Line by line processing of video on IT Hardware

Kieran Kunhya <kierank@obe.tv>
Introduction

• Existing hardware able to process video on a line-by-line basis, sometimes less
  – Inherently property of minimising line buffers in hardware
  – Allows for $O(\text{lines})$ processing

• Advent of Uncompressed IP makes this doable using IT hardware
  – Lines are inherently packetised

• COVID world, focus on higher compression H.264/HEVC rather than Intra VC-2/JPEG-XS
Who we are?

- Company specialising in software-based encoders and decoders for Sport, News and Channel contribution (B2B)
- Based in Central London
- Build everything in house
  - Hardware, firmware, software
- Not to be confused with:

Open Broadcaster Software
Commercial/Operational Drivers

- Operational requirements for lower and lower latency with reduced budgets
- REMI/Remote Production for sports
  - Backhaul feeds to one or more locations for now socially distanced production
- Editorial requirements for news
  - Harder to interrogate politicians with latency
  - Disadvantage in debates for remote contributors
Benefits of IT Approach

• More flexible approach to infrastructure
• A decoupling of video processing from physical hardware
  – Move processing resources to people’s homes
• Uncompressed IP deployments allowed for simple scale up and down during lockdown
  – Point and click
• Less disrupted supply-chain for IT equipment (larger pool)
Downsides of IT approach

• IT industry does not work on line-by-line approach, common frameworks (DirectShow, GStreamer, FFmpeg) etc all frame based
  – Wait for frame buffer to arrive, do processing, output it, wait for it to arrive, decode it
  – Frames add up quickly

• Small niches of IT understand this
  – Cloud Gaming
  – DSC (Display Stream Compression) – 1 line latency!
    • Used in DisplayPort and HDMI
  – Will be interesting to see if podcast/YouTube world changes this
“Latency” according to the streaming industry
Latency according to the streaming broadcast industry

STREAMING LATENCY AND INTERACTIVITY CONTINUUM

- **COMMON HTTP LATENCIES TODAY**: One-way streams of live events to large audiences; linear programming
- **REDUCED LATENCY**: OTT providers; live-streaming news and sports
- **LOW LATENCY**: UGC live streams; game streaming and e-sports
- **NEAR REAL-TIME**: Two-way web conferencing; telepresence; real-time device control (e.g., PTZ cameras, drones)

Latencies:
- **45+ seconds**
- **18 seconds**
- **05 seconds**
- **01 second**
- **< 01 second**

Us!
Uncompressed IP line by line

- Theoretically could busy loop and read line at a time. Quite costly.
- In practice, have to do 2022-7 and maybe wait for audio, so 5-10 lines is acceptable
- Decapsulate whilst the next chunk arrives
- Ideally go straight from 2110 packet pixel format to internal encoder pixel format (or decoder -> 2110)
  - Difficult to do in practice with modular codebase
- All of this sounds easy but it’s hard!
Uncompressed IP line by line (2)

• Receive chunks
• Do processing on them, insert a logo, whatever
• Release on next PTP epoch
• Plenty of time to do pixel processing
• Again not widely done in IT industry frameworks
• Can’t use most fashionable programming languages with garbage collection that takes 5ms
Some quick thoughts on SDI

• Likewise most standard SDI capture/output boards work on a frame-basis
  – One Chinese manufacturer supporting “early reads”
  – Got to wait for a frame, then convert it to the pixel format you need (~5-10ms)

• Built our own with ~32 line buffer
  – Also can convert to desired pixel format whilst frame arrives on wire, reduces latency further

• Go through a similar codepath to Uncompressed IP
H.264/HEVC Encoding

- A need for < ~100ms encode/decode (glass-to-glass) but at “COVID-friendly bitrates”
- Encode a frame or a field at a time
- Use intra-refresh modes, intra frame spread over stripes, small VBV (= low latency)
- Unlike intra modes, best to allow ratecontrol to work on entire frame
  - Some hardware vendors doing sub-frame encode
- Uncompressed audio essentially mandatory
  - Audio encode latency too high (except Opus)
- **No hit to compression efficiency for frame/field**

Motion adaptive intra refresh for the H.264 video coding standard Ralf M. Schreier, A. Rothermel
Decode frame as it arrives on the wire (1)

• Slices arrive at destination

• Complete frame is built
Decode frame as it arrives on the wire (2)

- Avoid deblocking on slice boundaries
  - Independently decodable slices
- On high bitrate streams latency/throughput tradeoff
  - May need to buffer slices and decode simultaneously to gain enough throughput to decode in realtime
- Decoders output chunks when done
  - Uncommon in IT decoders
  - Cloud gaming uses this
- Render slices when buffer is big enough
Decode frame as it arrives on the wire (3)

• Compressed video likely not PTP locked, so have to frame synchronise before outputting as 2110
  – Have to resample audio as well, also have to decode audio as it arrives on the wire...
  – Again a concept not supported in IT

• Some applications (where not used for production, e.g monitoring, PTZ control) could free-run to lower latency
  – Output slices to network as soon as it’s done
  – Illegal 2110 stream
VC-2/JPEG-XS

• Similar except chunks encoded as they arrive
  – 32-128 lines of latency
  – ~100-200Mbps of bitrate
COVID Tradeoffs

- Some operations will take place in a home or non-traditional environment
  - Could be that 5-7 round trip time of RIST/Zixi is tolerable, gives Zoom-like experience, low-latency with the odd hit here and there. ~200ms end to end not bad.

- Start seeing broadcast products having home focus instead of Zoom and others being used as stopgap

- Use of 4:2:2 10-bit, SDI and other things not usually used at home
  - Again for lower latency than PC/cellbonding based products
  - Remote editing/grading etc
  - Employers paying for dedicated connections
COVID Tradeoffs (2)

• Quite a few domestic issues with 100GbE switches at home
  – Probably not going to see much 2110 in a home environment
Conclusions

• In the end best of both worlds
  – COTS hardware but sub-frame encoding and decoding

• Right image 5 frames end to end at modest bitrate (sub-100ms), can cut more frames in the future
  – 2110 in and 2110 out
  – Could be SDI

• 2110 -> MPEGTS -> 2110 remote production