Recent Developments in Scalable Coding in VP9

Alex Eleftheriadis, Ph.D.
Chief Scientist & Co-Founder
October 18, 2016
Why scalability in video coding?

Multipoint Control Units - MCUs
Transcoding MCU

- High Complexity & Cost
- High Delay
- Transcoding Loss
• “Scalable Video Coding”
• Multiple layers, each one adding fidelity to the representation provided by the lower layer(s)
• Fidelity dimensions:
  – Frame rate (frames per second)
  – Spatial resolution (width x height)
  – Spatial fidelity (SNR)
• ~20% more bits than single layer (but a great way to spend bits!)
The VidyoRouter (Feb. 2008)

- Selective forwarding
- Very low delay
- No signal processing
Perfectly matches WebRTC client architecture
What did we gain?

- **Error Resilience**
  - More than 20% packet loss rates, using retransmission techniques, among others
- **Low Delay**
  - Less than 10ms input-to-output, compared with >150ms with transcoding MCU
- **Rate Matching**
  - The rate of each output stream can be individually controlled
- **Personalized Layout**
  - Each receiving participant controls their layout
- **Error Localization**
  - Hop-by-hop error correction (e.g., retransmissions) isolates problems
- **Low Complexity**
  - No signal processing
- **Cascading**
  - Very important for large-scale deployments, e.g., in clouds
Simulcasting

For 2:1 resolution ratios:
• ~50% overhead vs. single layer
• ~20% overhead vs. scalable
SFU – Selective Forwarding Unit

- VidyoRouter was introduced by Vidyo in Feb. 2008, and not foreseen by RFC 5117 (RTP Topologies), published at the same time.
- In Oct. 2013 I coined the term: Selective Forwarding Unit (SFU)
- Covers both the scalable and simulcast operations.
- Added in RFC 7667 which obsoleted RFC 5117 in Nov. 2015
  - Section 3.7, “Selective Forwarding Middlebox”
- Already used in academic papers and marketing materials.
**SFUs & Scalability**

- **SFUs are now the norm:**
  - Vidyo & VidyoCloud, Google+ Hangouts, Microsoft Lync 2013 and Skype for Business, Jitsi VideoBridge, etc.

- **Scalability in RTC codecs is now everywhere:**
  - H.264 (Annex G, SVC)
  - VP8 (temporal only)
  - HEVC (H.265) v.1 (temporal) and v.2 (SHVC – spatial/quality)
  - VP9 (Vidyo & Google)
  - Alliance for Open Media - AV1 (temporal mandatory, spatial desirable)

- **Also incorporated in all RTP payload formats**
• **Scalable and Simulcast Video (SSV) Activity Group**
  – [H.264 SVC Bitstream Modes](#) (2012)
    • Temporal & Spatial Scalability
  – [H.264 SVC RTP Transport](#) (2014)
  – [H.265 HEVC Bitstream Modes](#) (2016)
    • Temporal & Spatial Scalability

• **SuperOp! Events**
  – Weeklong, face-to-face meetings among engineers for interop testing
Scalable VP9
Where did Scalable VP9 come from?

- Google originally used SVC in Hangouts
- In August 2013 it switched to VP8 to pursue a WebRTC-based, open source foundation
- In August 2013 we announced that we will be jointly developing scalability extensions to VP9 and WebRTC
- Also co-developing RTP payload format for VP9
- Code has been available in WebM and WebRTC repositories since November 2015
- First widely available release in Chrome in January 2016 (M48)
  - Decoder fully capable of decoding scalable streams
  - Encoder is lacking API support
Well hello there VP9. We’ve been waiting for you.
We will continue to improve VP9 in the next several Chrome releases. This release incorporates the proposed standard RTF packetization for VP9, which was co-designed and developed with Vidyo, Inc., and we will attempt to maintain backwards compatibility from M48 onward. We’re excited about the future of VP9, and we invite developers to experiment and provide feedback.
What is in Chrome today

- Vidyo has developed the spatial and temporal scalability code, as well as the payload format code
- Google has made available the VP9 codec but without enabling the scalable features at the encoder
- The decoder is fully capable of processing scalable streams, and will thus be backwards compatible
- Enabling the scalability features at the encoder requires API additions to the WebRTC specifications
Scalability in VP9
Reference Pictures in VP9

• VP9 allows coding an INTER frame with 3 reference frames: LAST, GOLDEN, and ALTREF.
• Frame header chooses up to three reference frames from a pool of 8, and specifies the frame buffer(s) the coded frame will replace.
• VP9 allows spatial resampling of reference pictures when used for prediction, mid-stream.
  – Reference buffers are scaled to the currently predicted frame’s resolution.
• Spatial scalability design is simpler than H.264 SVC, and similar to HEVC, to enable implementation without additional hardware.
In the buffer pool, previously reconstructed pictures of two different resolutions have been saved, in slots 0/1, 2/3, and 4/5.
The current input picture is downsampled using a non-normative downsampler to get the base layer input picture.
Use LAST_FRAME as a reference frame, pointing to buffer 4.
Steps 3-4: Base Layer Buffering

The reconstructed picture replaces buffer 6.
Use LAST_FRAME and GOLDEN_FRAME as references with buffers 5 and 6, respectively (5 is the full resolution picture and 6 has the low resolution picture).
Steps 7-8: Enhancement Layer Buffering

The reconstructed enhancement layer picture is stored in buffer 7.
• Buffer size limitation (8) dictates what structures can be implemented.
• libvpx implementation supports several schemes:
  – Number of spatial layers can be set through an API call.
  – Three different temporal layering schemes are available through another API.
    • No scalability
    • Two layers (0-1)
    • Three layers (0-2-1-2)
    • Bypass (custom temporal layering, only with single spatial layer)
Performance Analysis
What did we want to test?

- Comparison codecs: VP9, H.264, and H.265
- Relative performance of temporal scalability vs. single-layer coding
  - VP9 only, both compression and speed
- Relative performance of spatial scalability vs. simulcast
  - All codecs, compression only (non-real-time software for H.264 and H.265)
- Relative performance of spatial scalability vs. single layer coding (for sanity purposes)
Data Points

- **VP9 (WebM)**
  - Temporal Scalability Coding Efficiency
  - Encoding Speed Overhead
- **Spatial VP9 (WebM)**
  - Encoding Speed Overhead
  - Coding Efficiency: Intra-only & Low-delay
- **Spatial H.265 (SHM)**
  - Coding Efficiency: Intra-only & Low delay
- **Spatial H.264 (JSVM)**
  - Coding Efficiency: Intra-only & Low delay
Scalability Configurations
Single Layer Coding
Temporal Scalability with Two Layers

<table>
<thead>
<tr>
<th>Temporal Layer</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

- P0
- P1
- P2
- P3
- P4
- P5

...
Temporal Scalability with Three Layers
Spatial Scalability with Two Layers


I → P → P → P → P → …

0 1 2 3 4 → Time
Simulcast Coding

I, P, P, P

0, 1, 2, 3, 4

Time
Combined Spatial/Temporal Scalability
Performance Analysis Methodology
PSNR vs. Bitrate

"Normal" RD-plot

PSNR (dB) vs. Bitrate
BD-PSNR and BD-rate

\[ \text{SNR} = a + b \cdot \text{bit} + c \cdot \text{bit}^2 + d \cdot \text{bit}^3 \]

\[ \text{Bit} = a + b \cdot \text{SNR} + c \cdot \text{SNR}^2 + d \cdot \text{SNR}^3 \]
Experimental Conditions

• Three 720p 30fps sequences:
  – “Four People”, “Kristen and Sara”, “Vidyo4”
• 2:1 spatial scalability
• VP9 encoded using WebM with rate control, CPU mode 7 (default mode for WebRTC) as of December 23, 2015
• H.265 and H.264 encoded using SHM and JSVM with fixed QPs
• Bit allocation across temporal layers: 60-20-20%
• Spatial scalability allocation:
  – QPs selected for 2:1 ratio (base = 33% of total)
  – VP9 rate control set up for 2:1 ratio
Experimental Results
VP9 Temporal Scalability: Coding Efficiency

3 Temporal Layers vs. 1 Temporal Layer

BD-PSNR -0.3 dB (Y)
BD-rate +0.8% (Y)

Temporal scalability is nearly free
3 Temporal Layers vs. 1 Temporal Layer

Just 4.7% slower on the average.

Temporal scalability is nearly free (zero cost in bits and minor cost in speed)
VP9 Spatial Scalability: Speed

2 Spatial Layers vs. Simulcast

8.9% slower on the average; but way faster than real-time ➔ Minor impact
VP9 Spatial Scalability: Compression, Intra-only

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, intra-only

vs Simulcast:
BD-PSNR +2.36 dB
BD-rate -31.2%
⇒ Excellent job as expected since intra prediction works great

vs Single layer:
BD-PSNR -0.20 dB
BD-rate +3.2%
⇒ Base layer costs very little
H.264 Spatial Scalability: Compression, Intra-only

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, intra-only

Vs Simulcast:
BD-PSNR +1.37 dB
BD-rate -21.9%

Vs Single layer:
BD-PSNR -0.72 dB
BD-rate +13.8%
H.265 Spatial Scalability: Compression, Intra-only

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, intra-only

vs Simulcast:
BD-PSNR +0.74 db
BD-rate -12.6%

vs Single layer:
BD-PSNR -1.15 dB
BD-rate +23.2%
Summary, Intra-only

- VP9 works great for creating the enhancement layer vs. simulcast across H.264 and HEVC
- The overhead vs. single layer is smallest in VP9, and is due to the different qualities provided by the base layers in H.264 and HEVC at the target rates

<table>
<thead>
<tr>
<th>Intra-Only</th>
<th>Gain vs Simulcast</th>
<th>Overhead vs Single Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD-Rate</td>
<td>BD-PSNR</td>
</tr>
<tr>
<td>VP9</td>
<td>-31.21%</td>
<td>2.36</td>
</tr>
<tr>
<td>JSVM</td>
<td>-21.87%</td>
<td>1.37</td>
</tr>
<tr>
<td>SHM</td>
<td>-12.65%</td>
<td>0.74</td>
</tr>
</tbody>
</table>
VP9 Spatial Scalability: Compression, Low-delay

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, low delay

vs Simulcast:
BD-PSNR +0.52 dB
BD-rate -13.4%
⇒ Some gain over simulcast, as expected

vs Single layer:
BD-PSNR -0.76 dB
BD-rate +25.4%
⇒ On the high end (~20% expected)
H.264 Spatial Scalability: Compression, Low-delay

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, low-delay

vs Simulcast:
BD-PSNR +0.46 dB
BD-rate -12.3%

vs Single layer:
BD-PSNR -0.74 dB
BD-rate +23.8%
H.265 Spatial Scalability: Compression, Low-delay

2 Spatial Layers vs. 1 Spatial Layer and Simulcast, low-delay

vs Simulcast:
BD-PSNR +0.17 dB
BD-rate -6.6%

vs Single layer:
BD-PSNR -0.55 dB
BD-rate +24.2%
**Summary**

<table>
<thead>
<tr>
<th></th>
<th>Gain vs Simulcast</th>
<th>Overhead vs Single Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD-Rate</td>
<td>BD-PSNR</td>
</tr>
<tr>
<td><strong>Intra-Only</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>-31.21%</td>
<td>2.36</td>
</tr>
<tr>
<td>JSVM</td>
<td>-21.87%</td>
<td>1.37</td>
</tr>
<tr>
<td>SHM</td>
<td>-12.65%</td>
<td>0.74</td>
</tr>
<tr>
<td><strong>Low-Delay</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>-13.37%</td>
<td>0.52</td>
</tr>
<tr>
<td>JSVM</td>
<td>-12.34%</td>
<td>0.46</td>
</tr>
<tr>
<td>SHM</td>
<td>-6.59%</td>
<td>0.17</td>
</tr>
</tbody>
</table>

- **Low-delay:**
  - VP9 has slightly better gain over simulcast vs. H.264 and H.265
  - VP9 has slightly higher overhead vs. single layer but all are around 24-25%
• H.265 appears better BUT VP9 here is real-time!
Relative Performance, “Four People”, Temporal Only

<table>
<thead>
<tr>
<th>Bit rate (Kbps)</th>
<th>VP9</th>
<th>JSVM</th>
<th>SHM</th>
<th>Real Time H.265</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Four People (1 Spatial, 3 Temporal)
Summary

- **Scalable VP9 is ready for prime time**
  - Temporal scalability at zero cost in bits, minor cost in speed (<5%)
  - Real-time performance across the board
  - Relative compression efficiency with spatial scalability vs. simulcast and single layer “on par” with H.264 & HEVC

- **Encoding performance can be improved, potentially outside the open source realm**
Microsoft Edge - VP9 & VP8 status

Platform status
What we've built and what's on our roadmap. Find information about our backlog and upcoming priorities on the Microsoft Edge Dev Blog.

Video tracks
Supported
Build Number 10240+

Vorbis Audio Codec
Under Consideration
5668 Votes

VP8 for RTC
In Development

VP9 Video Playback
Defacto standard
Support in Microsoft Edge
Shipped in Chrome
Shipped in Firefox
Not Supported in Internet Explorer 11
Shipped in Opera
Not Supported in Safari

WAV Audio Support
Supported
Build Number 10240+
Alliance for Open Media – www.aomedia.org

- Founded by Amazon, ARM, Cisco, Google, Intel, Microsoft, Mozilla, Netflix, and NVIDIA on 9/2015
  - 18 members today including Vidyo
- Open source, royalty-free codec ("AV1")
- Aggressive performance and delivery timelines building on VP9
- Real-Time Communications (RTC) Subgroup started in 6/2016
  - Microsoft, Polycom, and Vidyo co-chairing
- Temporal scalability mandatory, spatial scalability desirable