CHAPTER 7

REUSE ENGINE

7.1. Introduction

Instead of performing the standalone motion estimation or mode decision, the decoded syntax elements can be reused in encoder based on the resizing ratio as they have enough information about the video to be encoded again. This chapter explains the reuse engine to modify the decoded syntax elements for given resizing ratio.

![Fig. 7.1 Reuse Engine in a Transcoder](image)

In Fig. 7.1, the syntax elements for each block of the video that are parsed from input H.264 bitstream are Width and Height of decoded frame, MB type, sMB type, Intra Luma and Chroma modes, MVs, coded block pattern and QP. The research work includes all those parameters and modifies to required syntax elements based on resizing ratio. These modified syntax elements are used to encode the compressed domain resized frame to the best and optimized bitstream with reduced complexity. The intra-in-inter frame, SKIP MBs, and H.264 Standard compliance are addressed in addition to the general usage of syntax elements. The reuse techniques can be used in arbitrary resizing frames also.

7.2. Algorithm for reusing H.264 Syntax Elements

The algorithm for reuse techniques involves four steps, as follows.
1. Extraction of syntax elements from decoder (Section 7.3)
2. Decision of MB type, sMB type, and QP (Section 7.4)
3. Decision of Intra modes and mode refinement (Section 7.5)
4. Decision of Inter modes and mode refinement (Section 7.6)

The modified syntax elements (newmbtypemat, newsubmbtype, newQPmat, modeeY, modeeC, mvye and mvxe) are sent to compressed domain encoder for encoding the compressed domain resized frame.

7.3. Extraction of syntax elements from Decoder

The input syntax elements, which are extracted from decoder, are listed down.

1. declumawd – Width of decoded Luma frame.
2. declumaht – Height of decoded Luma frame.
3. mbtypemat – MB Type of each MB.
4. submbtypemat – sMB type of each 8x8 sMB.
5. QPmat – QP of each MB.
6. mvxd – x-MV of each 4x4 sMB.
7. mvyd – y-MV of each 4x4 sMB.
8. modedY – Luma mode of each 4x4 sMB.
9. modedC – Chroma mode of each 8x8 Chroma MB.

1st and 2nd syntax elements are obtained from SPS and kept same for the entire sequence. 3rd to 9th syntax elements are calculated in MB Layer of Slice Data for each MB. Those are stored in appropriate places while decoding and modified in reuse engine to the required syntax elements.

The following parameters which are required for reuse engine are derived from Resizer.

1. enclumawd – The Luma width of resized frame
2. enclumaht – The Luma height of resized frame

The following parameters are calculated to know the ratios and number of blocks in row wise and column wise in the reuse engine.

The number of 4x4 blocks in width of decoded frame is, 
\[ \text{decsMBwd} = \frac{\text{declumawd}}{4} \]
The number of 4x4 blocks in height of decoded frame is, \[ \text{decsMBht} = \frac{\text{declumaht}}{4} \]

The number of 4x4 blocks in width of resized frame is, \[ \text{encsMBwd} = \frac{\text{encluma wd}}{4} \]

The number of 4x4 blocks in width of resized frame is, \[ \text{encsMBht} = \frac{\text{enclumaht}}{4} \]

The height ratio is, \[ \text{htratio} = \frac{\text{decsMBht}}{\text{encsMBht}} \]

The weight ratio is, \[ \text{wdratio} = \frac{\text{decsMBwd}}{\text{encsMBwd}} \]

### 7.4. Calculation of mbtype, QPmat and submbtype of resized frame

The parameters, like the MB type, sMB type and QP are calculated from decoded information for each MB of the encoder.

At first, the Syntax Elements are extracted as below.

- \( rMB = \frac{\text{enclumaht}}{16} \), is number of MBs in a column in encoder
- \( cMB = \frac{\text{encluma wd}}{16} \), is number of MBs in a row in encoder

For each MB of the encoder, the corresponding MB types, sMB types and QPs are extracted from decoder appropriately.

\[
\begin{align*}
\text{candmbtype} &= \text{mbtypemat}(i, j) \\
\text{candsubmbtype} &= \text{submbtypemat}(i, j, 1:4) \\
\text{candQP} &= \text{QPmat}(i, j)
\end{align*}
\]

where

\[
\begin{align*}
i &= \text{rowstart} \ldots \text{rowend and } j &= \text{colstart} \ldots \text{colend} \\
\text{rowstart} &= \lfloor \text{row} \times \text{htratio} \rfloor, \quad \text{row} = 0 \ldots rMB - 1 \\
\text{rowend} &= \lfloor (\text{row} + 1) \times \text{htratio} \rfloor - 1, \quad \text{row} = 0 \ldots rMB - 1 \\
\text{colstart} &= \lfloor \text{col} \times \text{wdratio} \rfloor, \quad \text{col} = 0 \ldots cMB - 1 \\
\text{colend} &= \lfloor (\text{col} + 1) \times \text{wdratio} \rfloor - 1 \quad \text{col} = 0 \ldots cMB - 1
\end{align*}
\]

MB Type is calculated for each new MB as follows. The highest MB type among the \( \text{candmbtype} \) is decided as MB type and stored in \( \text{newmbtypemat} \). If all candidate MB types are different, P8x8 is considered as MB type and stored.

The sMB types are also calculated for each new sMB as follows. If the MB type is P8x8, its sMB types are found out considering the candidate sMB types. For each 8x8
sMB, the highest occurrence among \textit{candsubmbtype} is decided. If all candidate sMB types are different, P4x4 is considered as sMB type. The decided sMB types are stored in \textit{newsubmbtypemat}.

Then QP is calculated for new MB. The QP is the representation index of quantization used in H.264 compression. Finding new QP with the help of candidate QPs will not be ideal. So instead of QP, QSTEPs are used, where QSTEP has exponential variation. So, the QP in the \textit{candQP} are converted to QSTEP with the help of Table 7.1.

<table>
<thead>
<tr>
<th>QP</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>QStep</td>
<td>0.625</td>
<td>0.6875</td>
<td>0.75</td>
<td>0.875</td>
<td>0.9375</td>
<td>1.0625</td>
<td>1.25</td>
</tr>
<tr>
<td>QP</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>QStep</td>
<td>1.375</td>
<td>1.5625</td>
<td>1.75</td>
<td>1.9375</td>
<td>2.1875</td>
<td>2.5</td>
<td>2.75</td>
</tr>
<tr>
<td>QP</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>QStep</td>
<td>3.125</td>
<td>3.5</td>
<td>3.9375</td>
<td>4.4375</td>
<td>5</td>
<td>5.5625</td>
<td>6.25</td>
</tr>
<tr>
<td>QP</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>QStep</td>
<td>7.0625</td>
<td>7.875</td>
<td>8.875</td>
<td>10</td>
<td>11.1875</td>
<td>12.5625</td>
<td>14.125</td>
</tr>
<tr>
<td>QP</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>QStep</td>
<td>15.8125</td>
<td>17.8125</td>
<td>20</td>
<td>22.4375</td>
<td>25.1875</td>
<td>28.25</td>
<td>31.6875</td>
</tr>
<tr>
<td>QP</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>QStep</td>
<td>35.625</td>
<td>40</td>
<td>44.875</td>
<td>50.375</td>
<td>56.5625</td>
<td>63.4375</td>
<td>71.25</td>
</tr>
<tr>
<td>QP</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>QStep</td>
<td>80</td>
<td>89.75</td>
<td>100.75</td>
<td>113.125</td>
<td>126.9375</td>
<td>142.5</td>
<td>160</td>
</tr>
<tr>
<td>QP</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td>QStep</td>
<td>179.5625</td>
<td>201.5625</td>
<td>226.25</td>
<td>250</td>
<td>275</td>
<td>300</td>
<td>325</td>
</tr>
</tbody>
</table>

The QSTEPs are averaged to get QSTEP for the given MB. This QSTEP is converted to QP with the help of Table 7.1. This new QP is stored in \textit{newQPmat}. The flowchart in Fig. 7.2 explains the decision of QP, MB Type and sMB type for a given MB.
Fig. 7.2 (a) Flowchart of Macroblock type and QP decision
Consider P8x8 as Macroblock Type

Determine Sub-Macroblock Type Matrix

Is a recurring Sub-Macroblock type present?

YES

Determine Sub-Macroblock Type

NO

Consider P4x4 as Sub-Macroblock Type

Determine Candidate Luma Modes

Are more than half the numbers of candidate Luma modes inter modes?

YES

Declare Luma Mode As Inter Mode

NO

Fig. 7.2 (b) Flowchart (Contd) of sMB Type and Luma Mode Decision
Fig. 7.2 (c) Flowchart (Contd) for Intra Mode Type Decision
7.5. **Intra Mode Decision**

If the modified MB Type is Intra, then Intra mode decision is performed as follows: (i) Extraction of Decoded Modes, (ii) Intra Mode Decision, and (iii) Intra Mode Refinement for Luma and Chroma. The following section explains about Luma and the next section details about Chroma mode decision.

### 7.5.1. Extraction of Decoded Luma Modes

Notations of all the decoded modes are listed in Table 7.2. \textit{modedY} has decoded modes of each 4x4 sMB. The Luma Mode for each 4x4 sMB of encoding frame is decided as follows. The decoded Luma Modes corresponding to each 4x4 sMB of encoding frame are extracted from \textit{modedY} as shown in equation (7-2).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Mode</th>
<th>Notation</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>I4x4 V</td>
<td>12</td>
<td>I8x8 DDL</td>
</tr>
<tr>
<td>1</td>
<td>I4x4 H</td>
<td>13</td>
<td>I8x8 DDR</td>
</tr>
<tr>
<td>2</td>
<td>I4x4 DC</td>
<td>14</td>
<td>I8x8 VR</td>
</tr>
<tr>
<td>3</td>
<td>I4x4 DDL</td>
<td>15</td>
<td>I8x8 HD</td>
</tr>
<tr>
<td>4</td>
<td>I4x4 DDR</td>
<td>16</td>
<td>I8x8 VL</td>
</tr>
<tr>
<td>5</td>
<td>I4x4 VR</td>
<td>17</td>
<td>I8x8 HU</td>
</tr>
<tr>
<td>6</td>
<td>I4x4 HD</td>
<td>18</td>
<td>I16x16 V</td>
</tr>
<tr>
<td>7</td>
<td>I4x4 VL</td>
<td>19</td>
<td>I16x16 H</td>
</tr>
<tr>
<td>8</td>
<td>I4x4 HU</td>
<td>20</td>
<td>I16x16 DC</td>
</tr>
<tr>
<td>9</td>
<td>I8x8 V</td>
<td>21</td>
<td>I16x16 P</td>
</tr>
<tr>
<td>10</td>
<td>I8x8 H</td>
<td>22</td>
<td>P16x16 / P16x8 / P8x16 / P8x8 / P8x4 / P4x8 / P4x4</td>
</tr>
<tr>
<td>11</td>
<td>I8x8 DC</td>
<td>23</td>
<td>PSKIP</td>
</tr>
</tbody>
</table>

\( \textit{candmodes} = \textit{modedY}(i,j) \) ........................................................................................................ (7-2)

where,

\[
i = \text{rowstart} \ldots \text{rowend} \text{ and } j = \text{colstart} \ldots \text{colend}
\]

\[
\text{rowstart} = \lfloor \text{row} \times \text{hratio} \rfloor, \quad \text{row} = 0 \ldots 4 \times rMB - 1
\]

\[
\text{rowend} = \lfloor (\text{row} + 1) \times \text{hratio} \rfloor - 1, \quad \text{row} = 0 \ldots 4 \times rMB - 1
\]

\[
\text{colstart} = \lfloor \text{col} \times \text{wratio} \rfloor, \quad \text{col} = 0 \ldots 4 \times cMB - 1
\]

\[
\text{colend} = \lfloor (\text{col} + 1) \times \text{wratio} \rfloor - 1 \quad \text{col} = 0 \ldots 4 \times cMB - 1
\]
7.5.2. Intra Luma mode decision for a sMB

If more than half of the numbers of notations in `candmodes` are Intra, then Intra is declared. The mode strength is calculated by averaging `candmodes`. Then notation of sMB is decided as follows. The flowchart of Intra mode decision is shown in Fig. 7.3.

\[
modeeY(sMB) = \begin{cases} 
23, & \text{PSIP, } \text{modestrength} \geq 22 \\
22, & \text{Inter, } \text{modestrength} \geq 21 \\
20, & \text{16 x 16, } \text{modestrength} \geq 18 \\
18, & \text{18 x 8, } \text{modestrength} \geq 9 \\
14, & \text{14 x 4, } \text{else}
\end{cases}
\]

![Flowchart for Intra Luma Mode Decision](image)

Fig. 7.3 Flowchart for Intra Luma Mode Decision
The H.264 intra modes are denoted by angles in this research work which is shown in Table 7.3. The cumulative angle of each 4x4 sMB is estimated from the angles of each modes in candmodes. The decided angle leads to the modified intra mode.

### Table 7.3 Angles for Intra Luma Modes

<table>
<thead>
<tr>
<th>V</th>
<th>H</th>
<th>DC</th>
<th>DDL</th>
<th>DDR</th>
<th>VR</th>
<th>HD</th>
<th>VL</th>
<th>HU</th>
<th>V</th>
<th>H</th>
<th>DC</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>135°</td>
<td>45°</td>
<td>0°</td>
<td>180°</td>
<td>90°</td>
<td>112.5°</td>
<td>67.5°</td>
<td>157.5°</td>
<td>22.5°</td>
<td>135°</td>
<td>45°</td>
<td>0°</td>
<td>90°</td>
</tr>
</tbody>
</table>

The steps for intra mode decision are given below.

i. The angle for all H264 intra mode of candmodes is found out from Table 7.3.

ii. The number of DC modes among candmodes is calculated.

iii. If the number of DC modes is more than half of number candidate modes, then DC mode is decided.

iv. Else the following is done.

   a. The average of angles of remaining modes, resultant mode \( r \), is found.

   b. H.264 Intra Mode is decided based on ‘\( r \)’ as shown equation (7-4).

\[
modeeY(sMB) = \begin{cases} 
2, & r < 11.25° \\
8, & 11.25° \leq r < 33.75° \\
1, & 33.75° \leq r < 56.25° \\
6, & 56.25° \leq r < 78.75° \\
4, & 78.75° \leq r < 101.25° \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldot
ii. If the highest mode is I8x8, \( \text{modee}Y(MB) = 9 \) to 17. If any mode is I16x16, then it is modified to I8x8 modes. But Plane mode is converted to DDL. If any mode is I4x4, it is modified to I8x8. The other modes are DC modes.

iii. If the highest mode is I4x4, \( \text{modee}Y(MB) = 0 \) to 8. If any mode is I16x16, it is modified to I4x4 modes. But Plane mode is converted to DDL. If any mode is I8x8, it is modified to I4x4. The other modes are DC modes.

Intra 16x16 mode: In the first row and first column MB, the mode is DC. In the first column, if mode is H or P, it is modified to DC. In the first row, if mode is V or P, it is modified to DC. Otherwise, \( \text{modee}Y(MB) = 116 \times 16 \)

Intra 4x4 / Intra 8x8 mode:

i. The mode in first row-first col of the frame is modified to DC.

ii. If the mode in first row is V / DDL / VR / HD / VL / HU, it is modified to DC.

iii. If the mode in first column is H / DDR / VR / HD / HU, it is modified to DC.

7.5.4. Chroma mode selection and decision

The corresponding Chroma Modes are extracted for a given MB.

\[
candmodes = \text{modedC}(i, j) 
\]

where

\[
i = \text{rowstart} \ldots \text{rowend} \text{ and } j = \text{colstart} \ldots \text{colend} \\
\text{rowstart} = \lfloor \text{row} \times \text{hratio} \rfloor, \quad \text{row} = 0 \ldots 2 \times rMB - 1 \\
\text{rowend} = \lfloor (\text{row} + 1) \times \text{hratio} \rfloor - 1, \quad \text{row} = 0 \ldots 2 \times rMB - 1 \\
\text{colstart} = \lfloor \text{col} \times \text{wdratio} \rfloor, \quad \text{col} = 0 \ldots 2 \times cMB - 1 \\
\text{colend} = \lfloor (\text{col} + 1) \times \text{wdratio} \rfloor - 1 \quad \text{col} = 0 \ldots 2 \times cMB - 1 \\
\]

<table>
<thead>
<tr>
<th>( \text{DC @ (0,0)} )</th>
<th>( \text{DC @ first row} )</th>
<th>( \text{DC @ first col} )</th>
<th>( \text{DC others} )</th>
<th>( \text{H} )</th>
<th>( \text{V} )</th>
<th>( \text{P} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>360°</td>
<td>0°</td>
<td>90°</td>
<td>45°</td>
<td>0°</td>
<td>90°</td>
<td>45°</td>
</tr>
</tbody>
</table>

Table 7.4 Angles for Intra Chroma Modes

The flowchart of Intra Chroma mode decision is shown in Fig. 7.4 and explained as follows.

i. The angle for each mode in \( \text{candmodes} \) is substituted (Ref. Table 7.4).

ii. The number of DC modes and Plane modes are counted.

iii. If the number of DC modes is more than half of number candidate modes, then DC mode only is decided.
iv. Else the following is done.
   a. The average of angles, called resultant angle, ‘r’ is calculated.
   b. Then $modeeC(MB)$ is calculated as follows.
      
      $modeeC(MB) = \begin{cases} 
      1, & r < 22.5^\circ \\
      3, & 22.5^\circ \leq r < 67.5^\circ \\
      2, & 67.5^\circ \leq r < 112.5^\circ \\
      0, & 112.5^\circ \leq r 
      \end{cases}$

v. In the first column of MBs in the frame, (i.e., column = 1), if the mode is H, it is modify to DC. If the mode is P, it is modified to V.

vi. In the first row of MBs in the frame, (i.e., row = 1), if the mode is V, it is modified to DC. If the mode is P, it is modified to H.

---

**Fig. 7.4 Flowchart for Intra Chroma Mode Decision**
7.5.5.  **Conclusion of Intra Mode Decision**

In this process, the Intra modes of Luma and Chroma of each MB of resized frame are calculated. The modified syntax elements are sent to compressed domain encoder for encoding the resized frame. The reuse technique of modifying the available decoded modes is developed and coded in MATLAB.

7.6.  **Inter Block Decision**

If the modified MB Type is Inter, the MVs for each sMB and its corresponding block sizes are calculated to represent a given MB. MV refinement decides the block type and its corresponding MVs at the qual-pel MV information.

7.6.1.  **Extracting the MVs**

The MVs for each 4x4 sMB of a given MB are calculated here. The corresponding candidate MVs are extracted for a given sMB.

\[ \text{candy} = \text{mvxd}(i, j) \quad \text{and} \quad \text{cand} = \text{mvyd}(i, j) \]

where

\[ i = \text{rowstart} \ldots \text{rowend} \quad \text{and} \quad j = \text{colstart} \ldots \text{colend} \]

\[ \text{rowstart} = \lfloor \text{row} \times \text{hratio} \rfloor, \quad \text{row} = 0 \ldots 4 \times r\text{MB} - 1 \]

\[ \text{rowend} = \lfloor (\text{row} + 1) \times \text{hratio} \rfloor - 1, \quad \text{row} = 0 \ldots 4 \times r\text{MB} - 1 \]

\[ \text{colstart} = \lfloor \text{col} \times \text{wdratio} \rfloor, \quad \text{col} = 0 \ldots 4 \times c\text{MB} - 1 \]

\[ \text{colend} = \lfloor (\text{col} + 1) \times \text{wdratio} \rfloor - 1 \quad \text{col} = 0 \ldots 4 \times c\text{MB} - 1 \]

The MV for each sMB of a given MB is calculated as follows.

\[ \text{mvxe}(s\text{MB}) = \left[ \begin{array}{c} \frac{\text{average}(\text{candy})}{\text{wdratio}} \\ \frac{\text{average}(\text{candy})}{\text{hratio}} \end{array} \right] \]

\[ \text{mvye}(s\text{MB}) = \left[ \begin{array}{c} \frac{\text{average}(\text{candy})}{\text{wdratio}} \\ \frac{\text{average}(\text{candy})}{\text{hratio}} \end{array} \right] \]

7.6.2.  **Refining Macroblock type based on MVs of each Macroblock**

Steps for refining MVs and Macroblock Type of each MB are given below.

1. If \( \text{modee}Y(MB) \) is Inter, the MVs of corresponding MB are taken into separate matrices for analysis, as \( \text{maty} = \text{mvye}(MB) \) and \( \text{matx} = \text{mvxe}(MB) \).
2. If MB Type is PSKIP or the QP > 27, the MVP under PSKIP (mvps) is calculated as per H.264 Standard. Then MB Type is declared as P16x16, i.e., mbtypemat(MB) = 0 and modeeY(MB) is declared as PSKIP.

3. If the MB Type is P16x16, mbtypemat(MB) = 0 and the following is done.
   a. Single MV for MB is found by averaging all the MVs, i.e., fmvy and fmxv.
   b. If the QP > 27 and |fmv - mvps| ≤ thr_skip, modeeY(MB) = PSKIP. Else modeeY(MB) = P16 × 16 is kept. Table 7.5 is referred for thr_skip.

<table>
<thead>
<tr>
<th>Table 7.5 Threshold values for MV deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPYe</td>
</tr>
<tr>
<td>thr_mv</td>
</tr>
<tr>
<td>thr_P16x16</td>
</tr>
<tr>
<td>thr_SKIP</td>
</tr>
<tr>
<td>QPYe</td>
</tr>
<tr>
<td>thr_mv</td>
</tr>
<tr>
<td>thr_P16x16</td>
</tr>
<tr>
<td>thr_SKIP</td>
</tr>
</tbody>
</table>

4. If the MB Type is P16x8, mbtypemat(MB) = 1 and the following steps are done.
   a. The MVs for TOP P16x8 are calculated by averaging the top 4x2 MVs of maty and matx.
   b. The MVs for BOTTOM P16x8 are calculated by averaging the bottom 4x2 MVs

5. If the MB Type is P8x16, mbtypemat(MB) = 2 and the following steps are done.
   a. The MVs for LEFT P8x16 are calculated by averaging the left 2x4 MVs.
   b. The MVs for RIGHT P8x16 are calculated by averaging the right 2x4 MVs.

6. If the MB Type is P8x8, mbtypemat(MB) = 4 is substituted.
   a. For each sMB, the following process is done.
      i. The MVs are taken into ty and tx from maty and matx.
         \[
         ty = \text{maty}(sMB) \\
         tx = \text{matx}(sMB)
         \]
         (7-10)
      ii. If the sMB Type is P8x8, the average of ty and tx are found.
      iii. If the sMB Type is P8x4, then the following is done using ty and tx.
         1. The MVs for TOP P8x4 are found by averaging the top 2x1 MVs.
         2. The MVs for BOTTOM P8x4 are found by averaging bottom 2x1 MVs.
      iv. If the sMB Type is P4x8, then the following is done.
         1. The MVs for LEFT P4x8 are found by averaging the left 1x2 MVs.
2. The MVs for RIGHT P4x8 are found by averaging the right 1x2 MVs.
   v. If the sMB Type is P4x4, then the MVs are kept as same.

b. If the QP is greater than 27, the following motion refinement is implemented.
   i. If sMB type is P4x4, the closeness of MVs is analysed. If they are closer, then they are combined to create bigger block sizes.
   ii. If absolute difference between top $MV_s \leq thr_{mv}$ (Ref. Table 7.5) and absolute difference between bottom $MV_s \leq thr_{mv}$, then sMB Type is changed to P8x4. The top MVs are averaged and stored in top. The bottom MVs are averaged and stored in bottom. If absolute difference between top and bottom $MV_s \leq thr_{mv}$, then sMB Type is changed to P8x8. The top and bottom MVs are averaged and stored as single MV for the sMB.
   iii. If absolute difference between left $MV_s \leq thr_{mv}$ and absolute difference between right $MV_s \leq thr_{mv}$, then sMB Type is changed to P4x8. The left MVs are averaged and stored in left and the right MVs are averaged and stored in right. If absolute difference between left and right $MV_s \leq thr_{mv}$, then sMB Type is changed to P8x8. The left and right MVs were averaged and stored as single MV for the sMB.
   iv. The ty and tx values were stored to $maty$ and $matx$.

7. If QP is greater than 27 and all the sMB Types in a MB are P8x8, then motion refinement is done as follows.
   a. If absolute difference between the MVs of TOPLEFT 8x8 and TOPRIGHT 8x8 $\leq thr_{mv}$ and absolute difference between the MVs of BOTTOMLEFT 8x8 and BOTTOMRIGHT 8x8 $\leq thr_{mv}$, MB type is modified to P16x8 and the top and bottom P16x8 MVs are calculated by averaging. If absolute difference of top and bottom MVs $\leq thr_{P16x16}$, MB type is modified to P16x16 and the MV is calculated by averaging.
   b. If absolute difference between the MVs of TOPLEFT 8x8 and BOTTOMLEFT 8x8 $\leq thr_{mv}$ and absolute difference between the MVs of TOPRIGHT 8x8 and BOTTOMRIGHT 8x8 $\leq thr_{mv}$, MB type is modified to P8x16, the left and right P8x16 MVs are calculated by averaging. If absolute difference of left and right MVs $\leq thr_{P16x16}$, MB type is modified to P16x16, the MV is calculated by averaging.
8. If QP is greater than 27 and MB Type is P8x16, if absolute difference of left and right MVs ≤ \( \text{thr}_{-16\times16} \), MB type is modified to P16x16 and the MV is calculated by averaging. The MVs are updated as \( \text{mvye}(MB) = \text{maty} \), \( \text{mvxe}(MB) = \text{matx} \).

9. If QP is greater than 27 and MB Type is P16x8 and if absolute difference of top and bottom MVs ≤ \( \text{thr}_{16\times16} \), then MB type is modified to P16x16 and the MV is calculated by averaging. The MVs are updated as \( \text{mvye}(MB) = \text{maty} \), \( \text{mvxe}(MB) = \text{matx} \).

10. If QP is greater than 27 and MB Type is P16x16 and absolute difference of P16x16 MV and SKIP MVs ≤ \( \text{thr}_{SKIP} \), then \( \text{modeeY}(MB) = \text{PSKIP} \) is substituted. Otherwise, \( \text{modeeY}(MB) = P16 \times 16 \) is kept.

7.6.3. Conclusion of Inter Block Decision

In this process, the MVs and block sizes of each MB of resized Frame are calculated. The modified syntax elements are sent to compressed domain encoder for encoding the resized frame. The reuse technique of modifying the available decoded modes was developed and coded in MATLAB. The output of reuse engine (MATLAB code) was verified and ensured that it always adhered to H.264 Standard.

7.7. Conclusion of Reuse Engine

The decoded syntax elements pertaining to different types of MBs are extracted. They are carefully modified to suit the encoding processing in compliance with H.264 Standard. The modified syntax elements are sent to compressed domain encoder. The reuse engine is developed in MATLAB which has extracted syntax elements from compressed domain decoder, modified the syntax elements and supplied the modified syntax elements to compressed domain encoder. It is ensured that all the syntax elements are adhered to H.264 Standard.