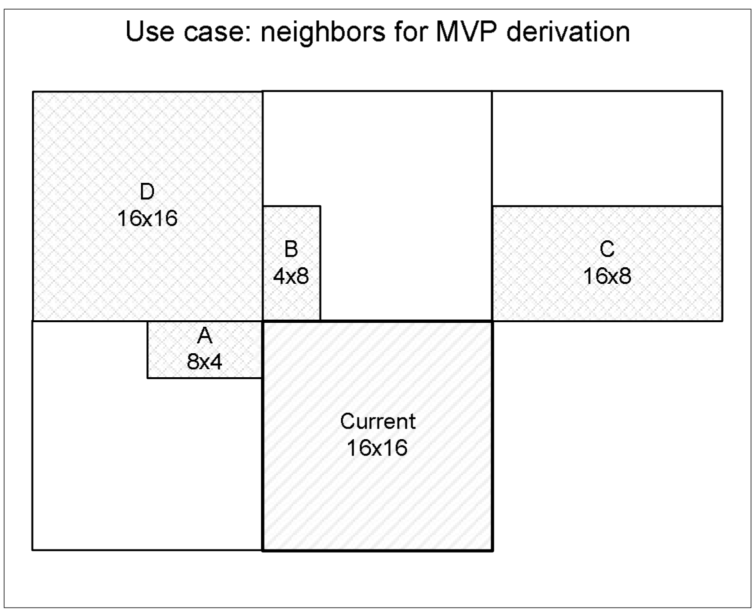
MPEG-4 AVC/H.264 Skip Direct Mode

Part from the book "Anatomy of H.264/AVC"
Written by Shevach Riabtsev
riabtsev@yahoo.com

Skip Mode for P slices

In AVC/H.264 the skip MB in P slice (as well as in B-slices) is indicated by the syntax element **mb_skip_flag** for CABAC entropy mode or **mb_skip_run** for CAVLC entropy mode. If MB is signaled as skip (i.e. mb_skip_flag=1 or mb_skip_run>0) then neither motion data nor residual is present excepting end_of_slice flag in CABAC entropy mode (in order to indicate whether the current MB is the last one in a slice or not). As a decoder detects that a current MB in P-slice is skip it derives motion data and residuals as follows:

- Residuals are derived as zero
- Prediction partition is derived as 16x16
- Reference index is derived as 0 (in L0 direction)
- Motion vector is derived as MVP (motion vector predictor) in non-skip case (the same logic, refer to the section "MPEG-4 AVC/H.264 Motion Vector Prediction"), the figure below illustrates an use case of neighboring blocks used for MVP derivation:

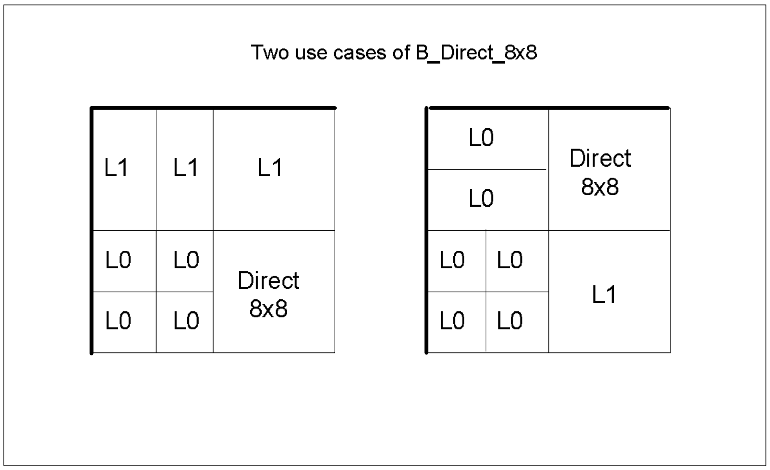


The motion vector predictor (MVP) is either identical to one of the neighboring motion vector or it is the median of the three candidates. Notice that the neighbor block D is taken into consideration only if the neighbor C does not exist.

Skip/Direct Mode for B slices

Unlike P-slice where only skip mode is allowed, in B-slices there an additional and similar mode - direct. The differences between skip and direct modes in B slice are summarized as follows:

- In the direct mode residual data is transmitted while motion data and partitions are derived.
- Direct mode can be specified for finer partition 8x8 block (B_Direct_8x8) while the B-slice skip mode is always applied to 16x16 partition.



In AVC/H.264 bit-stream the skip mode for B slice (B_Skip) is signaled identically to the skip mode in P slice, i.e. by **mb_skip_flag** for CABAC entropy mode or **mb_skip_run** for CAVLC entropy mode. Regarding to the direct mode, B_Direct_16x16 is signaled in mb_type and B_Direct_8x8 is signaled in sub_mb_type respectively.

There are two types of motion data derivation in B-slice direct/skip (the type of derivation is signaled by the syntax element direct_spatial_mv_pred_flag in slice header):

- spatial (actually spatial-temporal), direct_spatial_mv_pred_flag=1
- temporal, direct_spatial_mv_pred_flag = 0

Unlike the skip mode in P-slice, in B-slice skip/direct mode derivation of motion data requires some information from collocated MBs. Therefore H.264/AVC codec has to keep motion information of reference picture(s) (in addition to reference samples). This makes coding/decoding process of B-pictures more complicated and increases memory bandwidth if motion co-located motion data is stored onto off-chip memory. Designers of decoders are to consider where to keep collocated data – on-chip memory or off-chip memory? Notice that if collocated data is a burden an encoder can choose not to code skip/direct mode in order to simplify encoding process.

Tips for H.264/AVC encoder architect how to choose the correct B-slice skip/direct mode (temporal or spatial):

- Temporal mode is expected to be beneficial at static scenes, when motion is continuous and consistent and/or at extremely low QPs.

- Spatial mode is expected to be beneficial for hierarchical picture coding. Moreover the spatial mode requires considerably lower memory storage for co-located data.

For more details on skip/direct refer to the paper A.M. Tourapis, F. Wu, S. Li, "Direct mode coding for bipredictive slices in the H.264 standard," in IEEE Transactions on Circuits and Systems for Video Technology, Volume 15, Issue 1, pp.119-126, Jan. 2005.

1.1 Co-located Motion Data

1.1.1 Derivation of Collocated Picture

The first reference picture in L1 list is used for co-located data for both temporal and spatial direct modes with one exception if current picture is frame and the first reference frame in L1 list is a complementary field pair (i.e. a pair of top and bottom pictures or vice versa) then the co-located picture is specified slightly tricky:

If the current frame is MBAFF and the current MB pair is field (i.e. mb_field_decoding_flag=1) then
Co-located picture for the top MB (first MB in current MB pair) is the top picture in the complementary field pair.
Co-located picture for the bottom MB (second MB in current MB pair) is the bottom reference picture.

Otherwise (current frame is non-MBAFF or MB pair is frame)
If the top reference field is closer to the current frame (in POC distance) then that picture is selected as co-located one.

Else if the bottom reference field is closer to the current frame (in POC distance) then that picture is selected as co-located.

Else (POC distances from both fields to the current frame are equal) then bottom picture is selected as co-located.

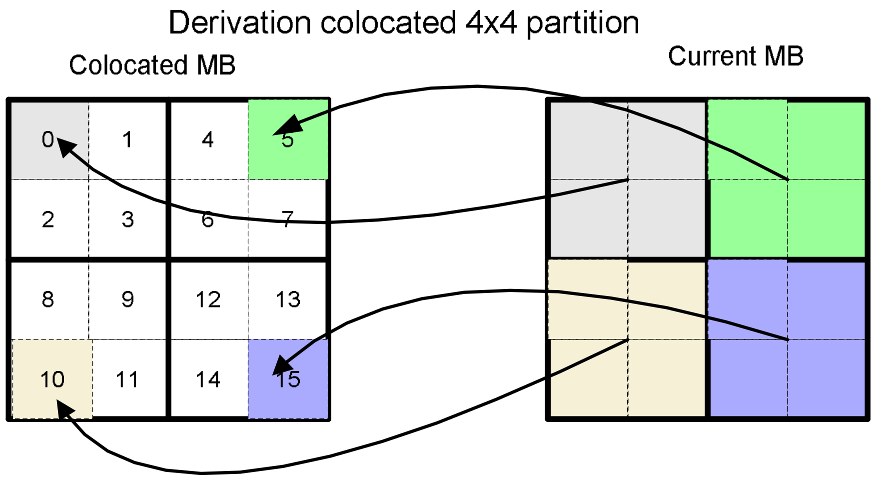
It's common to denote the co-located picture as Pic1.

The reason behind to utilize first pictures in List1 as co-located comes from observations that "best" motion vectors for direct interpolation of IBBPBBP and IBPBP coding structures are obtained from L1 pictures.
Generally speaking pictures in the List0 commonly comes from the past while pictures in the List1 comes from the future and hence suit better for interpolation.
Notice that in progressive case an encoder by means of reordering commands can shuffle L1 reference list such that any selected reference frame can be co-located. Therefore a decoder has to keep motion information from all reference pictures (this might be a significant amount of data to be stored).

AVC/H.264 has two granularities of co-located motion vectors: coarse and fine (each is specified by the syntax element *direct_8x8_inference_flag* in SPS header).

1.1.2 Case direct_8x8_inference_flag=1

If *direct_8x8_inference_flag*=1 then a process of motion data derivation for B_Direct_16x16 is executed for four times (for each 8x8 sub-block).
Each 8x8 partition in B_Direct_16x16/B_Skip or the direct 8x8 sub-block (sub_mb_type is B_Direct_8x8) refers to single co-located 4x4 block containing necessary co-located motion data, the corresponding co-located 4x4 block is illustrated graphically as follows:



Analytically collocated 4x4 block index can be expressed as 5 * 8x8PartitionIdx.

The following table contains co-located data to be held for each MB in each potential co-located reference picture for the sequence with *direct_8x8_inference_flag* =1:

Table 1: Database of co-located data

Partition Index of 4x4 Block	Field Name	Field Description	Number of Bits
NA	FieldFlagCol	Required for MV scaling. 1 – if co-located MB is field 0 - otherwise Frame/Field attribute of the Relevant only for MBAFF sequences	1
NA	ColIntra	Co-located MB is intra	1
0	mvxCol	co-located motion vector of 4x4 block #0, (x – component)	14
	mvyCol	co-located motion vector of 4x4 block #0, (y – component)	12
	RefPOCCol	POC of reference picture where co-located block point according to ref_idx_lx	16
	RefPicParityCol	reference picture parity (0 – top or frame, 1- bottom) where co-located block point according to ref_idx_lx	1
5	Same as for 8x8 Partition #0		
10	Same as for 8x8 Partition #0		
15	Same as for 8x8 Partition #0		
Total: 44x4 + 2 = 178 bits (per MB)			

Notes:

- If co-located block is coded as Intra then its motion vector is set to zero.
- If co-located block is L0-predicted or Bi-predicted then POC of the reference picture associated with ref_idx_l0 is assigned to RefPOCCol. Also forward MVs are assigned to mvCol (mvxCol,mvyCol).
- If co-located block L1-predicted POC of the reference picture associated with ref_idx_l1 is assigned to RefPOCCol. Also backward MVs are assigned to mvCol (mvxCol,mvyCol).
- For progressive streams MBFieldFlagCol and RefPicParityCol are not relevant and can be removed.

1.1.3 Case direct_8x8_inference_flag=0

In this mode the derivation of motion data for B_Direct_16x16/B_Skip is executed for each 4x4 sub-block (i.e. sixteen times). For B_Direct_8x8 the derivation is invoked for each 4x4 block (i.e. four times). The following table contains co-located data to be held for each MB in each reference picture for the sequence with *direct_8x8_inference_flag* =0:

Partition Index of 4x4 Block	Field Name	Field Description	Number of Bits
NA	FieldFlagCol	Required for MV scaling. 1 – if co-located MB is field 0 - otherwise Frame/Field attribute of the Relevant only for MBAFF sequences	1
NA	ColIntra	Co-located MB is intra	1
0	mvxCol	co-located motion vector (x – component)	14
	mvyCol	co-located motion vector (y – component)	12
	RefPOCCol	POC of reference picture where co-located block point according to ref_idx_lx	16
	RefPicParityCol	reference picture parity (0 – top or frame, 1- bottom) where co-located block point according to ref_idx_lx	1
1	Same as for 4x4 Partition #0		
2	Same as for 4x4 Partition #0		
3	Same as for 4x4 Partition #0		
…..	……..		
15	Same as for 4x4 Partition #0		
Total: 32x16 + 2= 706 bits (per MB)			

Notes:

- If co-located block is coded as Intra then its motion vector is set to zero.
- If co-located block is L0-predicted or Bi-predicted then POC of the reference picture associated with ref_idx_l0 is assigned to RefPOCCol. Also forward MVs are assigned to mvCol (mvxCol,mvyCol).
- If co-located block L1-predicted POC of the reference picture associated with ref_idx_l1 is assigned to RefPOCCol. Also backward MVs are assigned to mvCol (mvxCol,mvyCol).
- For progressive streams MBFieldFlagCol and RefPicParityCol are not relevant and can be removed.

1.1.4 Determination of co-located MB index and partition

If current picture is frame/field and co-located picture is frame/field respectively then collocated MB and partition index equals to the current MB number and the current partition index. **Throughout this section the partition indexes are numbered in raster scan (this numbering simplifies many calculations).**

Coding Order

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Raster Order

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

The mapping from current index to collocated one gets sophisticated in mixed cases, e.g. when field and frame pictures are interleaved.
Let's consider an arbitrary MB at MB-row R with index offset C (column index is relative to the MB row start) and an arbitrary 4x4 partition part4x4Idx in raster scan.
Notice that for *direct_8x8_inference_flag*=0 all 4x4 partitions 0..15 (in raster order) are considered, for *direct_8x8_inference_flag=1* partitions 0,2,8 and 10 (in raster order) are taken.

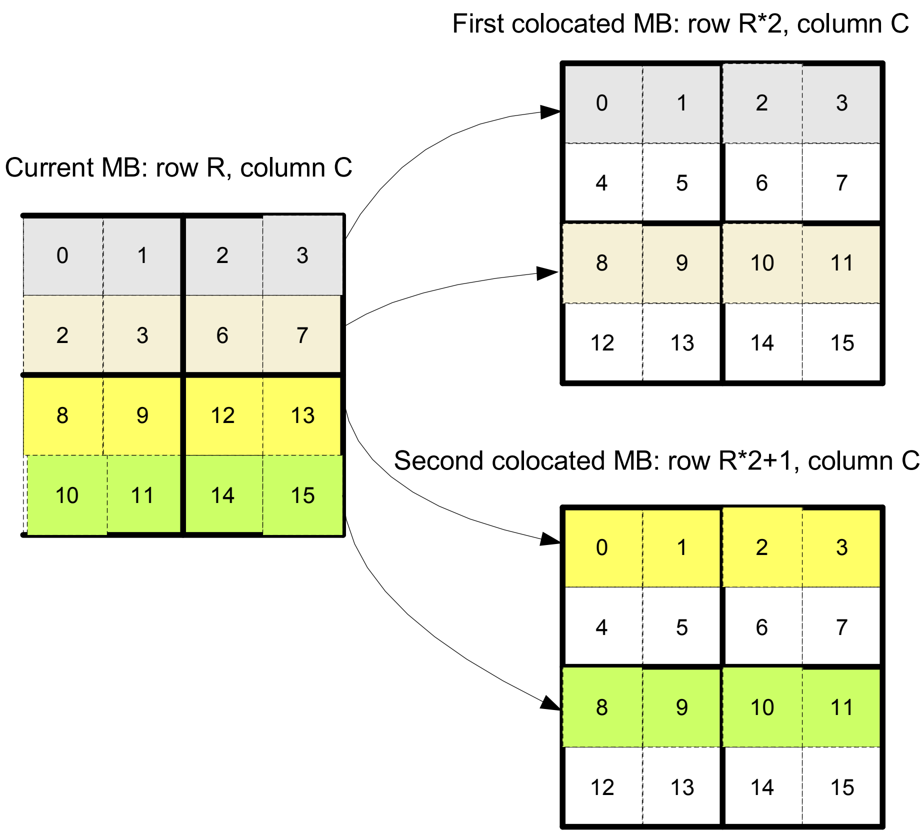
1.1.4.1 Use Case: current picture is field and collocated picture is frame

Two collocated MBs are exploited, the both MBs have the same column index C as the current one, however the row of the first collocated MB is R*2 and the second one is R*2+1 respectively. Determination of collocated block is executed in two steps:

- In the first step we select the collocated MB as follows:
  If partition index part4x4Idx<8 then the first collocated MB is selected, i.e. MB with coordinates (R*2, C).
  Otherwise the MB with coordinates (R*2+1,C) is selected.

- Upon selection of the co-located MB the collocated 4x4 partition within that MB is derived as follows:
  colx4PartIdx = ((part4x4Idx & 0xC) + part4x4Idx) % 16

The following figure illustrates mapping of the above method:



Notes:

- Y-coordinate of co-located MVs must be scaled by two (vertMvScale = Frm_To_Fld).

- Blocks #4 - #7 and #12 - #15 are useless for direct derivation. If a frame picture is used as reference only for field pictures then motion data of blocks #0 - #3 and blocks #8 - #11 should be stored for direct prediction, the rest is useless.

1.1.4.2 Current picture is field and collocated picture is MBAFF

TBD

1.1.4.3 Current picture is MBAFF and collocated picture is MBAFF

TBD

1.1.4.4 Current picture is frame and collocated picture is field

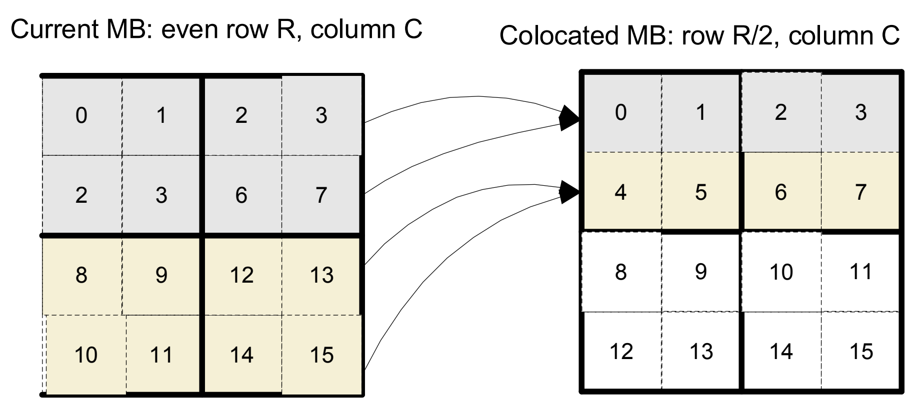
For current MB with coordinates (R,C) the collocated MB with coordinates (R/2,C) is selected. The partition index is mapped as follows:

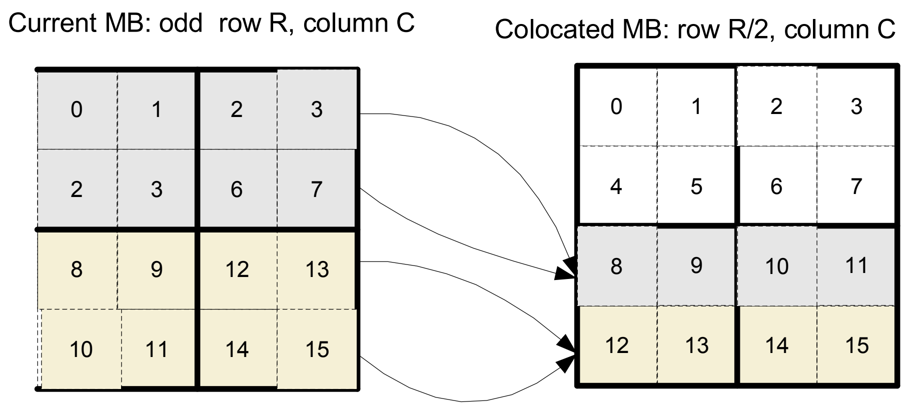
If R is even then

colx4PartIdx = (part4x4Idx % 4)+ 4 * (part4x4Idx >= 8)

Otherwise

colx4PartIdx = (part4x4Idx % 4)+ 4 * (part4x4Idx >= 8) + 8





Notes:

- Y-coordinate of co-located MVs must be divided by two (vertMvScale = Fld_To_Frm).

1.1.5 Conclusions

The mode direct_8x8_inference_flag=1 is beneficial for memory usage and performance (four invocation of motion vectors derivation) while coding efficiency might be deteriorated against direct_8x8_inference_flag=0.
So, direct_8x8_inference_flag=0 causes a burden on decoding process (on both performance and memory usage). This is a reason why direct_8x8_inference_flag is restricted to one for level 3 and higher.

1.2 Temporal Mode

Temporal direct mode is used if **direct_spatial_mv_pred_flag** is equal to 0 (remind **direct_spatial_mv_pred_flag** is signaled in slice header). The idea of this mode is inherited from the direct mode of MPEG-4. However the calculations in AVC/H.264 were simplified relative to MPEG-4.
Temporal direct mode is derived by scaling of collocated MVs in the subsequent (L1) reference frame. Temporal direct mode is expected to be useful when objects are moving with constant speed across pictures.

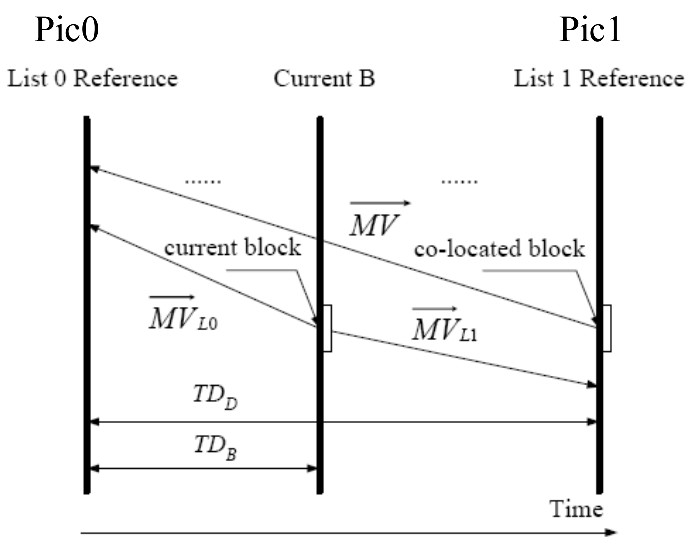


Figure 1

In the above figure:

- TD_B is the temporal distance in POCs between the current picture and the reference picture pointed by the co-located block (Pic0).
- TD_D is the temporal distance in POCs between the co-located picture Pic1 and the reference picture (Pic0) pointed by the co-located block.

The number of calculated motion vectors for each 8x8 partition in B direct/skip mode (B_Direct_16x16, B_Skip, B_Direct_8x8) depends on the *direct_8x8_inference_flag* syntax element:

- If *direct_8x8_inference_flag*=0, reference indexes and two motion vectors (forward and backward) are calculated for every 4x4 sub-partition in all 8x8 partitions.
- I*f direct_8x8_inference_flag=1*, reference indexes and two motion vectors (forward and backward) are calculated for the top-left 4x4 sub-partition in the corresponding 8x8 partition. The same motion vectors and reference indexes are shared among all the 4x4 sub-partitions inside the current 8x8 temporal direct partition.

Motion vector calculation for temporal direct 4x4-partition is done in two main steps:

- Derivation of reference indexes for L0 and L1 direction: ref_idx_l0 and ref_idx_l1.
- Derivation of forward and backward motion vectors.

1.2.1 Derivation of reference indexes

Output of this section is L0 and L1 reference indexes (ref_idx_l0 and ref_idx_l1) associated with the current 4x4 block.
For the current 4x4 block find the co-located picture Pic1 (see 1.1.1), co-located MB and partition (see 1.1.4) and extract the following parameters from the co-located database (see 1.1.2Table 1: Database of co-located data):

- RefPOCCol
- RefPicParityCol (top field or bottom)
- FieldFlagCol (whether collocated MB is field or not)
- ColIntra (whether co-located MB is intra)

Let's define the new variables - **current_mb_is_second_in_pair** and **CurPicBotFlag**.

curMbIsSecondInPair:
If the current picture is MBAFF then **curMbIsSecondInPair** is 0 if the current MB is the first MB in a pair
Otherwise **curMbIsSecondInPair** is set to 1.
Notice that the variable **curMbIsSecondInPair** is non-applicable for Non-MBAFF picture.

CurPicBotFlag:
0 – if current picture is top field or frame
1 – otherwise (current picture is bottom)

1.2.1.1 Derivation of ref_idx_l1

L1 reference index is always to zero, i.e. ref_idx_l1 = 0.

1.2.1.2 Derivation of ref_idx_l0

Unlike the derivation of ref_idx_l1, some logic is required to deduce ref_idx_l0. We limit ourselves with several use cases.

Case 1: Co-located block is intra coded (i.e. ColIntra=1)
ref_idx_l0 = 0

Case 2: Current picture is field and co-located one is also field
Or **Current picture is frame and co-located is also frame**

Find in the reference pictures list the picture with POC equal to RefPOCCol, actually Pic0 (see Figure 1)
The index of found picture is assigned to **ref_dix_l0**.

Case 3: Current picture is MBAFF and current MB pair is field

Find in the reference pictures list the picture with POC equal to RefPOCCol , actually Pic0 (see Figure 1)
The index of found picture is assigned to the temporal variable **TmpIdx**.
ref_idx_l0 = (RefPicParityCol == **curMbIsSecondInPair**) ? 2 * **TmpIdx** : 2 * **TmpIdx** + 1;

1.2.2 Derivation of MV

Output of this section is forward and backward motion vectors associated with the current 4x4 block – mvxL0, mvyL0, mvxL1 and mvyL1.
For the current 4x4 block find the co-located picture Pic1 (see 1.1.1), co-located MB and partition (see 1.1.4) and extract the following collocated motion data from the database specified in 1.1.2Table 1: Database of co-located data: mvxCol, mvyCol.
With ref_idx_l0 calculated in the above section we identify Pic0 picture (see Figure 1):

If Pic0 is long-term picture then
{
mvxL0 = mvxCol; mvyL0 = mvyCol
mvxL1 = 0; mvyL1 = 0
}
Else If POCs of Pic0 and Pic1 coincide
{
mvxL0 = mvxCol; mvyL0 = mvyCol
mvxL1 = 0; mvyL1 = 0
}
Else
{
// clip TD_D to -128 to 127
If (TD_D > 127) Then TD_D = 127
Else If (TD_D < -128) Then TD_D = -128

// clip TD_B to -128 to 127
If (TD_B > 127) Then TD_B = 127
Else If (TD_B < -128) Then TD_B = -128

// TD_D is excluded, already treated above
TX = (16 384 + Abs(TD_D / 2)) / TD_D
DistScaleFactor = int((TD_B *TX+32)/64)

// clip *DistScaleFactor* to -1024 to 1023

If (*DistScaleFactor* > 1023) Then *DistScaleFactor* = 1023

Else if (*DistScaleFactor* < -1024) Then *DistScaleFactor* = -1024

mvxL0 = (*DistScaleFactor* * mvxCol + 128) >> 8
mvyL0 = (*DistScaleFactor* * mvyCol + 128)>> 8
mvxL1 = mvL0 – mvxCol
mvyL1 = mvL0 – mvyCol

}

Note:
The standard avoids using of DistScaleFactor for derivation of Mvs if Pic0 is long-term. The rationale is that the POC of a long-term reference picture is considered not relevant to "temporal" computations.

1.3 Spatial Mode

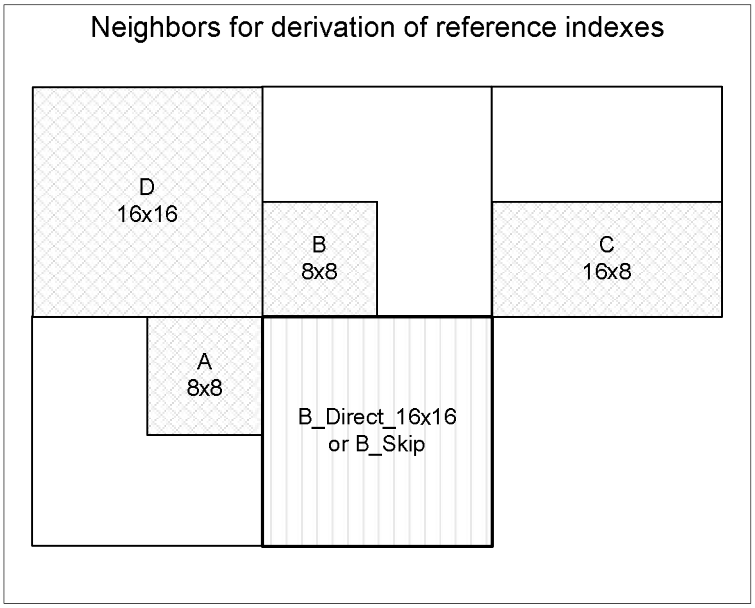
As we already mentioned spatial direct mode is signaled in slice header by direct_spatial_mv_pred_flag = 1.
The derivation of motion data in spatial direct mode is divided into two stages:

- Derive reference indexes
- Derive motion vectors

Notice that the above stages can not be executed in parallel since the derived motion vector is depending on the derived reference index.

1.3.1 Derivation of Reference Indexes

Unlike motion vectors the reference indices (refIdxL0 and refIdxL1) in spatial direct mode are calculated from the neighboring partitions. For B_Direct_16x16/B_Skip the reference indexes are derived once, i.e. all 8x8 blocks share the same reference indexes. The derivation of the reference indexes is illustrated below:



Notes

Like in the derivation of motion vector prediction if C-neighbor is not available then D-neighbor instead (and denoted C-neighbor).

1.3.1.1 Pseudo Code for Derivation of refIdxL0

```
For each neighbor N = A, B and C do
{
        If N is not available, i.e. out of slice/picture, non-causal (not yet decoded), coded in Intra mode or L1-predicted
        {
                refIdxL0N = -1
        }
        Else
        {
                Assign refIdxL0N to L0 reference index of current neighbor
        }

        If all following conditions are met:
                the current frame is MBAFF and
                the current MB mb_field_decoding_flag =1 and
                N-neighbor's mb_field_decoding_flag = 0
        {
                        RefIdxL0N = ( RefIdxL0N != -1 ) ? RefIdxL0N * 2 : RefIdxL0N;
        }

        If all following conditions are met:
                the current frame is MBAFF and
                the current MB mb_field_decoding_flag =0 and
                N-neighbor's mb_field_decoding_flag = 1
        {
                        RefIdxL0N = ( RefIdxL0N != -1 ) ? RefIdxL0N / 2 : RefIdxL0N;
        }

}

RefIdxL0 = MINPOS ( RefIdxL0A, MINPOS ( RefIdxL0B, RefIdxL0C ) );
```

Notes:
MINPOS (x, y) = MIN (x, y) if x >= 0 and y >= 0, otherwise MAX (x, y)

1.3.1.2 Pseudo Code for Derivation of refIdxL1

The derivation of refIdxL1 is the same as the derivation of refIdxL0 except replacement of L0 with L1.

1.3.2 Derivation of Motion Vectors

TBD