Efficient measurement of quality at scale in Facebook video ecosystem

Regunathan, Shankar, Wang, Haixiong, Zhang, Yun, Liu, Yu (Ryan), Wolstencroft, David, et al.

Shankar L. Regunathan, Haixiong Wang, Yun Zhang, Yu (Ryan) Liu, David Wolstencroft, Srinath Reddy, Cosmin Stejerean, Sonal Gandhi, Minchuan Chen, Pankaj Sethi, Amit Puntambekar, Mike Coward, Ioannis Katsavounidis, "Efficient measurement of quality at scale in Facebook video ecosystem," Proc. SPIE 11510, Applications of Digital Image Processing XLIII, 115100J (21 August 2020); doi: 10.1117/12.2569920

SPIE.

Event: SPIE Optical Engineering + Applications, 2020, Online Only

Efficient Measurement of Quality at Scale in Facebook Video Ecosystem

Shankar L. Regunathan, Haixiong Wang, Yun Zhang, Yu (Ryan) Liu, David Wolstencroft, Srinath Reddy, Cosmin Stejerean, Sonal Gandhi, Minchuan Chen, Pankaj Sethi, Amit Puntambekar, Mike Coward, Ioannis Katsavounidis
Facebook, 1 Hacker Way, Menlo Park, CA 94025

ABSTRACT

This paper describes FB-MOS, a metric that is used to measure video quality at scale in FB Video Ecosystem. Facebook processes a very large number of videos daily that collectively receive billions of views each day and hence both the accuracy and computational complexity of the metric are equally important. As the quality of uploaded user-generated content (UGC) source itself varies widely, FB-MOS consists of both a no-reference metric component to assess input (upload) quality and a full-reference component to assess quality preserved in the transcoding and delivery pipeline. FB videos can be watched on a variety of devices (Mobile/Laptop/TV) in varying network conditions, and often switched between in-line view and full-screen view during the same viewing session. We show how FB-MOS metric can accurately account for all this variation in viewing condition while minimizing the computation overhead to offer such measurement. We also discuss how this metric allows for end-to-end quality monitoring at scale, as well as guide encoding and delivery optimizations. The paper also discusses some of the optimizations to enable its use to achieve real-time quality measurement for Live videos. Another aspect of the Facebook video product is the wide variation in popularity of videos where less popular UGC content may receive relative very few views while highly watched professional or viral UGC content can receive millions of views. We discuss how the computational overhead of the metric can scale with the popularity of video where more compute is expended on more popular videos to get more accurate metrics while spending less compute on less popular videos.

Keywords: Video quality metrics, video compression, video coding, computational complexity, SSIM, VMAF

1. INTRODUCTION

With the recent rise of streaming media industry[1], video streaming has become the biggest application in terms of data usage in internet traffic worldwide. The central challenge in video streaming has been to deliver the best video quality to the user given the constraints of network bandwidth and viewing device. This has also led to increased interest in assessing the perceptual quality of compressed video.

While video quality means different things to different people, for the rest of this paper, we define video quality as the overall perceived quality of a video, expressed by a human subject who is watching it on a certain display, at a certain viewing distance. Further, with the increase in user-generated content (UGC), there needs to be a distinction between artistic quality ("production value") and video quality. In defining video quality this paper focuses on how that content is being presented, rather than what the content is. The gold standard when it comes to measuring video quality is to perform user subjective testing as in the popular international standard ITU-R BT.500[2]. However, due to both privacy concerns as well as the scale of videos on the leading video streaming video providers, there is a need for an automated way of measuring video quality that doesn't involve human observers. The problem of measuring video quality at scale[3] and its use for monitoring for regressions[4] in the video processing pipeline has received increased attention.

This paper focuses on measuring quality at scale in Facebook Video Ecosystem which consists of multiple video products including main Facebook (FB)-app, Instagram, Messenger, WhatsApp and Oculus. These products span both VOD and Live use-cases and includes both Professional and User-Generated-Content (UGC). These videos collectively receive billions of views each day and this billion-scale dictates many of the choices in FB video processing pipeline such as to maximize video quality of experience, under a given cost and energy budget in our data centers. As a result, computational efficiency is a key consideration to video quality measurement at FB-scale. An additional aspect of the Facebook video experience is the wide variation in ingested (uploaded) video quality and the diversity of viewing devices. We will show how FB-MOS addresses these aspects, as well.

Applications of Digital Image Processing XLIII, edited by Andrew G. Tescher, Touradj Ebrahimi, Proc. of SPIE Vol. 11510, 115100J · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2569920

The remainder of this paper is organized as follows: Section 2 gives an overview of the end-to-end video processing flow in FB Video Ecosystem as well as some of the key challenges in measuring quality. Section 3 presents the building blocks of the FB-MOS quality metric that was designed to address these challenges. Section 4 describes the use of FB-MOS for quality measurement at scale. Section 5 describes how FB-MOS metric allows for end-to-end quality monitoring as well as guiding encoding and delivery optimizations. Section 6 discusses some of the optimizations made to FB-MOS to make it suitable for measuring quality for the Live product use-case.

2. OVERVIEW OF FB VIDEO PROCESSING

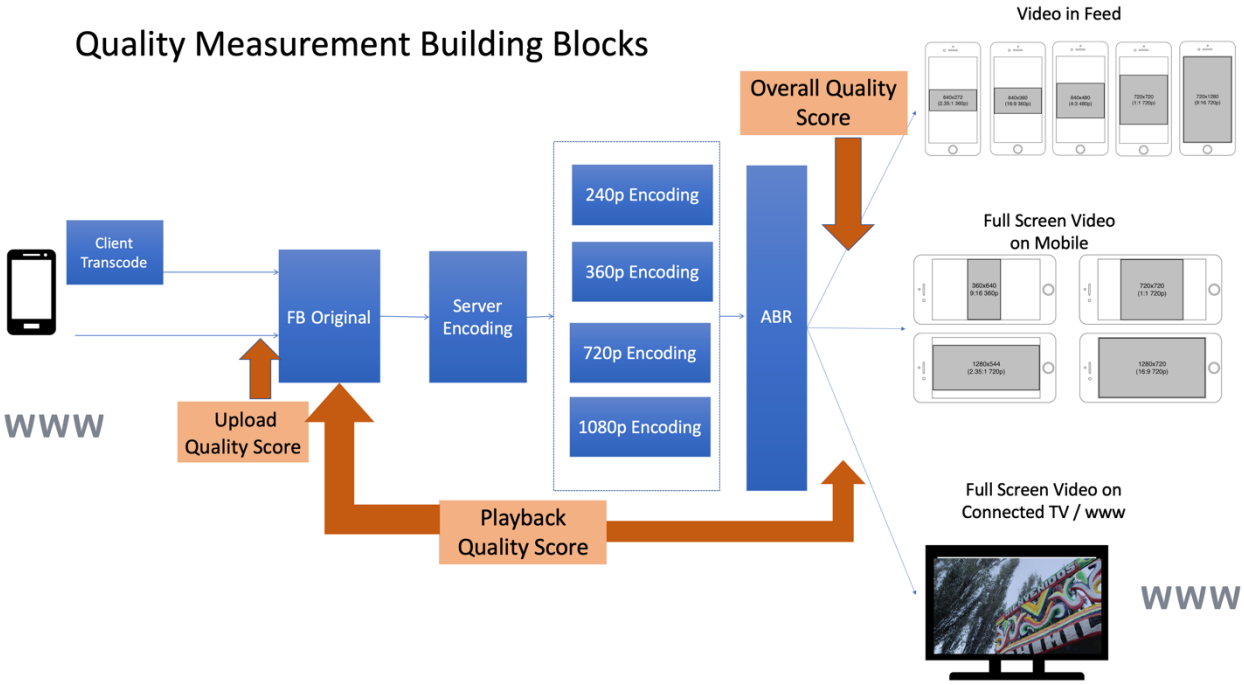


Figure 1. A simplified overview of end-to-end video processing flow in FB ecosystem

An overview of the end-to-end FB video processing flow is shown in Figure 1 which consists of two major components:

- Upload (Ingestion) video to FB by a publisher/user. We refer to this uploaded video as the FB-original.
- Encoding into multiple qualities/resolutions and ABR delivery for video playback on viewing devices. This encoding is performed using standard video compression algorithms (codecs) such as H.264/AVC, VP9 and AV1.

We discuss these components in more detail in the following sub-sections emphasizing the implication for quality measurement.

2.1 Upload/Ingestion

Most videos are uploaded from mobile clients or browser across multiple video products, such as Facebook-app and Instagram. However, some products also offer customized ingestion pipelines for specific curated content. It is important to note that the quality of uploaded (ingested) video can vary widely. At one end, curated content and some UGC can be of very high quality (1080p/2K/4K) at very high bitrates. At the other end, a significant fraction of uploaded UGC can be of very low quality where the resolution is low (360p and below) and the bitrate of ingested content is low (below 500Kbps).

There are two main reasons for low ingest quality:

1. These videos are effectively low bitrate video streams downloaded from other video products such as WhatsApp/YouTube/Messenger and re-uploaded back into the FB ecosystem.
2. These videos originate from high quality sources whose quality was degraded by transcoding on the client app prior to upload. Client transcoding is often necessary to optimize upload reliability and minimize latency when video is uploaded from poor network connections, such as 2G/3G mobile networks. In Live video upload, the ingested quality can keep varying as the client transcoding parameters adapt to changing network conditions.

Further, some video products support scenarios where it is easy to edit and remix videos before upload. These editing tools support addition of stickers/text/images/animation on top of video. A lot of popular "meme" videos often start as low-quality sources with stickers/text overlaid. These scenarios make it hard for automated algorithms to assess video quality as the quality is often in the "eye of the beholder". For example, viewers often regard the perceptual quality ("sharpness") of stickers/text as much more important than the background video quality. Notwithstanding the difficulty of this problem, the wide variation of upload quality makes it imperative to have an ingest quality metric component as part of measuring end-to-end quality.

2.2 Encoding/Delivery/Playback

Our video pipeline does the following processing steps on the uploaded FB original to make the video available on viewer devices

1. Server produces multiple encodings of an FB original at different bitrates and/or resolutions (qualities). Compression is performed using standard video compression algorithms (codecs) such as H.264/AVC, VP9 and AV1. As it is typically the case in ABR streaming, encoded segments produced for different representations are temporally aligned to enable seamless bitstream switching at the client side.
2. When a user starts to watch a video, the device fetches the manifest for this video. Note that the manifest generation needs to take into account the device and user characteristics as described below.
3. The playback client on the device chooses which one of the multiple encodings to fetch at any given time based on both the network conditions as well as device capability.

This basic approach to client-driven ABR streaming is very popular and is the de facto standard in all major streaming providers. A standard approach to measuring quality preserved between the uploaded original and the encoding delivered to the client is a full-reference metric such as SSIM[4] or VMAF[3]. We next discuss some of the specific challenges in ABR streaming video quality measurement in FB ecosystem.

First, there is a vast diversity in viewing devices and playback conditions. Mobile phones, web browsers, TV sets are very different in screen resolutions as well as device capability. Further, mobile devices (Android in particular) vary widely in their capability and screen resolution. For example, certain devices lack decoding support for VP9 or certain H264 profiles and ABR streaming must account for this.

Further FB video products allow for users to switch between in-line viewing, where video content occupies a relatively small portion of the screen, and full-screen viewing, which causes a change in effective playback resolution (also called viewport resolution). For example, a very common scenario is for a user to start watching a video in-line and then switch to full-screen.

These scenarios make it challenging for video quality assessment since the perceived quality of the exact same encoding is different based on the viewport resolution.

3. FB-MOS BUILDING BLOCKS

The two main building blocks for FB-MOS are a no-reference metric to measure quality of uploaded video and a full-reference metric to measure quality preserved in the video processing pipeline.

3.1 Upload Quality Metric

Estimating the quality of an uploaded video is - by definition - a no-reference video quality problem (particularly when client transcoding is not involved). Still, uploaded videos are ingested in some compressed form, for example H.264/AVC, which carries additional information about the source, such as motion vectors and QP values for each frame/macroblock, which can improve estimation of quality; in such case, we can treat it as a reduced-reference video quality problem.

Two very intuitive features that can be used for no-reference video quality assessment are a "blurriness" indicator and the actual bitrate of the ingested source. However, both no-reference and reduced-reference video quality assessment are very challenging problems in general. For example, the blurring of video could be due to poor quality or due to artistic intent and a pixel-domain no-reference metric may not be able to distinguish the two cases. Similarly, some low-quality videos have very high uploaded bitrate (possibly due to repeated transcoding) and a reduced reference metric that relies on the bitrate of the uploaded video may find it very hard to classify these videos as low quality.

Our approach to computing upload metric uses multiple algorithms, both no-reference algorithms that relies on a blurriness feature and reduced-reference algorithms which uses the bitrate achieved when transcoding the source video to a fixed quality. The final upload quality metric score is computed from a combination of these individual scores. We also compute confidence signals on the upload quality metric based on how much these quality scores agree.

Note that there are often product specific requirements in identifying low quality uploads. Certain products require high confidence in identifying high quality uploads (for ranking and promotion), while other products require high confidence in low quality uploads (for spam detection). These products can use a combination of the individual quality scores as well as the confidence values to address their needs.

3.2 Playback Quality Metric

Playback quality metric is the full-reference metric component that measures the quality preserved by the video processing pipeline between FB-original and the encoding delivered to the viewing device.

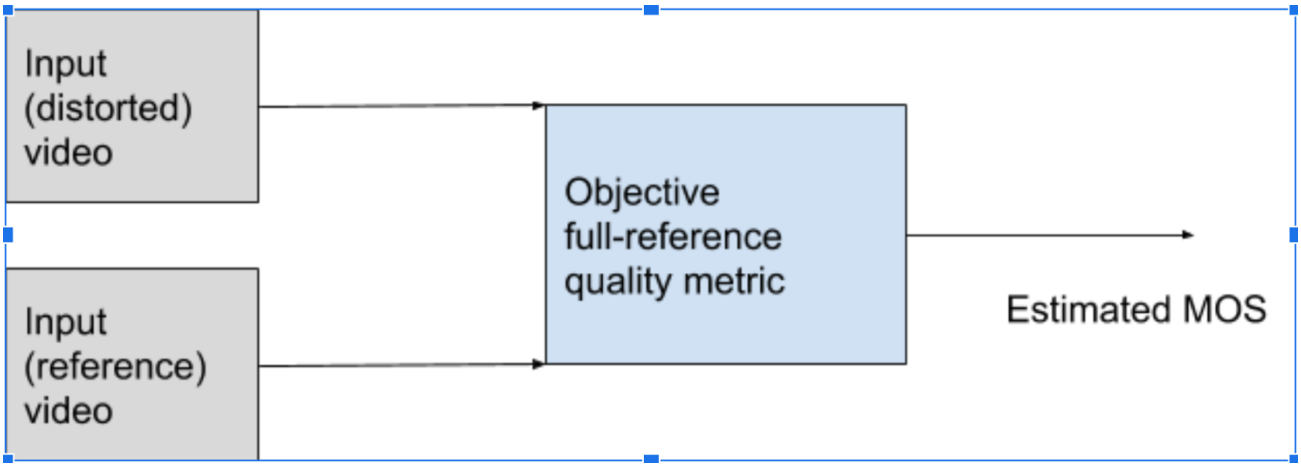


Figure 2. Typical full-reference video quality metric system diagram. The distorted video is the result of encoding (and optionally scaling) an input source video.

The oldest full-reference metric is called Peak-Signal-to-Noise-Ratio (PSNR). Structural similarity index (SSIM) was proposed in 2004 as a better alternative to PSNR. It has been shown through multiple subjective experiments that SSIM has higher correlation with subjective image quality than PSNR. While the complexity of SSIM is higher, it is still sufficiently low such that SSIM can be efficiently introduced in video transcoding pipelines. Yet another, 3rd generation video quality full reference metric that has gained wide popularity, in particular for premium videos, is VMAF - Video Multi-method Assessment Fusion. The key concept behind VMAF is that one can use existing metrics, such as PSNR and SSIM discussed earlier, but also higher performing metrics, namely VIF[5] and ADM[6], as well as a measure of temporal activity, and fuse them by support vector regression (SVR)[9] to get accuracy higher than that of each individual metric. This accuracy, although, comes at the cost of higher complexity, which is approximately 100x compared to SSIM and this complexity can be a significant burden for use of VMAF for long-tail video at FB-scale.

FB-MOS uses SSIM as the core full-reference metric to assess quality preserved in the video processing pipeline. Due to its non-linear nature - an issue known to many researchers in the video quality area - we use a mapping of SSIM scores into a linear scale of 0 to 100, by applying piecewise linear interpolation from the key points listed in Table 1, which was obtained through subjective validation. We are currently working on improving the accuracy of our FBMOS metric by selectively introducing VMAF for premium videos, while conducting further subjective experiments to make this integration better suited for the wide variety of FB video content.

SSIM	1.0	0.99	0.98	0.97	0.96	0.95	0.925	0.9	0.85	0.8	0.7	0.6	0.3	0.0
MOS	100	88.39	77.77	70.66	63.96	57.82	45.12	35.74	23.68	16.77	9.72	6.39	2.69	0

Table 1

3.3 FB-MOS Validation

Humans are the end user of videos, so subjective video quality assessment by humans is the ground truth of quality measurement. For this reason, we designed and performed subjective studies, with the goal to gather multiple opinion scores from viewers, then statistically calculate a single mean opinion score (MOS) value for each video in a representative group, and use the results to validate FB-MOS. We describe the process for the full reference component of FB-MOS in the following.

We selected 25 videos with high original perceptual quality, from various Facebook video products such as Watch,

Gaming, Live, Newsfeed, etc, in various orientations and aspect ratios - portrait (vertical), landscape (horizontal) and square videos. Each source (SRC) video was transcoded in 3 different bitrate/qualities, with all encodings offering reasonable quality, suitable for streaming at their corresponding encoding resolutions, thus creating 4 processed video sequences (PVS) per SRC. A total of 100 videos, including the hidden SRC reference, were thus available for viewing. These videos were then displayed on a typical mobile-sized viewport. It is reasonable to expect that, for a 240p encoding, the perceived quality on a 720p viewport is not great as the scaling makes the quality loss more perceptible. We collected subjective quality scores from 50 experienced content reviewers, who are not video experts, using a single-stimulus continuous scale methodology. The results were then analyzed, aggregated, and fitted against SSIM scores to obtain the best non-linear fit, thus obtaining the corresponding predicted FB-MOS scores which we then correlated against mean opinion scores (MOS) from subjects.

We've found Spearman ranked-order correlation coefficient (SROCC) to be 0.9147, and Pearson linear correlation coefficient (PCC) to be 0.9034. These correlation scores gave us confidence in using FB-MOS for end-to-end quality monitoring, and drive quality improvement.

4. FB-MOS MEASUREMENT AT SCALE

In the FB ecosystem, a popular video could have millions of views, each coming from a different device with a different screen resolution. Further, as described earlier, the view could switch between inline-view and full-screen view during the same viewing session. As a result, even if the same encoding was played for all these views (i.e. no ABR switching), the perceptual quality score for each view would be different. The adaptation of client ABR to network conditions by switching between the different encoding further complicates the quality metrics computation.

The key challenge is on how to compute FB-MOS for each view by accounting for the varying viewport resolution and played encoding at feasible computational complexity.

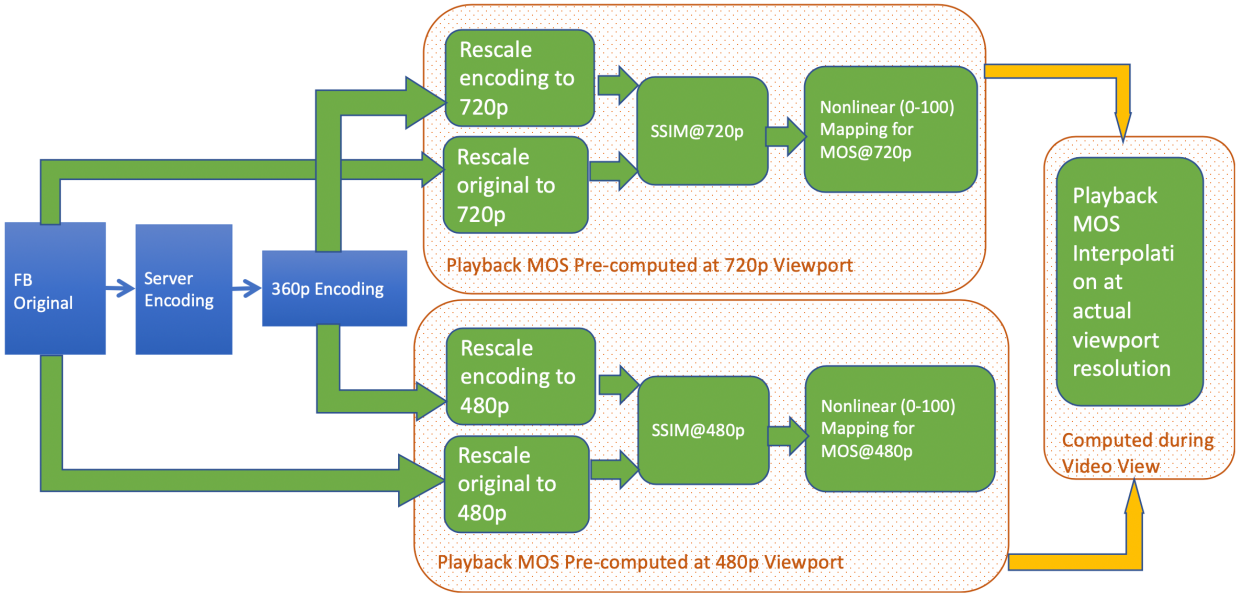
Efficient computation of FB-MOS at scale involves the following steps:

1. Encoding time MOS pre-computation of each encoding at certain fixed viewports
2. View time interpolation of MOS and aggregation.

4.1 MOS Precomputation

For each encoding in the ABR ladder, we compute the full-reference metric for that encoding against the original at a set of fixed viewport resolutions.

As an example, let us assume that the original is a 1080p source, and we produce a 360p encoding as part of the ABR ladder. Let us also assume that the viewport resolutions used for MOS pre-computation are 480p and 720p. In this case, we would first scale the original as well the 360p encoding to 480p resolution and compute the SSIM@480p score. Similarly, we would rescale the original and encodings to 720p resolution and compute the SSIM@720p scores.



Given the higher sensitivity of human eyes to lower spatial frequencies, it is expected that the exact same encoding will look better when rendered at a lower resolution (480p, in our example) and it will look worse at a higher resolution (720p, in this example).

This process is repeated for each resolution in the fixed viewport list as well as for each encoding in the ABR ladder. Thus, we produce and maintain a vector of quality scores for each encoding, estimating the perceived user quality when that encoding is presented at different viewports.

These quality scores are then stored as metadata along with the encoding in the FB video infrastructure datastore.

4.2 View time MOS Interpolation and Aggregation.

When this video is watched on a device, information about its screen resolution is sent to the server. Further, additional information about the time instants when a user switched from inline-view to full-screen view as well as the encoding chosen by the ABR client algorithm are saved by the server for each viewing session.

For example, let us assume that the first 60 seconds of playback involved a 360p encoding being viewed at full-screen (corresponding to a 540p viewport). In this case, we use the pre-computed SSIM scores for this encoding at the closest fixed viewports and interpolate to compute the SSIM at the actual viewport. This SSIM score is then mapped to the linear 0-100 scale as explained previously.

This process of MOS interpolation is performed for any time segment where there is a change in the encoding that was played or the viewport resolution to compute the quality score for that segment. Note that the quality score of each segment can be independently computed without dependency on any other segment.

Further, quality scores for all segments can be aggregated to compute a single quality score for the entire viewing session by either arithmetic or harmonic averaging or by computing other statistics – median or certain percentile.

5. USING FB-MOS FOR QUALITY MONITORING AND IMPROVEMENT

In this section, we discuss how FB-MOS measurement can be used for end-to-end quality monitoring and drive quality improvements.

5.1 End-to-end Quality Monitoring.

First, in addition to using the FB-MOS score (both Playback Score and Upload Score) of each viewing session or each video to identify issues, scores can be aggregated across many dimensions such as the viewing client (iPhone/Android/www) or video Product. Any change in aggregate MOS score provides an early signal into regressions across the entire video processing pipeline.

For example, a change in the no-reference Upload Score values indicates a change in the ingestion part of the pipeline, such as a potential drop in the number of 1080p or 720p uploads. This might also indicate that some low quality uploads have become high popular recently and receiving a lot of watch-time and view-time aggregation helps us identify these issues. For UGC content in particular, the vast majority of uploads are from mobile devices and are generally transcoded on the device prior to upload. Regressions in Upload MOS are therefore usually indicative of potential regression in the upload video quality. The Upload MOS can also be tracked at upload time, however tracking this at view time allows us to compute the watch time weighted impact of any regressions.

Changes in the full reference MOS may indicate regressions in compression efficiency, or more subtle changes in the interaction between the specific lanes being generated and served, the available network bandwidth, and the player ABR decisions. For monitoring we look at both overall watch time weighted FB-MOS, as well as the distribution across specific quality thresholds.

5.2 Quality Improvement and System Troubleshooting using FB-MOS

Full-reference MOS metric is used to track -

- Improvements in compression efficiency by using advanced codecs (such as AV1) or better encoding recipes (such as per-shot convex hull encoding and longer GOPs).
- Improvements in ABR client due to better bandwidth estimation or network protocols.

Videos uploaded to FB ecosystem vary widely in the watch-time distribution that they receive. The variance is based on both the product type, and the popularity of the uploader or the video content. A small percentage of uploads receive the majority of the watch time. We can use this to our advantage by producing encodings with higher compression efficiency (at a higher computational cost) for videos with a high predicted watch time. The CPU cost of x264 encodings at the “slow” preset is about 4x as that of x264 encodings at the very-fast preset, while the compression efficiency of x264 “slow” is ~20% higher. Similarly, the VP9 encodings we produce have ~6x higher CPU complexity as compared to the H264 at slow preset, while also having higher compression efficiency. These levers allow us to make per-video decisions on compute vs. egress, based on the predicted popularity.

To take advantage of the varying computational efficiency and compression efficiency, we have several ABR families including:

- Basic ABR: Encodings are produced with H264 codec "veryfast" preset and relatively smaller number of lanes. These encodings are typically produced for all videos.
- Full ABR: Encodings are produced with H264 codec "slow" preset and higher number of lanes per video. These encodings are produced either when the video product is of higher importance or when the watch-time of the video is higher than a threshold.
- VP9: Encoding are produced with Vp9 codec typically at the slower presets for most lanes. These videos are produced for the most popular content which are watched a lot, or if the video product is very important.

We then use FB-MOS to measure the performance of each of these families from the standpoint of delivered video quality. The performance of each family, in terms of watch time-weighted average FB-MOS, is tracked over time so we can determine both short term (week over week) and longer term (month over month) regressions in a particular family. The families can be compared with one another in order to ensure that the more computationally expensive families continue to provide the expected benefit over their computationally cheaper counterparts. Additionally, the overall performance of the system across all encode families is measured. If the performance of individual families and their relative performance to one another has not changed, then a change in overall delivered video quality may point to a resource shift that is causing the system to deliver less advanced encodings. For example, an increase in video uploads may reduce the available compute capacity for generating VP9 encodings, and a drop in VP9 coverage may move the overall FB-MOS of the system downward.

If there are no changes in encode production, then regressions in FB-MOS may indicate regressions due to delivery time decisions about what encodes to serve, or regressions in client-side playback performance. In order to isolate such effects, we can compare system-average FB-MOS scores over time by client application, and by specific application version. A regression that only affects a specific mobile application version is likely to be a client-side change, whereas a regression in delivered quality that affects, for example, all Android users is more likely to point to a delivery configuration.

In addition to end-to-end monitoring, we can also use FB-MOS to optimize encoding and delivery decisions. Video quality doesn't always offer a linear benefit to the viewer, as quality increases. Rather there are certain thresholds, where quality can be perceived as "unacceptable" vs. "acceptable" and "annoying" vs. "not-annoying", a phenomenon that has been studied by other researchers[7]. In other words, we can expect users to simply stop watching a certain video when its quality drops below a certain threshold; on the other hand, perceived video quality doesn't change much after it exceeds another threshold. By focusing on meeting or exceeding these thresholds, we can guide encoding and delivery improvements.

A focus on reducing the number of sessions with bad quality, according to FB-MOS, has led to a number of encoding optimizations, including increasing GOP size, optimizing resolutions at the low end of the ABR ladder, and optimizing the compute resource allocation across ABR lanes. By using slower presets for the lower bitrate lanes, small increases in compression efficiency may translate into larger reductions in the number of sessions with unacceptable quality, whereas a potential small loss of 1-2% in compression efficiency at 1080p may not meaningfully affect number of sessions with acceptable video quality.

For delivery and playback, we can leverage FB-MOS data to determine which video qualities to play in specific situations. Without any constraints, the player may choose to play the highest available quality. However due to imperfect bandwidth estimates the player needs to determine an acceptable level of risk when making ABR decisions. By taking FB-MOS into consideration, the player can decide whether the increase in quality offsets the risk of rebuffering while playing a higher bitrate lane. This allows the player to be more aggressive when the increase in quality would allow it to cross thresholds, and less aggressive when the marginal increase in quality may not be worth a higher risk of stalls.

In certain situations, we also need to conserve data usage and keep bitrates lower than what the user's available bandwidth would normally be able to support. One example is for mobile data connections where we want to be mindful of total data usage (to avoid rapidly using a user's available monthly data allowance). In other cases, we need to reduce total egress due to ISP capacity constraints, such as the recent Covid-19 related data caps[8]. Other examples include regional capacity constraints due to fiber cuts or loss of CDN capacity.

For these cases we need to limit what qualities are played and optimize the delivered quality per byte consumed. One approach is to limit resolutions, such as restricting playback to standard definition (SD), or reduce bitrates, such as restricting playback to 1 Mbps. However due to video content variations, bitrate and resolution constraints alone offer poor control over the resulting quality distribution, and risks increasing the number of sessions with annoying or unacceptable quality. Instead of relying on resolution and bitrate we can also directly leverage FB-MOS in order to make delivery decisions. By doing this we can create a more consistent experience, allowing higher bitrates for complex content, and cutting deeper on simple content, in order to achieve comparable byte savings with a better overall experience.

6. LIVE VIDEO-SPECIFIC FB-MOS OPTIMIZATION

As we saw in previous sections, computing FB-MOS involves a small, but not-negligible computational overhead. When leveraging FB-MOS for Live video streaming, there are additional challenges due to the necessity of maintaining real-time transcoding of the Live stream at all times. Hence, in the context of Live, the importance of an optimized FB-MOS is not only at scale, but also on a per-video basis.

When adapting and optimizing FB-MOS for Live, we leveraged a few key insights:

- The main computation cost of FB-MOS stems from two operations, scaling and SSIM. Scaling is the dominant contributor to computational cost, especially for larger resolutions.
- FB-MOS computation at any particular viewport involves a scaler operation of both the 'source' and the 'encoded' stream to that viewport and computing the SSIM at that resolution.
- The current implementations of FB-MOS rely only on the Luma (Y) plane, hence there is no value of scaling or performing SSIM on the chroma planes
- Computing FB-MOS at encoded resolution is more efficient, since SSIM can be computed within the encode loop itself
    - This is slightly more optimal due to in-cache-effects of the source and transcoded pixels and benefit of avoiding extra frame memory copies if used within separate FBMOS module
- The different FB-MOS operations have differing computational cost depending on initial and target resolutions and SSIMs.
- The viewer-side metrics aggregation for FB-MOS (as discussed before), relies on a pre-computed set of points that represent the quality-viewport curve. However, the algorithm is robust enough to leverage *any* such subset of points.
- If we reuse the encoded resolutions as viewport resolutions to calculate FB-MOS, we can reduce the number of scaling operations needed, since the source is already available

Based on above insights, we can reduce the computational cost of FBMOS significantly: *when computing FBMOS we set the viewport resolutions to be a subset of the N encoded resolutions.* This implies that we *never* need to scale the source image, since it is already available, as part of generating another ABR lane.

Analysis of computational savings
Computing FB-MOS for each encoded lane at N viewports:

- FB-MOS with no optimizations: (2*N scalers + N SSIM) * 3/2 (where 3/2: overhead of computing both luma/chroma)
- FB-MOS using above approach: N scalers + N SSIM

We implemented the above optimizations in two iterations, nicknamed ‘standard’ and ‘superpvqs’, for solely speed of deployment into production (Note: ‘pvqs’ is just an older in-house term for FB-MOS).

- ‘standard’: Compute FB-MOS for only Luma and reuse in-encode SSIM. Load-balance worker threads to ensure all MOS jobs are similarly sized for compute
- ‘superpvqs’: Compute FB-MOS at viewports that are subset of encoded resolutions and reuse frames wherever possible (i.e. avoid memory copies)

As an illustration of the speed improvement over the stock (no-optimization FBMOS compute), this table contains the offline performance (in seconds) for encoding BigBuckBunny using x264 at veryfast preset at common production CRFs. The encoding time varies depending on encoding preset but the overhead of FBMOS (in absolute time) is fairly consistent.

	No FB-MOS	FB-MOS	FB-MOS - standard	FB-MOS - superpvqs
Real (s)	410	587	526	503
User (s)	1010	1876	1588	1390
System (s)	5	8	7	6
Overhead				
Real	0	177	116	93
User	0	866	578	380
System	0	3	2	1
Overhead %		**86%**	**57%**	**38%**

Using the above optimizations, the real overhead of FB-MOS dropped from ~20% of our Live encoding CPU utilization to a more reasonable ~5-10%. We are further investigating optimizing the overhead in a few ways:

- Subsampling FB-MOS on randomized frames (instead of every frame).
- Compute all viewport SSIMs in-place during encoding process to leverage the effects of in-cache access.
- Investigate the SSIM implementation to identify further consolidation of processing stages across lanes

7. CONCLUSION

Measuring video quality at Facebook scale is a great engineering challenge. Our attempt to address this challenge resulted in the introduction and deployment of a no-reference video quality metric to capture upload video quality and an SSIM-based full-reference metric (FB-MOS), which takes into account the non-linearity and the different viewport resolutions

to offer both encoding video quality and delivery-side video quality figures. In developing FB-MOS, we relied heavily on subjective testing and validation of typical Facebook videos.

Our choices have been dictated by the harsh reality that we are operating on a very tight compute budget and thus we can only afford a small computational overhead, compared to the core video encoding complexity. We have successfully leveraged these video quality metrics to monitor, optimize and troubleshoot the vast Facebook video infrastructure systems and video products, including Live, in our attempt to improve quality of experience for the users of our services.

Most recently, we relied on FB-MOS to provide must-needed relief to network backbone and mobile network operators during the unprecedented global emergency brought by the COVID-19 pandemic, with a minimal impact in perceived quality for our users.

We are continuously improving our video quality metrics, using appropriate subjective validation and gradual deployment in our products and services, while being always mindful of conserving energy in our datacenters. We are also collaborating with our academic research partners in exploring innovative ways to introduce energy-efficient quality metrics in the Facebook video pipeline.

REFERENCES

[1] "Netflixonomics The tech giant everyone is watching", *The Economist,* June 2018.

[2] Recommendation ITU-R BT.500-14 - Methodologies for the subjective assessment of the quality of television images. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-14-201910-I!!PDF-E.pdf (2019).

[3] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M., "Toward a practical perceptual video quality metric," https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652 (2016).

[4] Wang, Z., Bovik, A. C., Sheikh, H.R., and Simoncelli, E.P. "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[5] H. Sheikh and A. Bovik, "Image Information and Visual Quality," IEEE Transactions on Image Processing, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[6] Li, S., Zhang, F., Ma, L., and Ngan, K.N., "Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments," IEEE Transactions on Multimedia, vol. 13, no. 5, pp. 935–949, Oct. 2011.

[7] Li, J., Krasula, L., Baveye, Y., Li., Z., and Callet, P. Le, "AccAnn: A New Subjective Assessment Methodology for Measuring Acceptability and Annoyance of Quality of Experience," in IEEE Transactions on Multimedia, vol. 21, no. 10, pp. 2589-2602, Oct. 2019, doi: 10.1109/TMM.2019.2903722.

[8] Schultz, A., Facebook Newsroom, "Keeping our services stable and reliable during the COVID-19 outbreak", Mar. 24, 2020, <https://about.fb.com/news/2020/03/keeping-our-apps-stable-during-covid-19/>

[9] Drucker, H., Burges, C. C., Kaufman, L., Smola, A. J., and Vapnik, V. N., "Support Vector Regression Machines", in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.