mode semantics

**palette\_rotated\_scan\_flag** equal to 1 specifies that the rotation process is applied to the current coding unit before reconstruction. palette\_rotated\_scan\_flag equal to 0 specifies that the rotation process is not applied to the current coding unit. When not present, the value of palette\_rotated\_scan\_flag is inferred to be 0.

8.4.1 **General decoding process for coding units coded in intra prediction mode**

Inputs to this process are:

– a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top‑left luma sample of the current picture,

– a variable log2CbSize specifying the size of the current luma coding block.

Output of this process is a modified reconstructed picture before deblocking filtering.

The derivation process for quantization parameters as specified in clause is invoked with the luma location ( xCb, yCb ) as input.

A variable nCbS is set equal to 1  <<  log2CbSize.

Depending on the values of pcm\_flag[ xCb ][ yCb ], palette\_mode\_flag[ xCb ][ yCb ], and IntraSplitFlag, the decoding process for luma samples is specified as follows:

– If pcm\_flag[ xCb ][ yCb ] is equal to 1, the reconstructed picture is modified as follows:

SL[ xCb + i ][ yCb + j ] =   
 pcm\_sample\_luma[ ( nCbS \* j ) + i ]  <<  ( BitDepthY − PcmBitDepthY ), with i, j = 0..nCbS − 1 (8‑12)

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0), if palette\_mode\_flag[ xCb ][ yCb ] is equal to 1, the following order steps apply:

1. The decoding process for palette intra blocks as specified in clause 8.4.4.2.7 is invoked with the luma location ( xCb, yCb ), the variable cIdx set equal to 0, and nCbSX and nCbSY both set equal to nCbS as inputs, and the output is an nCbS x nCbS array of reconstructed palette sample values, recSamples[ x ][ y ], x, y = 0..nCbS − 1
2. The reconstructed picture is modified as follows:

– If palette\_transpose\_flag is true,

* + If palette\_rotated\_scan\_flag is true

SL[ yCb + y ][ xCb + x ] = recSamples[ nCbS- x ][ nCbS- y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SL[ yCb + y ][ xCb + x ] = recSamples[ x ][ y ] (8‑13)

– Otherwise (palette\_transpose\_flag is false)

* + If palette\_rotated\_scan\_flag is true

SL[ yCb + x ][ xCb + y ] = recSamples[ nCbS- x ][ nCbS- y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SL[ xCb + x ][ yCb + y ] = recSamples[ x ][ y ] (8‑14)

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0 ), if IntraSplitFlag is equal to 0, the following ordered steps apply:

... …

… …

When ChromaArrayType is not equal to 0, the following applies.

The variable log2CbSizeC is set equal to log2CbSize − ( ChromaArrayType  = =  3 ? 0 : 1 ).

Depending on the values of pcm\_flag[ xCb ][ yCb ] and IntraSplitFlag, the decoding process for chroma samples is specified as follows:

– If pcm\_flag[ xCb ][ yCb ] is equal to 1, the reconstructed picture is modified as follows:

SCb[ xCb / SubWidthC + i ][ yCb / SubHeightC + j ] =  
 pcm\_sample\_chroma[ ( nCbS / SubWidthC \* j ) + i ]  <<  ( BitDepthC − PcmBitDepthC ),  
 with i = 0..nCbS / SubWidthC − 1 and j = 0..nCbS / SubHeightC − 1 (8‑15)

SCr[ xCb / SubWidthC + i ][ yCb / SubHeightC + j ] =  
 pcm\_sample\_chroma[ ( nCbS / SubWidthC \* ( j + nCbS / SubHeightC ) ) + i ]  <<  
 ( BitDepthC − PcmBitDepthC ),  
 with i = 0..nCbS / SubWidthC − 1 and j = 0..nCbS / SubHeightC − 1 (8‑16)

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0), if palette\_mode\_flag[ xCb ][ yCb ] is equal to 1 the following orderd steps apply:

1. The decoding process for palette intra blocks as specified in clause 8.4.4.2.7 is invoked with the chroma location ( xCb, yCb ), the variable cIdx set equal to 1, nCbSX set equal to nCbS / SubWidthC, and nCbSY set equal to nCbS / SubHeightC as inputs, and the output is an ( nCbS / SubWidthC ) x (nCbS / SubHeightC ) array of reconstructed palette sample values, recSamples[ x ][ y ], x = 0 … nCbS / SubWidthC − 1, y = 0..nCbS / SubHeightC − 1.
2. The reconstructed picture is modified as follows:

– If palette\_transpose\_flag is true, ~~S~~~~Cb~~~~[ yCb + y ][ xCb + x ] is set equal to recSamples[ x ][ y ],~~

* + If palette\_rotated\_scan\_flag is true

SCb[ yCb/ SubHeightC + y ][ xCb/SubWidthC + x ] = recSamples[ nCbS/ WidthC - x ][ nCbS/ SubHeightC - y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SCb[ yCb/SubHeightC + y ][ xCb/SubWidthC + x ] = recSamples[ x ][ y ]

– Otherwise (palette\_transpose\_flag is false), ~~S~~~~Cb~~~~[ xCb + x ][ yCb + y ] is set equal to recSamples[ x ][ y ].~~

* + If palette\_rotated\_scan\_flag is true

SCb[ xCb/SubWidthC + x ] [ yCb/ SubHeightC + y ] = recSamples[ nCbS/ WidthC - x ][ nCbS/ SubHeightC - y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SCb[ xCb/SubWidthC + x ] [ yCb/ SubHeightC + y ] = recSamples[ x ][ y ]

1. The decoding process for palette intra blocks as specified in clause 8.4.4.2.7 is invoked with the chroma location ( xCb, yCb ), the variable cIdx set equal to 2, nCbSX set equal to nCbS / SubWidthC, and nCbSY set equal to nCbS / SubHeightC as inputs, and the output is an ( nCbS / SubWidthC ) x (nCbS / SubHeightC ) array of reconstructed palette sample values, recSamples[ x ][ y ], x = 0 … nCbS / SubWidthC − 1, y = 0..nCbS / SubHeightC − 1.
2. The reconstructed picture is modified as follows:

– If palette\_transpose\_flag is true, ~~S~~~~Cr~~~~[ yCb + y ][ xCb + x ] is set equal to recSamples[ x ][ y ]~~

* + If palette\_rotated\_scan\_flag is true

SCr[ yCb/ SubHeightC + y ][ xCb/SubWidthC + x ] = recSamples[ nCbS/ WidthC - x ][ nCbS/ SubHeightC - y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SCr[ yCb/SubHeightC + y ][ xCb/SubWidthC + x ] = recSamples[ x ][ y ]

– Otherwise (palette\_transpose\_flag is false), ~~S~~~~Cr~~~~[ xCb + x ][ yCb + y ] is set equal to recSamples[ x ][ y ]~~

* + If palette\_rotated\_scan\_flag is true

SCr[ xCb/SubWidthC + x ] [ yCb/ SubHeightC + y ] = recSamples[ nCbS/ WidthC - x ][ nCbS/ SubHeightC - y ]

* + Otherwise (palette\_rotated\_scan\_flag is false)

SCr[ xCb/SubWidthC + x ] [ yCb/ SubHeightC + y ] = recSamples[ x ][ y ]

– Otherwise (pcm\_flag[ xCb ][ yCb ] is equal to 0, palette\_mode\_flag[ xCb ][ yCb ] is equal to 0 ), if IntraSplitFlag is equal to 0 or ChromaArrayType is not equal to 3, the following ordered steps apply:

……

……

9.3.3.1 General

**Table 9‑42 – Syntax elements and associated binarizations**

|  |  |  |  |
| --- | --- | --- | --- |
| palette\_coding( ) | … | … | … |
| palette\_rotated\_scan\_flag | FL | cMax = 1 |
| … | … | … |
|
|
|
|

9.4.1.2.1 General

**Table 9‑47 – Assignment of ctxInc to syntax elements with context coded bins**

| **Syntax element** | **binIdx** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **>= 5** |
| … | … | … | … | … | … | … |
| palette\_rotated\_scan\_flag | 0 | na | na | na | na | na |
| … | … | … | … | … | … | … |