Complexity reduction of inter-layer inter-prediction in Scalable extension of HEVC

Karuna Gubbi Shivashankar Sastri
Department of Electrical Engineering
University of Texas at Arlington
Advisor: Dr. K. R. Rao
Outline

• Introduction to Video
• Basics of Video Compression
• Need for Scalable Video Coding
• Introduction to SHVC
• Introduction to HEVC
• Coding structure in HEVC
• Inter layer Prediction in SHVC
• Proposed algorithm for inter layer prediction
• Experimental Conditions and Results
• Conclusions
• Acronyms
• References
What is a Video Sequences [1]

- Video is sequence of image frames shown at certain frame rate.

- Video sequence has properties such as:
  - Resolution
  - Frame Rate
  - Pixel value
  - Pixel format
Data within a picture [1]

- **Y Cb Cr format**
  - \( Y \) – mean luminance
  - \( Cb = B - Y \)
  - \( Cr = R - Y \)
  - \( Cg = G - Y \)

- **Each channel requires 8 bits for its representation**
  hence an image of resolution 1980x1080 will require 51321600 bits !!
Too much data - So how to compress? [1]

- Discard what cannot be seen
  - Pixel Representation
    - Eye perceives Intensity better than Color

- Discard Redundancies
  - Spatial
  - Temporal
New age display devices [33]

- Usage of variety of display devices.

Mobile devices

SD (1280x720) and HD (1920x1080)

UHD

iPad
How to cater? [34]

• Same content @ different resolutions

• Limitations -
  Storage expense, Computational expense, Bandwidth expense
Alternative? – Scalability [34]

- Single stream @ different resolutions !!

- Benefits -
  - Bandwidth savings
  - Computation savings
  - Storage savings
Scalability [14]

- A video bit stream is called scalable when parts of the stream can be removed in a way that the resulting sub-stream still forms another valid bit stream for some target decoder.

- Usual modes of scalability:
  1) Spatial scalability
  2) Quality/SNR/Fidelity Scalability
  3) Temporal Scalability
Scalable video coding Applications

• Streaming: different resolution and quality video sequences can be streamed with lesser usage of bandwidth

• Video Conferencing (real time): Encoding is carried out only once to meet the needs of all the end users.

• Video surveillance: Surveillance can be carried out using different resolution devices with limited usage of encoding resources.

• Broadcast: videos can be broadcasted to wide variety of end users.

• Storage: videos can be stored with lesser usage of memory space

• Base layer is encoded using usual HEVC encoder.
HEVC Encoder[2]
Introduction to HEVC [2]

• HEVC is latest video standard introduced by Joint Collaborative Team on Video Coding (JCT-VC) in January, 2013 [2][3].

• Addresses all existing applications of H.264/MPEG-4 [4] particularly focuses on two key issues:
  - Increased video resolution
  - Support for parallel processing architectures

• Achieve a compression gain of 50% over H.264/AVC at the same visual quality. [2]
Coding structure in HEVC[2][34]

- Quad tree Structure for CTU
Coding structure in HEVC [33]
Coding structure in HEVC [2]

- Prediction Unit and Transform Unit
  - CU has an associated partitioning into prediction units (PUs) and transform units (TUs).
Coding structure in HEVC

• Block partition for HEVC in a frame
Inter-Prediction in HEVC [2]

• For each PB, a motion vector will be transmitted which indicates the displacement of the block.

Motion compensation of the current picture using multiple reference pictures.
Encoding layers in SHVC[13]
Inter-layer prediction in SHVC [13]

- Step1: Performs CU size partitioning in EL
- Step2: Decomposes the CU into 8x8 sub-blocks.
- Step3: Finds a co-located block in the BL for this sub-block
- Step4: Extracts the motion information from the BL sub-block to the EL block
- Step5: Combines blocks with same motion information to form larger PU blocks as described ahead
- If 4-8x8 neighbors have same motion info combine to form a 16x16 block

- Next, if 4-16x16 blocks have same motion info combine again to form 32x32 block

- Same goes for a 4-32x32 blocks to form a 64x64 block.

- Step 6: Uses Motion Compensation to form a Prediction Image from the reference image in the EL

- Step 7: Forms the final residual image, encodes it
Limitations and Motivation

• Involves the calculations of CU partitioning

• Involves decomposition of CU into smaller sub-blocks

• Involves finding of co-located blocks in the BL

• Involves looping mechanism to Join the sub-blocks to form a larger PU
Proposed Algorithm

- Step 1: Scan the BL in left to right fashion starting at top left for PUs
- Step 2: Extract the Size and Motion Information
- Step 3: Scale the PU size and Motion Vector according to the resolution ratio between BL and EL
- Step 4: Directly use this information to find the reference block in the reference image of EL using the reference indices of BL
Proposed algorithm (contd)

- **Step 5**: Produce a reference image for the PU

- **Step 6**: Check for next available BL PU,
  - If found, go to step 1
  - Else, go to step 7

- **Step 7**: Obtain the residual between predicted image and actual image of EL and encode it
Proposed Algorithm

1. Obtain the PU in BL
2. Scale the Size and Motion Vector
3. Locate the reference picture for EL using the reference index of BL
4. Obtain the predicted image using the Scaled Size and Motion Vector
5. Next PU available?
   - Y: Proceed to the next step
   - N: Obtain Residual image and encode
Experimental conditions

• Random access profile is used for coding with GOP (Group of pictures) size equal to 8.

• Coding tree block size is fixed to 32x32 pixels for luma with maximum depth of 3 resulting in minimum CU size of 8x8 pixels for luma.

• The proposed algorithm is implemented using C++ programming language.

• The sequence is not encoded in real time due to hardware limitations.

• The code revision used for this work is revision HM 10.0-dev-SHM [15].

• The work was carried out using an Intel(R) Xeon(R) CPU E3-1220 V2 with Microsoft Windows 7 64bit version running with 8 GB RAM at a speed of 3.1GHz.
Experimental conditions [16]

- The proposed algorithm is evaluated using following test sequences at 30 frames/sec with 4 QPs of 22, 27, 32 and 37 for BL and 20, 25, 30, 35 for EL recommended by JCT-VC.

<table>
<thead>
<tr>
<th>No.</th>
<th>Sequence name</th>
<th>Resolution</th>
<th>Type</th>
<th>No. of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>City</td>
<td>176x144</td>
<td>QCIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>352x288</td>
<td>CIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>704x576</td>
<td>4CIF</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Crew</td>
<td>176x144</td>
<td>QCIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>352x288</td>
<td>CIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>704x576</td>
<td>4CIF</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>Harbour</td>
<td>352x288</td>
<td>CIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>704x576</td>
<td>4CIF</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>Ice</td>
<td>176x144</td>
<td>QCIF</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>352x288</td>
<td>CIF</td>
<td>30</td>
</tr>
</tbody>
</table>
Test Sequences [16]
Results: Encoding time gain (1)

CITY: (BL- QCIF, EL- CIF) - 30 frames

- Original
- Proposed

CREW: (BL- QCIF, EL- CIF) - 30 frames

- Original
- Proposed

CITY: (BL- CIF, EL- 4CIF) - 30 frames

- Original
- Proposed

CREW: (BL- CIF, EL- 4CIF) - 30 frames

- Original
- Proposed
Results: Encoding time gain (2)

HARBOUR: (BL- CIF, EL- 4CIF)- 30 frames

ICE: (BL- QCIF, EL- CIF)- 30 frames
BD-PSNR vs. QP (1)

CITY: (BL- QCIF, EL- CIF)- 30 frames

- Proposed vs Original

- BD-PSNR in dB

- QP

- 20
- 30
- 35

0.275
0.274
0.311

CITY: (BL- CIF, EL- 4CIF)- 30 frames

- Proposed vs Original

- BD-PSNR in dB

- QP

- 20
- 30
- 35

0.298
0.284
0.244

CREW: (BL- QCIF, EL- CIF)- 30 frames

- Proposed vs Original

- BD-PSNR in dB

- QP

- 20
- 25
- 30
- 35

0.242
0.253
0.270
0.271

CREW: (BL- CIF, EL- 4CIF)- 30 frames

- Proposed vs Original

- BD-PSNR in dB

- QP

- 20
- 30
- 35

0.266
0.202
0.228
0.228
BD-PSNR vs. QP (2)

**HARBOUR: (BL- CIF, EL- 4CIF)- 30 frames**

- QP 20: 0.275
- QP 25: 0.247
- QP 30: 0.274
- QP 35: 0.251

**ICE: (BL- QCIF, EL- CIF)- 30 frames**

- QP 20: 0.231
- QP 25: 0.288
- QP 30: 0.248
- QP 35: 0.240
BD-bitrate vs. QP (1)

**CITY: (BL- QCIF, EL- CIF)- 30 frames**

- BD-bitrate in %
  - QP 20: -21.869
  - QP 25: -29.103
  - QP 30: -27.301
  - QP 35: -20.619
  - Proposed vs Original

**CITY: (BL- CIF, EL- 4CIF)- 30 frames**

- BD-bitrate in %
  - QP 20: -26.047
  - QP 25: -23.995
  - QP 30: -20.039
  - QP 35: -22.233
  - Proposed vs Original

**CREW: (BL- QCIF, EL- CIF)- 30 frames**

- BD-bitrate in %
  - QP 20: -20.403
  - QP 25: -18.328
  - QP 30: -19.599
  - QP 35: -19.215
  - Proposed vs Original

**CREW: (BL- CIF, EL- 4CIF)- 30 frames**

- BD-bitrate in %
  - QP 20: -21.964
  - QP 25: -19.988
  - QP 30: -18.267
  - QP 35: -17.083
  - Proposed vs Original
BD-bitrate vs. QP (2)

HARBOUR: (BL- CIF, EL- 4CIF)-
30 frames

ICE: (BL- QCIF, EL- CIF)- 30 frames
PSNR vs. bitrate (1)

CITY: (BL- QCIF, EL- CIF)- 30 frames

CITY: (BL- CIF, EL- 4CIF)- 30 frames

CREW: (BL- QCIF, EL- CIF)- 30 frames

CREW: (BL- CIF, EL- 4CIF)- 30 frames
PSNR vs. bitrate (2)

**HARBOUR: (BL- CIF, EL- 4CIF)- 30 frames**

**ICE: (BL- QCIF, EL- CIF)- 30 frames**
Conclusions

- Decrease in encoding time ranges from 23% to 28%
- Increase in BD-PSNR ranges from 0.2dB to 0.3dB
- Decrease in BD-bitrate ranges from 17% to 29%
- PSNR drop ranges from 0.93% to 1.5%
- Extra memory used by the proposed algorithm is 152bytes
Future work

- Time complexity of the base layer codec in scalable HEVC can be reduced by many other ways [23][24][25]
  - using early termination for inter-prediction mode decision can be implemented to reduce the base layer encoding complexity.
  - Early CU decision leading to early termination can also be employed to reduce the encoding complexity.
- Areas for complexity reduction in inter-layer prediction can be explored [26][27].
- Mode decision algorithms to reduce the complexity in scalable video coding can be investigated [23][26][27].
- Intra prediction, inter-prediction complexity reduction algorithms in both base and enhancement layers along with fast inter-layer prediction algorithms can be applied for further reduction in the codec complexity to get better results.
Acronyms

• **AVC** – Advanced Video Codec.
• **AMVP** – Advanced motion vector prediction.
• **BL** – Base Layer.
• **BO** – Band Offset.
• **CABAC** – Context Adaptive Binary arithmetic coding.
• **CAVLC** – Context Adaptive Variable Length Coding.
• **CB** – Coding Block.
• **CIF** – Common Intermediate Format.
• **CTB** – Coding Tree Block.
• **CTU** – Coding Tree Unit.
• **CU** – Coding Unit.
• **DC** – Direct Current.
• **DCT** – Discrete Cosine Transform.
• **DST** – Discrete Sine Transform.
• **ED** – Entropy Decoder.
• **EL** – Enhancement Layer.
• **HD** – High Definition.
• **HDTV** – High Definition Television.
Acronyms

- **HEVC** - High Efficiency Video Coding.
- **IP** - Intra Prediction.
- **IQ** - Inverse Quantization.
- **IT** - Inverse Transform.
- **ITU-T** - International Telecommunication Union-Telecommunications standardization sector.
- **ISO** - International Standardization Organization.
- **JCT-VC** - Joint Collaborative Team on Video Coding.
- **LCU** - Largest Coding Unit.
- **LM** - Linear Mode.
- **LP** - Loop Filtering.
- **MC** - Motion compensation.
- **MPEG** - Moving Picture Experts Group.
- **MV** - Motion Vector.
- **PB** - Prediction Block.
- **PDA** - Personal Digital Assistant.
- **PU** - Prediction Unit.
Acronyms

- QP - Quantization Parameter.
- QVGA - Quarter Video Graphics Array.
- SAO - Sample Adaptive Offset.
- SHVC - Scalable High efficiency Video Coding.
- SNR - Signal to Noise Ratio.
- SVC - Scalable Video Coding.
- TB - Transform Block.
- TU - Transform Unit.
- TV - Television.
- VCEG - Video Coding Experts Group.
- VCL - Video Coding Layer.
- VGA - Video Graphics Array.
- 1-D - 1 Dimensional.
- 2-D - 2 dimensional.
- 3D - 3 Dimensional.
- 4CIF - 4x CIF.
References

References


References


References


Thank You