Analysis of Motion Estimation Algorithms in HEVC

Multimedia Processing EE5359
Spring 2014
Final Report

Update: 4/27/2014

Advisor: Dr. K. R. Rao
Department of Electrical Engineering
University of Texas, Arlington

Tuan Ho
1001006444
tuanpho@mavs.uta.edu
Outline

- Scope of this work
- HEVC Overview
- Motion Estimation
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
Outline

- Scope of this work
- HEVC Overview
- Motion Estimation
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
Scope

- Analysis the computational complexity in HEVC motion estimation (ME) and state-of-the-art approaches.
- Implement the approaches into HEVC Model.
- Quality assertion methods:
  - PSNR [20]
  - Bjontegaard BD-PSNR [22]
Outline

- Scope of this work
- HEVC Overview
- Motion Estimation
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
HEVC Overview

- HEVC is the current joint video coding standardization project of the ITU-T and ISO/IEC MPEG [1].
- Aimed at doubling the compression ratio of H.264 at the same level of visual quality [1].
- Exceed H.264 in terms of performance and computational complexity.
- Lot of tools and features have been brought forth into HEVC: quadtree partion, SAO, etc.
HEVC Overview

HEVC Encoder Block Diagram (Decoder components modeling in light gray) [1]
HEVC Quadtree Partition
Outline

- Scope of this work
- HEVC Overview
- **Motion Estimation (ME)**
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
ME Overview

Motion Estimation Process Illustration [3]
ME Computational Complexity

- Inter prediction can take up to 84% of the entire encoding time [17].

HEVC encoding time portions. These execution time results were obtained by profiling the encoding of the “BasketballDrive” Full HD (1920×1080) sequence. [17]
ME Computational Complexity

- Largest block size in HEVC is 64x64, 4 times bigger than its of H.264.
- This significantly raises the ME computational complexity.
Outline

- Scope of this work
- HEVC Overview
- Motion Estimation
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
ME Challenges

- Many partition sizes in the quadtree HEVC 8x8 to 64x64

Quadtree Partition [1]
ME Challenges

- HEVC ME for large-sized partition (i.e. 64x64) requires heavy computation exceeding its predecessor. In H.264 largest block size is 16x16. [Ref]

  - Big interpolation logic.
  - Time-consuming SAD process.
  - Large memory storage.

HEVC ME for 64x64 PU
ME Challenges

- HEVC Interpolation on large-sized partitions

HEVC Interpolation Filter [1]
Outline

- Scope of this work
- HEVC Overview
- Motion Estimation
- Challenges for Motion Estimation in HEVC
- Approaches
- Implementation and Results
Fast search algorithms

Employ optimized fast search algorithms.

*Fast search algorithms - (a) Two dimensional logarithmic (2DLOG), (b) Three steps search (TSS), (c) New three steps search (NTSS)* [6]
Fast search algorithms

Employ optimized fast search algorithms.

*Fast search algorithms - (a) Four steps search (FSS), (b) One-at-a-time search (OTA), (c) Orthogonal search algorithm (OSA) [6]*
Fast search algorithms

Diammon and Hexagonal Searches
To improve the motion estimation performance, HEVC employs a combination of zone search algorithms called TZ Search whose implementation was introduced in the JMVC software [25].

The TZ search algorithm combines diamond search and raster search methods together to produce a superior performance comparing to the full search.

TZS flow-chart [24]
Approaches

Hierarchical Search

Hierarchical Search [6]
Approaches

Complexity Control Employment

Complexity of processing regarding to CU Depth [15]
Experiment

Simulation is executed on:

- **HEVC Model (HM) version:** [13.0rc1][Windows][VS 1800][32 bit]. Built in Release version.
- **Encoding configuration:** random access main.
- **PC specification:** Intel i7 (8 CPUs) @ 2.4GHz, 8GB memory running on Microsoft Windows 7.
- **Test sequences and settings are shown in Table 1.**
  - Test sequences are running with four values of quantization parameter of 22, 27, 32, 37.
  - Image size from full HD to small size.
  - Sequences involve lots of activities for motion estimation test.
### Coding Structure

<table>
<thead>
<tr>
<th>#</th>
<th>Type</th>
<th>POC</th>
<th>QPoffset</th>
<th>QPfactor</th>
<th>tcOffsetDiv2</th>
<th>betaOffsetDiv2</th>
<th>temporal_id</th>
<th>ref_pics_active</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>Frame1:</td>
<td>B</td>
<td>8</td>
<td>1</td>
<td>0.442</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>Frame2:</td>
<td>B</td>
<td>4</td>
<td>2</td>
<td>0.3536</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>Frame3:</td>
<td>B</td>
<td>2</td>
<td>3</td>
<td>0.3536</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>41</td>
<td>Frame4:</td>
<td>B</td>
<td>1</td>
<td>4</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>42</td>
<td>Frame5:</td>
<td>B</td>
<td>3</td>
<td>4</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>Frame6:</td>
<td>B</td>
<td>6</td>
<td>3</td>
<td>0.3536</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>Frame7:</td>
<td>B</td>
<td>5</td>
<td>4</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>Frame8:</td>
<td>B</td>
<td>7</td>
<td>4</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

### Motion Search

- **FastSearch**: 1
- **SearchRange**: 64
- **BipredSearchRange**: 4
- **HadamardME**: 1
- **FEN**: 1
- **FDM**: 1

### Quantization

- **QP**: 32
- **MaxDeltaQP**: 0
- **MaxCuQDPDepth**: 0
- **DeltaQpRD**: 0
- **RDOQ**: 1
- **RDOQTS**: 1

---

Random access main configuration incomplete snapshot
## Test sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Resolution</th>
<th>FPS</th>
<th>Frame Count (each QP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kimono</td>
<td>22,27,32,37</td>
<td>1920x1080</td>
<td>24</td>
<td>100</td>
</tr>
<tr>
<td>BQMall</td>
<td>22,27,32,37</td>
<td>832x480</td>
<td>60</td>
<td>200</td>
</tr>
<tr>
<td>BasketballPass (TZ search)</td>
<td>22,27,32,37</td>
<td>416x240</td>
<td>50</td>
<td>200</td>
</tr>
<tr>
<td>BasketballPass (full search)</td>
<td>22,27,32,37</td>
<td>416x240</td>
<td>50</td>
<td>200</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>2800</strong></td>
</tr>
</tbody>
</table>
Test sequences

BQMall
Test sequences

BasketBall Pass
Test sequences

Kimono
# Simulation Results

<table>
<thead>
<tr>
<th>Run at</th>
<th>25-Apr-14 (2nd experiment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td></td>
</tr>
<tr>
<td>Seq info</td>
<td></td>
</tr>
<tr>
<td>YUV size for frame count (original) (Mbyte)</td>
<td>711 for 240 frames</td>
</tr>
<tr>
<td>YUV size for frame count (experiment) (Mbyte)</td>
<td>296.3 for 100 frames</td>
</tr>
<tr>
<td>Enc info</td>
<td></td>
</tr>
<tr>
<td>TZ Search</td>
<td></td>
</tr>
<tr>
<td>qp</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>22</td>
<td>42.45</td>
</tr>
<tr>
<td>27</td>
<td>40.61</td>
</tr>
<tr>
<td>32</td>
<td>38.47</td>
</tr>
<tr>
<td>37</td>
<td>36.43</td>
</tr>
</tbody>
</table>
## Simulation Results

<table>
<thead>
<tr>
<th></th>
<th>BasketballPass</th>
<th>BasketballPass (FullSearch)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>416x240, 50fps, 4:2:0, 8bit/bpp</td>
<td>416x240, 50fps, 4:2:0, 8bpp</td>
</tr>
<tr>
<td></td>
<td>72 for 500 frames</td>
<td>71.5 for 500 frames</td>
</tr>
<tr>
<td></td>
<td>28.8 for 200 frames</td>
<td>28.6 for 200 frames</td>
</tr>
<tr>
<td><strong>TZ Search</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Full Search</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>41.68</td>
<td>41.68</td>
</tr>
<tr>
<td>Bitstream Size (MByte)</td>
<td>0.60</td>
<td>0.60</td>
</tr>
<tr>
<td>Bitrate (kbps)</td>
<td>1,263.35</td>
<td>1,261.69</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>779.79</td>
<td>14,085.35</td>
</tr>
<tr>
<td>Comp Ratio</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>38.24</td>
<td>38.26</td>
</tr>
<tr>
<td>Bitstream Size (MByte)</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>Bitrate (kbps)</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>686</td>
<td>13,840</td>
</tr>
<tr>
<td>Comp Ratio</td>
<td>96</td>
<td>95</td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>35.22</td>
<td>35.24</td>
</tr>
<tr>
<td>Bitstream Size (MByte)</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Bitrate (kbps)</td>
<td>312.38</td>
<td>311.62</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>607.56</td>
<td>13,527.15</td>
</tr>
<tr>
<td>Comp Ratio</td>
<td>193</td>
<td>192</td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>32.76</td>
<td>32.78</td>
</tr>
<tr>
<td>Bitstream Size (MByte)</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>Bitrate (kbps)</td>
<td>162</td>
<td>162</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>544</td>
<td>13,214</td>
</tr>
<tr>
<td>Comp Ratio</td>
<td>373</td>
<td>370</td>
</tr>
</tbody>
</table>
Discussion

Bitrate reduction corresponds to QP
Discussion

• Bitrate reduces by half when qp increase by a step of 5 (e.g. 22 to 27, 27 to 32, 32 to 37).

• For easier to grasp the information, the graph is re-drawn in log2 scale. Each unit change in log2(bitrate) corresponds to the half increase/decrease in normal scale.
Discussion

Bitrate (in log2) reduces by half for each 5 qp step increases
Discussion
Discussion
Discussion

![Graph showing PSNR (dB) vs Bitrate (kbps)](image_url)

- **BasketballPass (TZ Search)**
- **BasketballPass (Full Search)**
Discussion

TZ search has the same performance as full search with significant reduction in encoding time (more than ten times faster)
CU partition
Motion Vectors

BasketballPass: Frame 2 (B)
Future work

• Even TZ search is much faster than the exhaustive full search, it can be improved by applying the early termination method to reduce the encoding time.
References


References


References


References


References


References


References


Acronyms

- SAO: Sample Adaptive Offset
- 2DLOG: Two dimensional logarithmic
- CTU: Coding Tree Unit
- CU: Coding Unit
- FS: Full Search
- FSS: Four steps search
- HEVC: High Efficiency Video Coding
- HD: High Definition
- JCT-VC: Joint Collaborative Team on Video Coding
- PU: Prediction Unit
- ME: Motion Estimation
- MPEG: Moving Picture Experts Group
- TZS: a new group search employed in HEVC HM 13
Acronyms

- NTSS: New three steps search
- OSA: Orthogonal search algorithm
- OTA: One-at-a-time search
- SAO: Sample Adaptive Offset
- T.B.D: To Be Determined
- TSS: Three steps search
- VCEG: Video Coding Experts Group
Thank you!
YUV size manually calculated based on frame size and format 4:2:0, 8bpp

<table>
<thead>
<tr>
<th>Kimono (Mbyte)</th>
<th>BQMall (Mbyte)</th>
<th>BaskBPass (Mbyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 1080</td>
<td>832 480</td>
<td>416 240</td>
</tr>
<tr>
<td>711.9140625</td>
<td>342.773438</td>
<td>71.4111328</td>
</tr>
</tbody>
</table>