HEVC Entropy Coding

한밭대학교 정보통신·컴퓨터공학부
최 해철
1. Introduction

2. HEVC CAVLC
   - LCEC: Low-complexity entropy coding

3. HEVC CABAC (PIPE)
   - PIPE(probability interval partitioning entropy)

4. Performance comparison between CAVLC and CABAC

5. Summary
Entropy coding?

Assigning codes to symbols so as to match code lengths with the probabilities of the symbols
Information source

- Discrete memoryless source: statistically independent events
- Source Alphabet: \( S = \{s_1, s_2, \ldots, s_n\} \)
- Probabilities of occurrence: \( P(S) = \{p(s_1), p(s_2), \ldots, p(s_n)\} \)

Self Information

- Less probable event \( \rightarrow \) more information
- Amount of information
  \[
  \text{amount of information} \propto \frac{1}{\text{probability of information}}
  \]
- Information of independent events = sum of each information
- Measure of Self information
  \[
  I(s_i) = \log \left( \frac{1}{p(s_i)} \right) = - \log p(s_i)
  \]
Entropy

- Average amount of information per source symbol
- Entropy:
  \[ H(S) = \sum_s P(s) I(s) = - \sum_s P(s) \log_2 P(s) \text{ bits/symbol} \]
- Measure of information or uncertainty

Best lossless compression rate

- For a given information source, the best lossless compression rate that we can achieve is the source entropy.
  \[ H(S) \leq R < H(S) + 1 \]
- Shannon’s first coding theorem (noiseless coding theorem)
A principle

- Short codewords for frequently occurring values
- Longer codewords for less common parameter values

Table 7.13 - Macroblock type values 0 to 4 for P and SP slices

<table>
<thead>
<tr>
<th>mb_type</th>
<th>Name of mb_type</th>
<th>NumMBPart (mb_type)</th>
<th>MBPartPredMode (mb_type, 0)</th>
<th>MBPartPredMode (mb_type, 1)</th>
<th>MBPartWidth (mb_type)</th>
<th>MBPartHeight (mb_type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>P_I.0_16x16</td>
<td>1</td>
<td>Pred_I.0</td>
<td>na</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>P_I0.I0_16x8</td>
<td>2</td>
<td>Pred_I0</td>
<td>Pred_I0</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>P_I0.I0_8x16</td>
<td>2</td>
<td>Pred_I0</td>
<td>Pred_I0</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>P_8x8</td>
<td>4</td>
<td>na</td>
<td>na</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>P_8x8Ref0</td>
<td>4</td>
<td>na</td>
<td>na</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>inferred</td>
<td>P_Skip</td>
<td>1</td>
<td>Pred_L0</td>
<td>na</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>
Entropy Coding of H.264|AVC

- Entropy coding tools
  - Fixed- or variable-length binary codes
  - UVLC: Exponential Golomb codes, CAVLC
  - CABAC

Ref. 2009년 MPEG 표준기술 심화교육, 심동규(광운대)
Entropy Coding of H.264|AVC

- **Context-adaptive?**
  - Adapting the probability of a symbol based on (neighbor) context

- **CAVLC (context-adaptive variable-length coding)**
  - Used for transform coefficients
  - Multiple VLC tables for a single syntax are used and the selection depends on context
  - It is supported in all H.264/AVC profiles
    - entropy_coding_mode_flag (in PPS) is equal to 0

- **CABAC (context-adaptive binary arithmetic coding)**
  - Used for all syntax elements except ones on sequence, picture, and slice header level
  - 4 context model types are used individually or jointly
  - Enormous context models are used and probabilities are updated depending on context
  - Both of coding efficiency and complexity is higher than CAVLC
  - It is only supported in main and high profiles
    - entropy_coding_mode_flag is equal to 1
CAVLC of H.264|AVC

Assume that table Num-VLC0 is used to encode coeff_token

1. **coeff_token**
   - TotalCoeff = 5, T1s = 3
   - coeff_token = 0000100

2. **T1 sign**
   - T1(5), T1(4), T1(3)
   - T1 sign = 011

3. **Level**
   - Level(2): +1 (using Level_VLC0)
   - Level(1): +3 (using Level_VLC1)
   - Level = 10010

4. **TotalZeros**
   - TotalZeros = 3
   - TotalZeros = 111

5. **run_before**
   - (5) ZeroLeft=3, run_before=1
   - (4) ZeroLeft=2, run_before=0
   - (3) ZeroLeft=2, run_before=0
   - (2) ZeroLeft=2, run_before=1
   - (1) ZeroLeft=1, run_before=1
   - run_before = no coded

Ref. 2009년 MPEG 표준기술 심화교육, 심동규(광운대)
CAVLC of H.264|AVC

- **CAVLC**
  - **coeff_token**
    - The total number of non-zero coefficient (TotalCoeff) and the number of trailing +/- 1 (T1)
    - 4 tables for Luma and chroma AC + 2 tables for chroma DC
    - The selection of table depends on neighboring syntax elements
      - nC: the average number of non-zero coefficients in upper (nB) and left blocks (nA)

![Table 9-5 – coeff_token mapping to TotalCoeff(coeff_token) and TrailingOnes(coeff_token)](image)
CAVLC of H.264|AVC

**CAVLC**

- **Level**
  - There are 7 VLC tables (Level_VLC0 ~ Level_VLC6)
  - VLC table is selected depending on the magnitude of each successive coded level

- **total_zeros**
  - VLC table is selected depending on TotalCoeff

Table 9-7 – total_zeros tables for 4x4 blocks with TotalCoeff(coeff_token) 1 to 7

<table>
<thead>
<tr>
<th>total_zeros</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>111</td>
<td>0101</td>
<td>0001</td>
<td>1</td>
<td>0101</td>
<td>000001</td>
<td>000001</td>
</tr>
<tr>
<td>1</td>
<td>011</td>
<td>110</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>1100</td>
<td>00001</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>101</td>
<td>110</td>
<td>0101</td>
<td>0011</td>
<td>111</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>100</td>
<td>101</td>
<td>0100</td>
<td>111</td>
<td>110</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>0010</td>
<td>011</td>
<td>0100</td>
<td>110</td>
<td>110</td>
<td>101</td>
<td>011</td>
</tr>
</tbody>
</table>
The design of CABAC

Binarization, Context Modeling, Binary Arithmetic Coding.

mb_type = 5
(I_16x16_0_1_0)
Binaryzation

- Process of converting a data symbol into a variable length code since CABAC uses binary arithmetic coding which means that only binary decisions (1 or 0) are encoded.
- A “binarized” value is named bin in H.264/AVC
Context modeling

- Use of conditional probabilities of symbols
- 460 context models for various syntax elements
  - Probability state index (pStateIdx), binary value of MPS (valMPS)
- Adaptively select one context model for each input bin (1 or 0)
- Adapt probability estimates based on local statistics
Context modeling

- Neighboring syntax elements
  - mb_skip_flag, mb_field_decoding_flag, mvd, cbp, ref_idx, transform_size and etc

- The values of prior coded bins
  - mb_type and sub_mb_type

- The position in the scanning path
  - Residual data (significant_coeff_flag, ...)

- The accumulated number of encoded levels with a specific value prior to the current level bin
  - Residual data (abs_level_minus_1)
HEVC Entropy Coding

- **CAVLC**
  - Entropy coding for low-complexity configuration
  - LCEC (Low complexity entropy coding)
    - JCTVC-A119 (Tandberg, Ericsson, and Nokia)
    - Syntax and mapping table
    - Transform coefficients coding
    - Adaptive table update
    - Used for other syntax besides transform coefficients

- **CABAC**
  - Entropy coding for high-efficiency configuration
  - Similar to H.264/AVC CABAC
  - PIPE(probability interval partitioning entropy)
    - Parallel context modeling

- Mode dependent Coefficient Scanning
- **Mode Dependent Coefficient Scanning (MDCS)**
  - JCTVC-D393 [Qualcomm]
  - Intra
  - 3 scans
    - 0: zigzag
    - 1: horizontal
    - 2: vertical

- 2 mapping tables
  - TU size
  - Intra prediction mode

| Mode TU | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 32x32   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 16x16   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 8x8     | 1  | 2  | 0  | 1  | 0  | 2  | 2  | 0  | 1  | 1  | 0  | 2  | 2  | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 2  | 2  | 2  | 2  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 4x4     | 1  | 2  | 0  | 1  | 1  | 2  | 2  | 0  | 1  | 1  | 0  | 2  | 2  | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 2  | 2  | 2  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 2x2     | 1  | 2  | 0  | 1  | 1  | 2  | 2  | 0  | 1  | 1  | 0  | 2  | 2  | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 2  | 2  | 2  | 2  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
HEVC CAVLC Overview

- Additional step for code-word generation
  - Symbol → code number (codeNum) → codeword

- One more context adaptive mechanism
  - Multiple VLC tables
  - Sorting table

Ref. 2011년 7월 MPEG 보고회, 이상윤
Adaptive Mapping Table Update

Sorting table

H.264/AVC

event
e.g. inter/
16x16/
ref=0
e.g. 7
e.g. 010
code number
inverse VLC table

HEVC

event
e.g. inter/
16x16/
ref=0
e.g. 12
table index
table
inverse sorting table
inverse enumeration
e.g. 7
e.g. 010
code number
inverse VLC table
Adaptive Mapping Table Update

Example

<table>
<thead>
<tr>
<th>구분 정보</th>
<th>발생 빈도 순서</th>
<th>VLC 테이블</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Table index</td>
<td>Code number</td>
</tr>
<tr>
<td>Split</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Skip</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Merge</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Direct_L0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Direct_L1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Direct_Bi</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Others</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Ref. 심동규, 유은경, HEVC 엔트로피 부호화 기술 동향, 대한전자공학회지, 2011
Coefficient Coding

Transform coefficients coding structure

- Scanning
  - 1D array using the conventional zig-zag scan

- Last position
  - Position, level, and sign of the last non-zero coefficient
  - Different scheme 4x4 or larger TU
  - 4x4 using sorting table, others using equations

- Run mode
  - Run and level
    - Inter(chroma) and intra separately
    - Table based equation
      - Intra: Feature table based on coded $trOnes$
      - Inter: Table and equation

- Level mode
  - Each coefficient
    - Similar to H.264/AVC level coding

Transition
- TU size
- Position of the current transform coefficient
- Sum of non-zero transform coefficients

ex) 4x4 block

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Ref. 2011년 7월 MPEG 보고회, 이상윤
## Coefficient Coding

### Syntax and semantics for coefficient coding (1/3)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Syntax</th>
<th>Semantics</th>
<th>Variables</th>
</tr>
</thead>
</table>
| last position| last_pos_level_one   | the position of the last non-zero valued transform coefficient and whether the absolute value of the transform coefficient is larger than 1 | \[ n = ( 1 \ll ( \log_2 \text{TrafoSize} \ll 1 ) ) \gg ( \text{cIdx} > 0 ? 2 : 0 ) \]  \[
\text{lastPos} = \text{last_pos_level_one} \% n
\text{levelGreaterThanOneFlag} = ( \text{last_pos_level_one} > n )
\] |
| run mode     | run_level_one        | the number of consecutive transform coefficients in the scan with zero value before a non-zero valued transform coefficients and whether the absolute value of the non-zero valued transform coefficient is larger than 1 | \[ \text{runOfZeros} = \text{run_level_one} \% n \]  \[
\text{levelGreaterThanOneFlag} = ( \text{run_level_one} > n )
\] |
## Syntax and semantics for coefficient coding (2/3)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Syntax</th>
<th>Semantics</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>last position and</td>
<td>level_minus2_and_sign</td>
<td>When the value of a transform coefficient that has an absolute value &gt; 1</td>
<td>level = ((level_minus2_and_sign &gt;&gt; 1) + 2</td>
</tr>
<tr>
<td>run mode</td>
<td></td>
<td></td>
<td>sign = 1 - 2*(level_minus2_and_sign &amp; 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transCoeffLevel[...] = level * sign</td>
</tr>
<tr>
<td>sign_flag</td>
<td></td>
<td>When level == 1, the sign of a non-zero valued transform coefficient</td>
<td>sign = 1 – 2 * sign_flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transCoeffLevel[...] = 1 – 2 * (sign_flag)</td>
</tr>
</tbody>
</table>
## Coefficient Coding

### Syntax and semantics for coefficient coding (3/3)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Syntax</th>
<th>Semantics</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level mode</td>
<td><strong>level</strong></td>
<td>the absolute value of a non-zero valued transform coefficient</td>
<td><strong>level</strong></td>
</tr>
<tr>
<td></td>
<td><strong>sign_flag</strong></td>
<td>When level &gt; 0,</td>
<td><em><em>sign = 1 - 2</em>(level_minus2_and_sign &amp; 1)</em>*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>transCoeffLevel[...] = level * sign</strong></td>
</tr>
</tbody>
</table>
CAVLC for Other Syntax

- **H.264/AVC**
  - CAVLC is used for transform coefficients

- **HEVC**
  - CAVLC is used for
    - Transform coefficients
    - Intra prediction mode
      - `rem_intra_luma_pred_mode[x0][y0]` | `ce(v) | ae(v)`
    - CU Split, SKIP, Prediction mode, and partition mode
      - `if(!entropy_coding_mode_flag && slice_type != I)`
      - `cu_split_pred_part_mode[x0][y0]` | `ce(v)`
      - ...

...
**CABAC**

- **PIPE (probability interval partitioning entropy)**
  - Binarization and context modelling as in CABAC of H.264/AVC
  - Modified coding of binary decisions (bins)
    - LPB probabilities are quantized (12 classes in implementation)
    - Separate bin encoders for each class (fixed LPB probabilities)
    - Supports *high degree of parallelization*
    - Supports *variable length codes* without compromising coding efficiency
Entropy Coder comparison

Coding efficiency

- HE configuration + CAVLC: 6.5~7.6% performance degradation
- LC configuration + CABAC: -6.1~7.0% performance gain
- Performance gap decrease in RDOQ off conditions or CU level parallel processing(tiles)
- CAVLC shows better performance in chroma channel coding

<table>
<thead>
<tr>
<th>BD-rate (%)</th>
<th>High efficiency</th>
<th>Low complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AI</td>
<td>RA</td>
</tr>
<tr>
<td>Class A</td>
<td>6.3</td>
<td>9.3</td>
</tr>
<tr>
<td>Class B</td>
<td>7.5</td>
<td>9.0</td>
</tr>
<tr>
<td>Class C</td>
<td>6.0</td>
<td>6.4</td>
</tr>
<tr>
<td>Class D</td>
<td>5.3</td>
<td>5.4</td>
</tr>
<tr>
<td>Class E</td>
<td>9.1</td>
<td>7.1</td>
</tr>
<tr>
<td>Average</td>
<td>6.7</td>
<td>7.6</td>
</tr>
</tbody>
</table>
Entropy Coder comparison

- HE configuration + CAVLC: 1~9% faster (encoding)
- LC configuration + CABAC: 10~18% slower (encoding)

<table>
<thead>
<tr>
<th></th>
<th>HE</th>
<th>LC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AI  RA  LB  LP</td>
<td>AI  RA  LB  LP</td>
</tr>
<tr>
<td>Enc. Time (%)</td>
<td>91%  99%  99%  95%</td>
<td>118%  110%  112%  114%</td>
</tr>
<tr>
<td>Dec. time (%)</td>
<td>94%  97%  99%  98%</td>
<td>109%  104%  102%  101%</td>
</tr>
</tbody>
</table>
References

- CAVLC
  - LCEC (JCTVC-A119, Tandberg, Ericsson, and Nokia)
  - Coefficient sign PCP (JCTVC-B088 Section 3.2)
  - Coefficeint level BinIdx 0 PCP (JCTVC-B088 Section 3.3)
  - Coded block flag signaling in VLC (JCTVC-C262)
  - Coded block flag redundancy removal (JCTVC-C277)
  - Intra mode coding (JCTVC-D366)
  - Coefficient coding (JCTVC-D374)
  - Reference index coding (JCTVC-D141/D184)
  - Inter mode signaling (JCTVC-D370)

- CABAC
  - JCTVC-A116 (HHI) and JCTVC-A120 (RIM)
  - Simplified context generation for significance map (JCTVC-D260)

- Mode-dependent coefficient scanning (JCTVC-D393)
CAVLC

- LCEC (Low complexity entropy coding)
  - adaptive mapping table update
  - transform coefficients coding (last position, run mode, level mode)
  - syntax and semantics

CABAC

- Similar to H.264/AVC CABAC
- PIPE (probability interval partitioning entropy)

Mode dependent Coefficient Scanning

Performance comparison between CAVLC and CABAC
THANK YOU