Hardware Acceleration of Video Quality Metrics

Xing C. Chen, Deepa Palamadai Sundar, Visala Vaduganathan
ASIC & FPGA Engineer
1. Quality Metrics for Video Transcoding
2. Why Hardware?
3. Supported Metrics in Proposed Accelerator
4. Accelerator Architecture
5. Experimental Results
6. Conclusion
Quality Metrics for Video Transcoding

• Video quality metrics evaluate the loss of fidelity of a transcoded video w.r.t. its original

• Three categories of objective video quality metrics
  o Full reference: pixel-wise comparison between distorted and original
  o Reduced reference: comparison between extracted features of both videos
  o No reference: measure of quality without an original

• Quality scores are usually computed at different viewport resolutions

• Scores are used to determine best streaming resolution
Why Quality Metrics in Hardware?

• Quality metrics important for high quality transcoder systems
• Quality metrics are often complex and compute intensive
  o Per-pixel computation
  o Local image mean/variance
  o Elaborate/Wide filters: Gaussian, Sobel
  o Non-trivial functions such as log2()
  o High precision requirement (floating point in SW)

• Wide range of viewport resolutions to compute: 240p to 4K
• Could consume much more CPU resource than encoding itself
• Power hungry
• Large data transfer overhead between QM SW and encoder HW
Supported Metrics in Proposed Accelerator (1)

- **PSNR (Peak Signal-to-noise Ratio)**
  - Pixel-wide difference in both luminance and chrominance

- **SSIM (Structural Similarity Index)**
  - \[ SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\mu_x^2 + \mu_y^2 + c_1}(\sigma_x^2 + \sigma_y^2 + c_2) \]
  - Local mean (\( \mu \)) and variance/covariance (\( \sigma \)) computation
  - Libvmaf version: as described in original SSIM paper
  - FFmpeg:
    - Scores computed only on 4x4 pixel grid
    - Overlapped 8x8 window approximation
Supported Metrics in Proposed Accelerator (2)

- **MS-SSIM (Multi-Scale SSIM)**
  - SSIM scores computed on 5 scales
  - Gaussian filtering for local mean/variance, decimation
  - Final score is product of per level scores

- **VIF (Visual Information Fidelity)**
  - Per level score computed on 4 scales
  - Gaussian filtering for local mean/variance, decimation
  - Multiple Log2() computations for each pixel

- **No-reference blurriness metric**
  - Gaussian blur
  - Sobel filter
  - Edge width search
Accelerator Architecture

- Accelerator can speed up the quality metrics compute
- Can be programmed to compute scores for any of the supported full reference metrics
- Ability to provide No-reference blurriness score and PSNR in addition to full reference metric

- Two main components:
  - DMA controller
  - Compute Kernel
Compute kernel

- Compute kernel is the Heart of the accelerator
- Three different kernels available
  - FFMPEG kernel
  - SSIM kernel
  - Blur kernel
- Scaler support to upscale/downscale both reference and distorted frames
  - Allows inline processing
  - Programmable coefficients that offer flexibility
  - Optimize memory BW – avoid the need to read/write scaled output to/from memory
- Block level scores support – this is useful in identifying regions that have higher impact on quality within a frame
FFMPEG Kernel

- Computes SSIM index based on 8x8 overlapped approximation algorithm
- 5 components are computed – a, b, \(a^2\), \(b^2\) and \(ab\) which corresponds to mean, variance and covariance components
- Using these components the L and CS score are computed which are then combined to generate SSIM index per pixel.
- Cost(area/power) is directly proportional to the number of dividers and multipliers used. This dictates:
  - the number of pixels processed per cycle
  - the number of kernels that can operate in parallel to improve performance
SSIM Kernel

- Unified kernel to compute SSIM index for single scale as well as multi-scales
- Single scale:
  - Five components are computed - a, b, a^2, b^2 and ab
  - These components are smoothened using a 11 tap Gaussian blur filter before computing L and CS score and the final SSIM Index
- Multi scale:
  - Same kernel as single scale used for compute
  - The blurred output components ‘a’ and ‘b’ of each scale are sent through a dyadic downsampler in addition to computing the SSIM index
  - This downscaled data is fed back as input to the same kernel to compute SSIM index for higher scales
SSIM Kernel for VIF scores

- VIF metric relies on same fundamentals (the nature-scene statistics framework) as SSIM. This helps to reuse the same kernel to compute VIF metrics.

- To support VIF scores:
  - Kernel is enhanced to perform logarithm operation on the variance/covariance ($\sigma$) components.
  - 11 tap Gaussian filter used across all levels.
Blur Kernel

- This kernel computes the blur score used in No-reference quality metrics
- Once the reference frame is read from memory:
  - Smoothen input image: 5 tap Gaussian blur filter
  - Edge detection: Sobel filter to compute gradients and search direction
    \[ G_z = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \]
  - Compute edge width (spread): search in direction computed by Sobel operator within a search window of size NxN
Experimental Results
QM validation: Floating point vs Fixed Point

• QM A-model:
  - Floating point score computation function directly from ffmpeg/libvmaf
  - Put in same test harness as HW C-model

• QM C-model:
  - Fixed point representations
  - Any other HW approximations for complex functions such as log2()
  - Numerical stability guards

• Feeds 400 sequences at 4 resolutions for two quality levels (qp values) to both models
## Fixed vs Floating point approximation – Average Absolute Error

<table>
<thead>
<tr>
<th>QP VALUE</th>
<th>RESOLUTION</th>
<th>SSIM FFMPEG</th>
<th>SSIM LIBVMAF</th>
<th>MS_SSIM</th>
<th>VIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>360p</td>
<td>0.00004</td>
<td>0.00023</td>
<td>0.00079</td>
<td>0.01352</td>
</tr>
<tr>
<td></td>
<td>480p</td>
<td>0.00004</td>
<td>0.00062</td>
<td>0.00102</td>
<td>0.01358</td>
</tr>
<tr>
<td></td>
<td>720p</td>
<td>0.00004</td>
<td>0.00032</td>
<td>0.00081</td>
<td>0.01387</td>
</tr>
<tr>
<td></td>
<td>1080p</td>
<td>0.00004</td>
<td>0.00050</td>
<td>0.00087</td>
<td>0.01258</td>
</tr>
<tr>
<td>31</td>
<td>360p</td>
<td>0.00005</td>
<td>0.00024</td>
<td>0.00084</td>
<td>0.01409</td>
</tr>
<tr>
<td></td>
<td>480p</td>
<td>0.00004</td>
<td>0.00063</td>
<td>0.00108</td>
<td>0.01436</td>
</tr>
<tr>
<td></td>
<td>720p</td>
<td>0.00004</td>
<td>0.00035</td>
<td>0.00092</td>
<td>0.01451</td>
</tr>
<tr>
<td></td>
<td>1080p</td>
<td>0.00004</td>
<td>0.00052</td>
<td>0.00093</td>
<td>0.01307</td>
</tr>
</tbody>
</table>
Experimental Results - Bandwidth Comparison

• CPU vs Accelerator read bandwidth to compute the below metrics for different resolutions
  o FFMPEG SSIM
  o PSNR
  o No-reference blurriness metric

• Number of frame reads:

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>Accelerator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference frame</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Distorted frame</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

• The proposed architecture can improve the performance per unit of power (perf/W) by 100x magnitude
Conclusions

- Proposed architecture can tremendously improve performance of objective quality metrics compute compared to CPUs.

- The current architecture can also be enhanced to offer support:
  - to compute supported metrics for chroma components
  - to calculate VMAF scores by using programmable Gaussian blur filters per VIF level and addition of DLM metric

- This being the first step in enhancing the quality compute operations, more complex algorithms can be explored to offload them to ASIC.
Thank you!