Standard quality metadata in compressed bitstreams

Ioannis Katsavounidis
Meta
Vision

- Make large-scale video quality data available to users/service operators/researchers
- Full-reference metrics are not reproducible and only available at the source/encoder
- Some metrics are very expensive (computationally) and these pre-calculated scores, properly combined, can offer lower complexity and improved accuracy
- Such FR scores at the encoder side can supplement - and even replace - NR metric calculations at the receiver side
Current status

- Working document, tracking rationale and various discussions:
  https://docs.google.com/document/d/1v02cd7tFz-YozfctAy2OkKz1dGX8eHuk-1lueaxobYE/edit?usp=sharing

- Working document, on the proposed standard for T.35 signaling:
  https://docs.google.com/document/d/1zrUnttz4LxYbBclsf8nYQ__13TVom6iH54ZPK6GaNws/edit?usp=sharing
Rationale
- Incorporate video quality metric as standard metadata in compressed bitstreams
- Track all phases in the lifetime of a video (capture, transmission, transcoding, distribution)
- Allow optimized decisions of service providers in choosing encoding options (codec, resolution, quality/bitrate)
- Offer higher QoE to subscribers of video services

Transcoding example (FFMPEG/x264)

Elementary video quality information about this encode is readily available
- Per frame average QP
- Per frame PSNR (Y/U/V)
- Per frame SSIM

At near-zero compute overhead
How about camera capture?

Lens/CMOS sensor -> RGB/YUV frame -> Video encoder ASIC -> Compressed file

Most HW video encoders include video quality metrics per frame – at least for debugging issues.
The life-cycle of a UGC video

- Video captured on a cellphone
- Video sent by SMS/WhatsApp/Messenger/(...) to friend
- Video sent by SMS/WhatsApp/Messenger/(...) to friend
- Video sent by SMS/WhatsApp/Messenger/(...) to friend
- Another user saves on mobile/tablet
- Video gets uploaded to YouTube
- A user saves video on their desktop
- Video is posted on Facebook
- Video is posted on Instagram

Diagram shows a decrease in quality over time.
Challenge

- Each transcoding pipeline estimates source video quality using no-reference metrics to determine best ingestion strategy.
- During transcoding, full-reference quality metrics are generated to determine best encoding settings/ABR strategy.
- Estimation errors propagate and accumulate when cascading multiple transcoding pipeline.
- No-reference metrics require significant compute overhead.
VQEG as a T.35 “terminal provider”


Three parts: country code, terminal provider code and terminal provider oriented code in the case of terminal specific non-standard facilities. The country code identifies the country (US), the terminal provider code identifies the provider (VQEG) and the terminal provider oriented code is defined by each provider (various codes, one corresponding to each payload).

VQEG; registered in the US with ITS street address

The Manufacturer code assigned to you is:

Mfg Code - first (MSB) byte: Hex: 48
Mfg Code - second (LSB) byte: Hex: 10

T.35 messages can be incorporated through minimal header in existing and future compressed bitstreams (for example, by introducing special SEI messages in AVC)

They can be specified at both elementary video streams and system-bitstreams, using the exact same syntax
Proposal summary

3 different payloads, corresponding to 3 distinct “terminal provided oriented” codes

- Summary payload (intended to store information about the entire compressed video sequence)
- Metric instance descriptor (intended to describe the exact configuration to calculate one specific metric value)
- Metric instance value (value corresponding to one of the metric instance descriptors)
Proposal summary (cont’d)

Frequency of these payloads (suggested, for VOD applications)

- Summary payload: 1 per sequence/file
- Metric instance descriptor: N per sequence/file, capturing N quality metrics
- Metric instance value:
  - Fine granularity: N per frame
  - Medium granularity: N per GOP/independently decodable coded unit
  - Coarse granularity: N per minute
  - Summarized version: same as for “summary payload”

Frequency of these payloads (suggested, for LIVE applications)

- Consider every 10 minutes of a LIVE stream as a “sequence” and apply the corresponding frequencies listed above
Detailed proposal

- https://docs.google.com/document/d/1zrUnttz4LxYbBcisf8nYQ___13TVom6iH54ZPK6GaNws/edit?usp=sharing
Next steps

- Have a working draft of the proposed T.35 payloads
- Share the working draft with SSOs (JVET, AOM, others?)
- Collect and incorporate feedback