How to Encode Video for the Future

Dror Gill, CTO, Beamr

Abstract

The quality expectations of viewers paired with the ever-increasing shift to over-the-top (OTT) and mobile video consumption, are driving today's networks to be more congested with video than ever before. To counter this congestion, this paper will cover advanced techniques for applying content-adaptive encoding and optimization methods to video workflows while lowering the bitrate of encoded video without compromising quality.

Intended Audience:
Business decision makers
Video encoding engineers

To learn more about optimized content-adaptive encoding, email info@beamr.com

BEAMR

Table of contents

Encoding for the future.

The standard method of encoding video for delivery over the Internet utilizes a pre-set group of resolutions and bitrates known as adaptive bitrate (ABR) sets. Industry guidelines, such as Apple's Tech Note TN2224 [1], define a typical ABR set as a fixed set of bitrates and resolutions for all video files regardless of the content type. In traditional terrestrial, cable, and satellite TV, bandwidth is pre-allocated and fixed. In the world of IP streaming, there is greater flexibility that allows for content-adaptive bitrate schemes to achieve reduced bitrate while maintaining high video quality.

As content owners and video streaming providers look to the future, they are assessing:

- How to balance bitrate and quality
- Different video coding approaches
- Techniques for content-adaptive encoding

Below, we cover approaches that operate with existing video coding standards such as H.264/AVC and H.265/HEVC. There are other approaches that use proprietary video coding schemes and add-ons, but they tend to overcomplicate deployments. The methods explained below range from simple techniques meant for small scale implementations, to more complex methods that need access to a large database of content and/or significant compute power. Lastly, we'll cover a closed-loop process that leverages a perceptually-aligned quality measure.

The trade off between bitrate and quality.

When streaming video over the Internet, video encoding is typically done using an ABR set. Each ABR set of encoding configurations includes resolutions, bitrates, and encoding parameters, which are used to encode a single video title. ABR sets are designed to deliver an optimal range of viewing experiences to connected users over varying network bandwidths and devices.

Block-based video encoding schemes are inherently lossy processes that achieve compression by removing information from the bitstream while taking into account how it will impact the file size and perceived visual quality of the video.

The rate control algorithm adjusts encoder parameters in order to achieve a targeted bitrate. This algorithm allocates a budget of bits to each group of pictures, individual frames, and in some cases sub-frames in a video sequence. The quantization parameter (QP) regulates how much spatial detail is retained:

- When using higher QP values, bitrate is reduced and perceived quality lowered.
- While using lower QP values, the bitrate will be increased along with the subjective quality of the video.

Legacy approaches to encoding your video content.

There are several approaches to encoding video content at scale.

- Constant bitrate encoding
- Variable bitrate encoding
- Constant rate factor encoding
- Capped constant rate factor encoding

Let's discuss the advantages and disadvantages to each approach.

What is constant bitrate (CBR) encoding?

CBR (constant bitrate) rate control produces an output stream with a relatively constant data rate. A highly regulated stream was required in the early days of video delivery by devices which were based on hardware architecture, and did not have the flexibility to support varying data rates.

The disadvantage of CBR is that applying a constant data rate to varying content complexity inherently causes unstable output quality:

- Complex scenes requiring more bits than the target data rate allows will suffer from low quality

- Simpler scenes will be encoded with an unnecessarily high bit budget

What about variable bitrate encoding?

Variable bitrate (VBR) encoding is more efficient than CBR, since it allows the bitrate to vary dynamically around a specified average data rate. The dynamic variations of bitrate in VBR are typically tied to scene complexity, so the available bits are allocated in a more optimal way, resulting in increased quality for complex scenes, and lower bitrate for simpler scenes. However, since the VBR algorithms determine the amount of bits based on the complexity of scenes and not according to a true perceptual quality measure, it is less efficient at adapting to the content than other methods that will be presented below.

Encoding content with constant rate factor encoding.

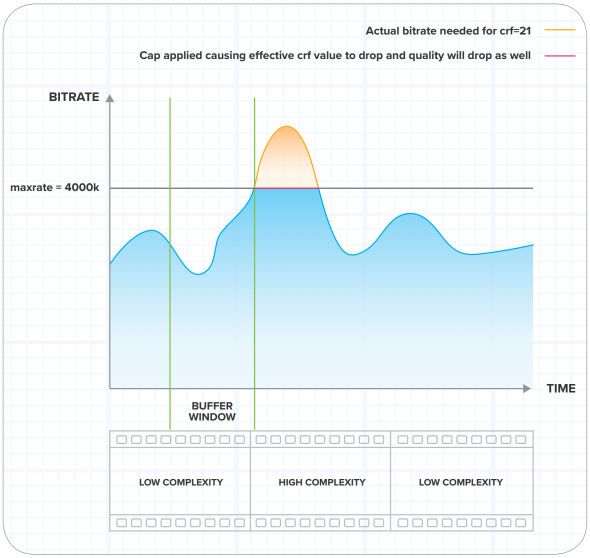
Constant rate factor (CRF) encoding requires workflows to set a desired quality level for encoding a video stream, as opposed to a fixed bitrate. While encoding with CRF, the encoder tries to maintain the desired quality level and reduces bitrate during high motion scenes, taking advantage of the fact that the human eye perceives less loss of detail in a moving scene than in a static scene. Using CRF, the bitrate of the encoded file is not set in advance, and can reach high levels for complex scenes. Another disadvantage of the CRF method is that the same CRF value can yield different levels of perceptual quality across a video library, and even between scenes in the same video.

Capped constant rate factor encoding for high complexity scenes.

Capped CRF encoding is a bitrate control technique which combines CRF encoding with a maximum bitrate "cap." The encoder adjusts the data rate to deliver the specified quality level, but never exceeds the specified maximum bitrate, even in relatively complex streams that might require a bitrate higher than the maximum to obtain the target quality. With lower complexity content, CRF encoding results in a data rate that is lower than the maximum.

Figure 1 illustrates the drawback of the capped CRF approach. In scenes where the bitrate cap is applied, which are typically scenes that require a higher bitrate in order to maintain the subjective visual quality, the bitrate is constrained. This causes a compromise in video quality in sections where the degradation will be the most visible.

FIGURE 1 - CAPPED CRF QUALITY DEGRADATION



The different encoding approaches described above offer tradeoffs between bitrate restriction and quality. Still, the question remains how can we obtain the best or optimal bitrate tradeoff decisions?

Encoding content for the future.

Content-adaptive encoding was engineered to configure video encoders according to the content in video streams, instead of applying predetermined parameters. Using content-adaptive encoding enables us to reach optimal bitrate vs quality trade-offs, allowing for improved quality at similar bitrates, or equivalent quality at a lower bitrate. These solutions help provide better user experience while reducing infrastructure costs. Content-adaptive encoding approaches include:

- Manual content-adaptive encoding per title
- Manual content-adaptive encoding per category
- Manual content-adaptive encoding by the title and chunk
- Content-adaptive encoding using neural networks
- Closed loop content-adaptive encoding by the frame

Manually encoding content by title.

One relatively straightforward method to reduce bitrates according to the needs of the content is to manually customize encoding parameters for each file. This approach encompasses manually experimenting with each video file, seeking parameters which provide the best quality at a given bitrate, or the lowest bitrate possible for a predetermined quality. The process is incredibly time consuming, not scalable, and not very effective because it would be impossible to manually tune the parameters for each section in the video stream, as made possible by some of the methods described below. The manual per-title encoding approach is typically used where volumes are extremely low, but the value of each video title, high. Blu-ray is one example of where this approach is workable.

Manually encoding content by the category.

Category specific content-adaptive encoding works by manually classifying a content library into different content categories, such as movies, talk shows, PowerPoint presentations, etc. Then the engineer finds the optimal encoding parameters and bitrates for each category and encodes each using category-specific encoding parameters.

Jan Ozer in Video Encoding by the Numbers [2] outlined the following steps for implementing category-specific encoding:

- Separate a video repository into distinct categories, and identify 3-5 videos in each category that represents a cross section of the content within that category. These videos will serve as the test set for encoding.

- Encode the videos in the test set using several CRF values. Examine the quality of the resulting videos, and for each video in the set, find the CRF value that delivers acceptable quality at the lowest bitrate, and average the bitrates across the videos in each category. This average bitrate will be the target bitrate for that category.

- Encode the test files at this target rate using either CBR or VBR encoding. Play the files and verify that encoding quality is still acceptable. Test the file with an objective quality metric (such as PSNR), and focus on both the average quality and the lowest quality regions or frames.

- If the quality is acceptable at the rate shown in step 2, build an ABR set for each class of content based on the selected resolution and bitrate.

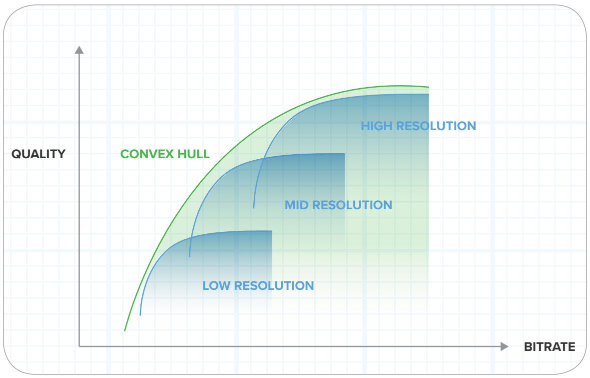
This per-category encoding approach is generally less effective for premium movie and TV shows, since the range of encoding complexity within a class can vary significantly. As an example, though animated movies are typically easier to compress than real world videos, the range of animated techniques is highly diverse, ranging from 2D cell animation and 3D animation to human-like computer graphics. This makes it challenging to create an ABR set that works across all animated content equally. Thus a weakness of the categorized content encoding approach can be seen when there is wide variety inside at least one category.

Another challenge of per-category encoding is the variability between scenes in a single title. For example a sports show may include talking head in-studio shots along with fast action game shots and slow-motion recaps, each requiring different bitrate values to maximize quality while ensuring the minimal bitrate possible.

Content-adaptive encoding by the title and chunk.

Netflix [3] proposed a fully automated approach for title specific encoding optimization, which creates an optimal ABR set for each video title. This technique is based on encoding the content at different CRF levels and resolutions, and then evaluating the quality of the resulting videos using an objective quality measure. Netflix is using the Video Multimethod Assessment Fusion (VMAF) quality metric [5] to evaluate video quality during the analysis stage. The quality scores generate a Convex Hull which identifies the resolution which produces the highest quality encode at each bitrate. An illustration of results is in Figure 2.

FIGURE 2 - CONVEX HULL GENERATED BY NETFLIX'S PER-TITLE ENCODING



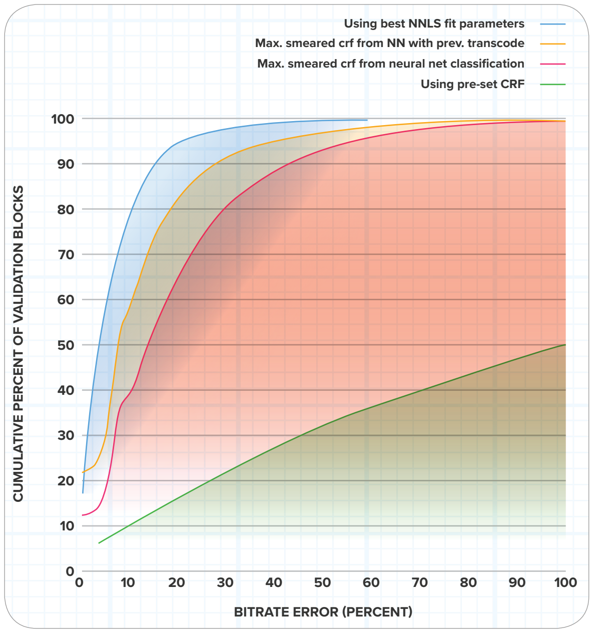
While the scheme in Netflix proposal [3] uses CRF for the trial encodes which determine quality, the final encode is done using 2-pass VBR. This approach is very computationally intensive and requires multiple encodes at different resolutions and bitrates. It is suitable for content libraries that are diverse, but limited in size, such as premium content including TV series and movies.

Content-adaptive encoding using neural networks.

YouTube [4] presented an approach for customizing encoding parameters for user-generated videos uploaded to their platform, by building a machine-learning algorithm that estimates the optimal encoding parameters based on features extracted from a target encode of the video.

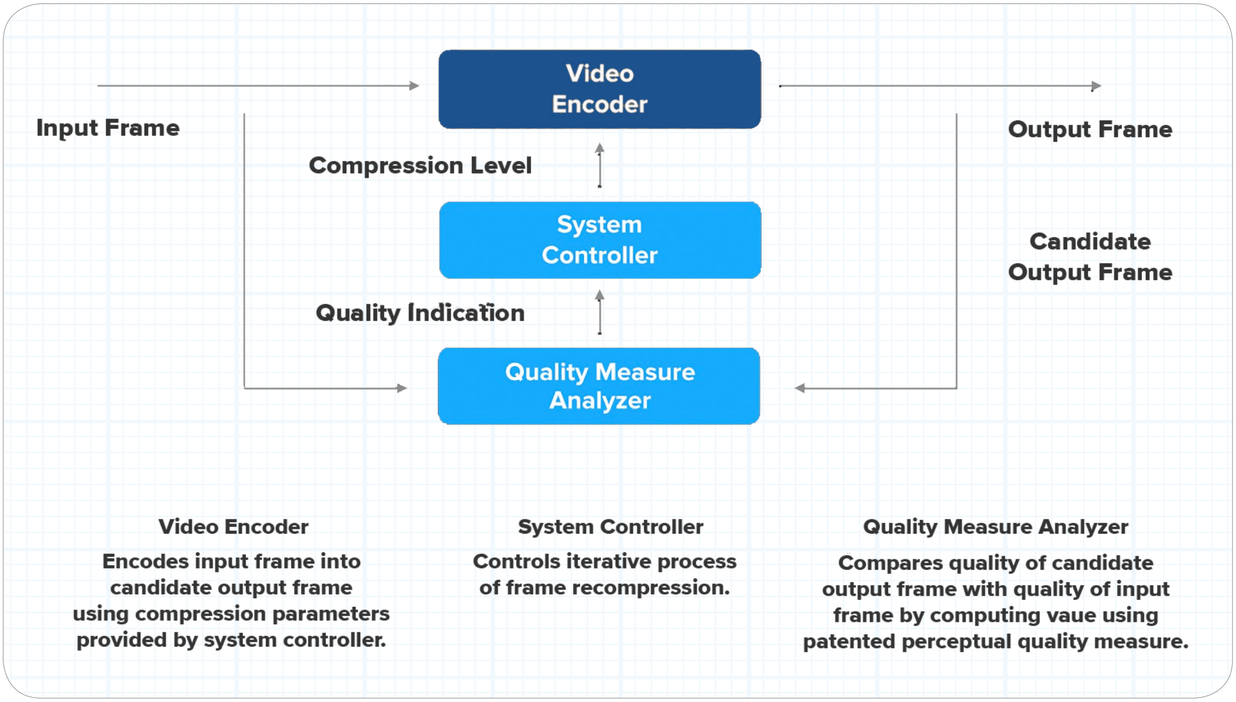
In the proposed technique, a sample of 10,000 videos are encoded using all possible CRF values, and the objective quality of each encode is measured using the SSIM quality measure [6]. A machine-learning algorithm correlates between the optimal CRF value for each clip, and the measurements of each clip including resolution, frame rate, bitrate and motion vectors. By applying this machine- learning algorithm to all the videos in the database, the optimal CRF value can be determined based on the features of the encoded video clip. Selection of the CRF value is improved by adding data from a low-resolution CRF encode of the actual clip to the machine learning algorithms.

FIGURE 3 - YOUTUBE RESULTS - CUMULATIVE DISTRIBUTION OF BITRATE ERRORS (%) ACROSS THE TEST SET



Results of this technique are shown in Figure 3, where the blue line illustrates best fit, the red line is the original result from the neural network, while the orange line illustrates the final result by combining the results of the low resolution CRF encode. The green line represents the baseline using a preset CRF value.

FIGURE 4 - CLOSED LOOP FRAME-LEVEL OPTIMIZATION PROCESS



Closed loop content-adaptive encoding per frame.

Beamr has developed a process for content-adaptive encoding based on closed loop re-encoding of input frames at various compression levels, while checking the value of a proprietary quality measure [8] that has high correlation with human subjective results. The input to the Beamr process is a video file that has been compressed at the quality level desired which serves as the quality reference for the re-encoding process. This process is applied after the initial encode of the video stream, but before packaging for streaming.

As described in Figure 4, the closed loop content-adaptive encoding process consists of the following steps:

1. Decode the source video frame.

2. Re-encode the source video frame using a compression level determined by the system controller, creating a candidate output frame.

3. Obtain the decoded candidate output frame.

4. Compare the decoded candidate output frame to the decoded source frame using an objective perceptual quality measure.

5. If the quality is above a high threshold, indicating that the source frame can be compressed to a lower size without compromising perceptual quality, increase the compression level and return to step 2.

6. If the quality is below a low threshold, indicating that the candidate output frame is not perceptually identical to the input frame, decrease the compression level and return to step 2.

7. If the quality is between the low threshold and the high threshold, output the candidate output frame to the output stream, move to the next source frame, and return to step 1.

By evaluating the quality of the video on a frame-by-frame basis, this method of content adaptive encoding ensures that the results of encoding are optimal - the overall video file is encoded to the lowest possible size, while fully retaining the source quality of each frame.

Results

Figure 5 shows a graph of bitrate as a function of time for a 1080p source video before optimization, and a corresponding output video after optimization. The average bitrate in this case was reduced from 3510 Kbps to 2584 Kbps, while the peak bitrate (marked in green) was reduced from 7593 Kbps to 5598 Kbps. It is also important to note that in sections of the video where the bitrate cannot be further reduced without compromising quality, the source bitrate is retained (marked in red). The result is a video that is visually identical to the source, only smaller.

Table 1 shows the results of Beamr's closed loop frame-level content-adaptive optimization on a set of movie trailers, comprising various content types including animation, action and drama. The results clearly show the different bitrate saving that can be achieved on titles that have different resolutions, bitrates and content. This means where bits per pixel are high, savings will be greater than where the ratio of bits per pixel are low.

FIGURE 5 - BITRATE GRAPH COMPARING ORIGINAL TO OPTIMIZED STREAM

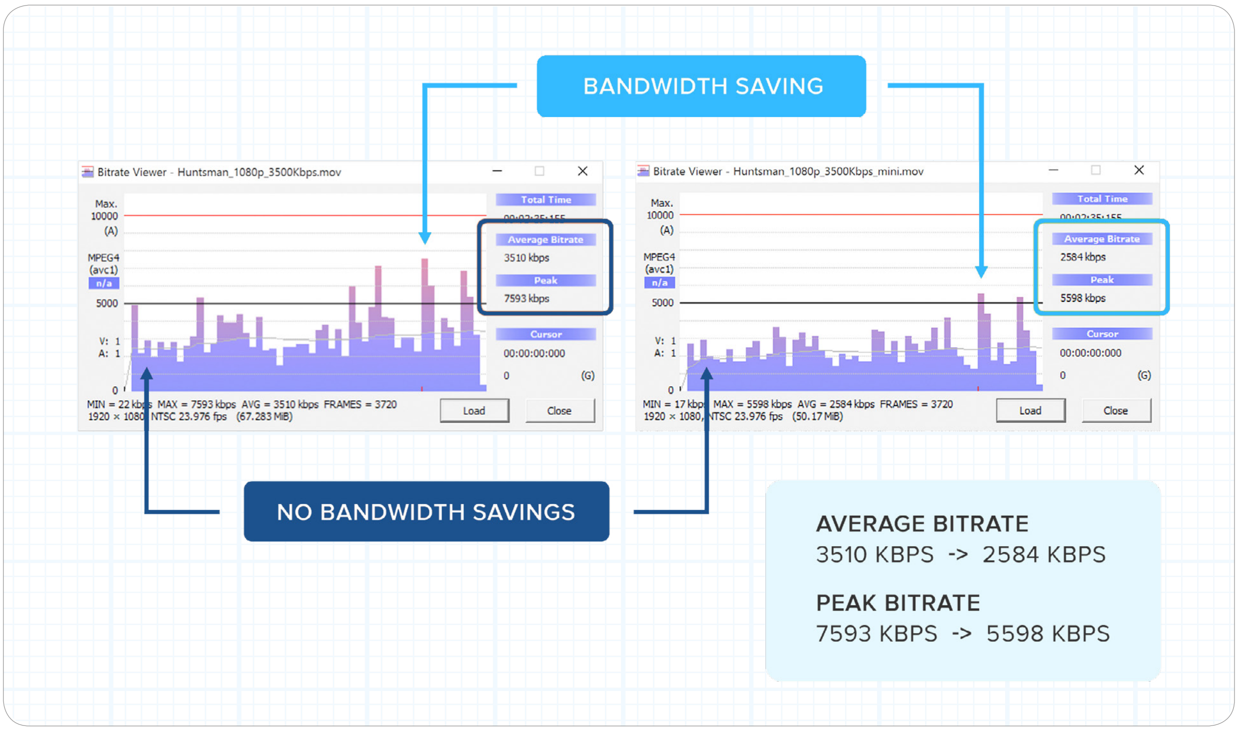


TABLE 1 - RESULTS OF BEAMR'S OPTIMIZATION PROCESS ON A COLLECTION OF MOVIE TRAILERS

Genre	Resolution / Original Bitrate	180p 0.2Mbps	180p 0.3Mbps	288p 0.5Mbps	360p 0.75Mbps	432p 1.2Mbps	720p 2Mbps	1080p 3.5Mbps	1080p 6Mbps
Fantasy	Optimized Bitrate	0.165	0.191	0.353	0.479	0.649	1.275	2.584	3.235
	Savings	18%	36%	29%	36%	46%	36%	26%	46%
Animation	Optimized Bitrate	0.159	0.178	0.303	0.395	0.515	1.065	2.07	2.29
	Savings	21%	41%	39%	47%	57%	47%	41%	62%
Comedy	Optimized Bitrate	0.18	0.222	0.388	0.556	0.78	1.56	2.976	3.988
	Savings	10%	26%	22%	26%	35%	22%	15%	34%
Action	Bitrate	0.18	0.218	0.4	0.537	0.771	1.536	2.936	3.536
	Savings	10%	27%	20%	28%	36%	23%	16%	41%

How should you be encoding your content?

In this paper, we detailed why the industry needs to address the quality-bitrate challenge that is caused by the massive consumer adoption of streaming video, in order to meet user expectations and network/ infrastructure constraints. In this article, we presented the case for content adaptive encoding and the various approaches possible including their advantages and disadvantages. All methods presented are a step-up from the existing preset recipes such as are outlined in [1] with a one size fits all approach.

Of the proposed solutions, the closed loop per-frame content adaptation, when combined with a reliable perceptual quality metric, provides the best solution in terms of scalability and adaptivity to different content types, as well as being the only method that can guarantee no loss of quality compared to the input clip. We believe that this approach will be a driving force in the industry, and will enable reaching optimal quality-bitrate trade-off decisions.

REFERENCES

[1] Technical Note TN2224, “Best Practices for Creating and Deploying HTTP Live Streaming Media for Apple Devices”, Apple Inc., August 2016 http://snip.ly/ev144

[2] Jan Ozer, “Video Encoding by the Numbers: Eliminate the Guesswork from Your Streaming Video”, Doceo Publishing, January 2017 http://snip.ly/gy9vn

[3] Anne Aaron, Zhi Li, Megha Manohara, Jan De Cock and David Ronca, Netflix, “Per-Title Encode Optimization”, The Netflix Tech Blog, December 14th, 2015 http://snip.ly/hh2lq

[4] Michele Covell, Martin Arjovsky, Yao-Chung Lin and Anil Kokaram, “Optimizing transcoder quality targets using a neural network with an embedded bitrate model”, Proceedings of the Conference on Visual Information Processing and Communications 2016, San Francisco http://snip.ly/ys7id

[5] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy and Megha Manohara, “Toward A Practical Perceptual Video Quality Metric”, The Netflix Tech Blog, June 6th, 2016. http://snip.ly/fl1pp

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

[7] The Case for Content-Adaptive Optimization, Beamr, March 2016 http://snip.ly/j08l4

[8] US Patent 9,491,464 “Controlling a video content system by computing a frame quality score” http://snip.ly/laz7t