https://doi.org/10.2352/ISSN.2470-1173.2020.9.IQSP-285

Statistical characterization of tile decoding time of HEVC-encoded 360° video

Henrique D. Garcia[a], Mylène C.Q. Farias[a], Ravi Prakash[b], and Marcelo M. Carvalho[a]
[a] University of Brasília, Brazil, [b] University of Texas at Dallas, USA

Abstract

In this paper, we present a statistical characterization of tile decoding time of 360° *videos encoded via HEVC that considers different tiling patterns and quality levels (i.e., bitrates). In particular, we present results for probability density function estimation of tile decoding time based on a series of experiments carried out over a set of* 360° *videos with different spatial and temporal characteristics. Additionally, we investigate the extent to which tile decoding time is correlated with tile bitrate (at chunk level), so that DASH-based video streaming can make possible use of such an information to infer tile decoding time. The results of this work may help in the design of queueing or control theory-based adaptive bitrate (ABR) algorithms for* 360° *video streaming.*

Introduction

The traffic of virtual reality (VR) and augmented reality (AR) applications is poised to grow considerably in the next few years [1]. Most certainly, the streaming of 360° videos is one of the main drivers for such traffic growth, especially with the proliferation of affordable head-mounted displays (HMD) coupled with the increasing popularity of 360° videos in platforms such as Facebook and YouTube. However, the streaming of 360° videos still faces many challenges to operate seamlessly while delivering high quality of experience to end users under varying network conditions. This is because this type of video demands considerable more network resources compared to standard 2D video streaming. For this reason, intense research activity has been observed in this field in the past few years.

Nowadays, the streaming of 360° videos mostly relies on the system architecture developed for 2D video streaming. Therefore, the whole video sphere needs to be first projected onto a plane using some projection technique (e.g., equirectangular or cubemap) before being encoded with a standard 2D-video encoder. Typically, the video is encoded in several qualities/bitrates before it is segmented into "chunks" of fixed time length. For streaming, the Dynamic Adaptive Streaming over HTTP (DASH) protocol is usually employed, by which the client side of the application can request video chunks from a remote server based on some adaptive bitrate (ABR) algorithm. However, unlike 2D videos, a user can only view a fraction of the whole sphere at any given time through the viewport (or Field of View (FoV)) of an HMD, as shown in Figure 1(a). Typically, an HMD has a FoV of $120° \times 90°$, which corresponds to only 16.6% of the entire sphere [2]. Consequently, significant system resources can be wasted if the whole spherical video is transmitted (a.k.a "monolithic" streaming), especially because such videos are streamed with very high definition.

To save resources (e.g., bandwidth and memory), a number of works [3, 4, 5, 6] have adopted a viewport-adaptive approach to the streaming of 360° videos. In this case, the video projection is spatially segmented into independently decodable *tiles*, so that the client can request only the tiles from a given region of interest (ROI) that encompass the viewports predicted for the next few seconds (or, alternatively, request all tiles under different quality levels). Figure 1(b) depicts a 4×4 tiling pattern with a ROI and corresponding viewport.

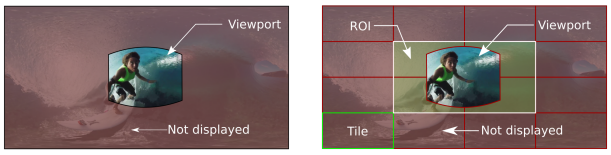


Figure 1. *(a) Viewport in a monolithic streaming: waste of requested and unseen pixels (red area). (b) A* 4×4 *tiling pattern with ROI and viewport.*

Along these lines, Qian et al. [7] have proposed a tile-based solution that targets commodity mobile devices. Their proposed ABR algorithm optimizes the number and size (i.e., quality) of tiles to request based on a constraint that takes into account the *tile decoding time* estimated from sample averages evaluated at runtime, among other parameters. Consequently, the constraint on which an optimal decision is made assumes that a single "typical" value of tile decoding time is valid for all searchable sizes of tiles (i.e., qualities). Such an assumption may render buffer occupancy issues that can lead to undesirable playback stalls. Hence, while the authors' system solution delivers remarkable performance, we believe that their work has also shed some light on the need to investigate tile decoding time in more detail. Indeed, the status of buffer occupancy is key in the design of ABR algorithms [8], and understanding its behavior via queueing or control theory has proven to be a powerful tool [8, 9, 10, 11]. To help on that, some statistical knowledge of input/output traffic behavior of a client's buffer is needed. In fact, while considerable attention in the literature has been given to the characterization of network bandwidth, no previous work has dealt with the characterization of tile decoding time of 360° videos, to the best of our knowledge.

In this paper, we present a statistical characterization of tile decoding time of 360° videos encoded via HEVC that considers different tiling patterns and quality levels (i.e., bitrates). In particular, we present results for probability density function estimation based on a series of experiments carried out over a set of 360° videos with different spatial and temporal characteristics. Additionally, we investigate the extent to which tile decoding time is correlated with tile bitrate (at chunk level), so that DASH-based video streaming can make possible use of such an information to infer tile decoding time.

Experimental Methodology

Figure 2 depicts the workflow adopted in this work to characterize the decoding time of HEVC-encoded 360° video chunks partitioned into different tiling patterns. The details of each step are explained next.

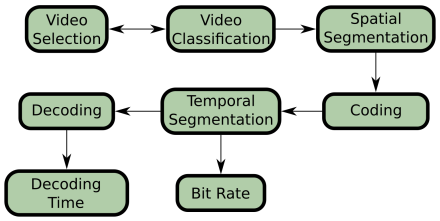


Figure 2. *Workflow for characterization of tile decoding time of 360° videos.*

First, we selected twelve 360° videos from YouTube, all in equirectangular projection[1]. Samples of 60 seconds duration of each video were converted to 4K resolution (4320×2160) at 30 fps. To guarantee a good representation of video content, we chose the samples based on their Spatial Information (SI) and Temporal Information (TI) [12]. Figure 3 shows the dispersion of TI and SI values of the selected videos. The dots indicate the median value, and the bars around the dots are limited by the first and third quartile, computed over all frames. Notice that no videos are represented in the upper-right corner of the graph because it is hard to find videos that lead to high TI an SI values. Probably, videos with high movement (i.e., high TI) tend to blur the scenes, which leads to lower SI values.

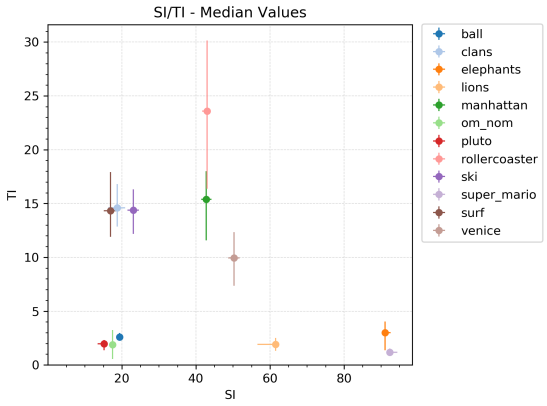


Figure 3. *SI and TI values for the selected videos. Large points indicate the median values, and the endpoints of the bars indicate the quartiles.*

Table 1 summarizes the SI and TI values of the videos, along with their average bitrate when encoded with a constant rate factor (CRF) of 28 (default encoder value). For the selected videos, the TI values have a Pearson correlation of 0.797 with the video bitrates, while the SI values are practically uncorrelated with the video bitrates (correlation of -0.011). Given that we are also interested in investigating whether the tile bitrate is strongly correlated with tile decoding time, knowing that TI has a good correlation with the tile bitrate may help to infer about tile decoding time, if the correlation between these two are also found to be strong.

[1] Available at http://gpds.ene.unb.br/databases/2020-360videos/

Table 1. SI/TI median values and average bitrate of videos.

Video	Rate (Mb/s) CRF 28	SI	TI
om_nom	0.589	17.3	1.85
super_mario	0.804	91.9	1.16
pluto	1.737	14.7	1.82
ball	3.248	18.9	2.48
ski	9.631	22.5	14.2
elephants	11.16	90.5	2.74
lions	12.54	61.2	1.45
venice	13.01	49.1	9.54
manhattan	13.22	41.7	14.9
clans	15.50	17.3	14.3
surf	16.34	15.6	14.0
rollercoaster	19.76	41.8	23.3

For each original uncompressed video in Table 1, we created nine new video sequences with specific frame tiling patterns: 3×2, 4×3, 6×3, 6×4, 6×5, 6×6, and 7×6. This led to videos with 6, 12, 18, 24, 30, 36, and 42 tiles per frame, respectively. For reference purposes, we also considered the 1×1 pattern (no tilling). In terms of number of pixels, the investigated tiling patterns correspond to the following spatial resolutions: 1440×1080, 1080×720, 720×720, 720×540, 720×432, 720×360, and 617×360, respectively. Then, we encoded each tile with the HEVC x265 encoder in FFmpeg[2], using six different values for the Constant Rate Factor (CRF) to control the quality/bitrate: 16, 22, 28, 34, 40, and 46. The lowest (highest) CRF value of 16 (46) corresponds to the highest (lowest) video quality. The default value of 28 is usually recommended for its good trade-off between quality and compression ratio. It is worth noting that each increment of 3 in the CRF value halves the bitrate, i.e., the bitrate decreases fourfold for CRF values in our set.

After encoding each tile, of each video, they were temporally segmented into 1-second chunks that correspond to a whole GOP of 30 frames. This choice resides in the fact that, in order for a chunk to be independently decoded, it must contain a number of frames that is a multiple of a GOP. We chose a 1-second chunk because this is the most likely chunk length that ABR algorithms would pick to work with DASH, considering that head movement prediction works well only within prediction windows of about 1 second [13]. The chunks were then encapsulated into an MP4 file so they could be decoded individually, which caused an overhead of about 100 bytes for an extra MP4 "box moov" header by chunk. Hence, each tile of each video contains 60 decodable chunks under six different qualities, which gives us a total of 725,760 chunks to work with.

Video decoding was performed on a 3.4-GHz i7-4770 desktop computer with 16 GB RAM, running Linux Ubuntu 18.04 using the FFmpeg native decoder with only one thread. The collected time measurements correspond to "user time," also known as "process time," which is the amount of CPU time used by a process on user mode. In other words, it is the time spent on non-kernel operations with decoding operations only. Each chunk of a given tile was decoded three times to obtain an average decoding time representative of that particular tile/chunk (a slight variation in measured decoding time was observed for about 17% of measurements, most probably due to the multitasking nature of the

[2] https://ffmpeg.org/

operating system). Additionally, we measured the bitrate of each chunk, as we are interested in understanding its correlation with tile decoding time. To compute that, each chunk file size (in bits) was divided by its duration (1 second).

For statistical analysis, we used the SciPy library and considered the following possibilities for positive continuous probability distribution fitting (the names in parentheses are used in the graphs): Burr type XII (*burr12*), Birnbaum-Saunders (*fatiguelife*), Gamma (*gamma*), Inverse Gaussian (*invgauss*), Rayleigh (*rayleigh*), Log Normal (*lognorm*), Generalized Pareto (*genpareto*), Pareto (*pareto*), Half-Normal (*halfnorm*), and Exponential (*expon*)[3]. The fitting was based on a maximum likelihood parameter estimation, and the best results are given in terms of root-mean-square error (RMSE). In the results, the parameter values *loc* and *scale* must be applied to the normalized distribution to obtain the fitted one, according to

$$\text{fitted\_pdf}(x) = \left(\frac{1}{scale}\right) \text{normalized\_pdf}\left(\frac{x-loc}{scale}\right). \quad (1)$$

Also, where it is applicable, the parameters c, μ, s, and d refer to specific distribution parameters, as defined in the SciPy library.

Experimental Results

First, we present an analysis of tile decoding time for each tiling pattern, when all videos and quality levels are considered altogether. Then, we look at specific quality levels and tiling patterns. Finally, we present the results for chunk decoding time, i.e., when all tiles of a chunk are decoded sequentially (for instance, to render the whole frames onto the sphere).

Statistics per Tile Pattern with All Quality Levels

Table 2 contains the average and standard deviation values of both tile decoding time and tile bitrate for each tiling pattern, including the correlation coefficient between both measurements. As expected, both tile decoding time and tile bitrate decrease as the tile size decreases (i.e., tile segmentation increases). The magnitude of the standard deviation, with respect to the mean value, increases as the tile size decreases. Such ratio is higher for the bitrate than for the decoding time. It is worth noting that the range of CRF values considered here (16 through 46) implies a 20-fold variation in bitrate. As we can see, the correlation between decoding time and bitrate is not very high, when all quality levels are considered, and it decreases as the tile size decreases.

Table 2. Mean and standard deviation for tile decoding time and tile bitrate by tiling pattern for all qualities.

Tile Pattern	Decoding Time (s)		Bitrate (Mbps)		Corr.
	Mean	Deviation	Mean	Deviation	
1×1	0.795	0.356	9.73	14.0	0.763
3×2	0.130	0.076	1.67	3.02	0.691
4×3	0.061	0.039	0.850	1.58	0.688
6×3	0.039	0.026	0.574	1.07	0.684
6×4	0.030	0.021	0.437	0.859	0.675
6×5	0.024	0.017	0.352	0.698	0.648
6×6	0.021	0.015	0.297	0.595	0.631
7×6	0.018	0.012	0.258	0.513	0.625

[3]https://docs.scipy.org/doc/scipy/reference/stats.html

Figure 4 depicts, for each tiling pattern, the three best probability density functions fitted to the empirical distribution of tile decoding time. The estimated values of distribution parameters are shown in Table 4 in the Appendix. From the results, the Log-Normal distribution is among the three best fitted distributions in 100% of the patterns, while the Inverse Gaussian and Birnbaum-Saunders distributions appears in 87.5% and 75% of the cases, respectively.

Statistics per Quality Level and Tile Pattern

Figure 5 presents the average tile decoding time and average bitrate per quality level, for each tiling pattern. As we can see, the decrease in bitrate as video quality decays is also followed by a decrease in tile decoding time, but it is less pronounced. For instance, in the 6×3 case, while the bitrate decays by 97.5% (1.91 Mbps to 47.6 Kbps) if the CRF varies from 16 to 46, the tile decoding time decreases by only 63.1% (0.067 to 0.025 seconds) for the same CRF range. The fitting parameters for the 48 distributions (6 quality levels and 8 tiling patterns), are given in Tables 5, 6, 7, 8, 9, 10, 11, and 12, shown in the Appendix. From the results, the three best fitted distributions vary by tiling pattern, resulting in the following frequency with which they appear among the top three: Birnbaum-Saunders (79.2%), Inverse Gaussian (72.9%), Log-Normal (70.8%), Burr Type XII (41.7%), Gamma (8.33%, Generalized Pareto (8.3%), Half-Normal (6.3%), Exponential (3.08%), and Rayleigh (2.1%).

Table 3 presents the results for the correlation between tile decoding time and average bitrate for all the 48 cases. For a given tiling pattern, the correlation increases if the quality level increases, while for a given CRF value, the correlation decreases if the number of tiles per frame increases. Considering the tiling pattern of 6×4, the correlation is about 0.728 for CRF = 28 (this tiling pattern was shown to achieve high bandwidth savings [4]).

Table 3. Correlation between tile decoding time and tile bitrate.

Pattern	Quality (CRF)					
	16	22	28	34	40	46
1×1	0.893	0.839	0.803	0.786	0.704	0.554
3×2	0.857	0.781	0.735	0.684	0.610	0.479
4×3	0.846	0.782	0.744	0.674	0.584	0.497
6×3	0.848	0.788	0.734	0.683	0.581	0.471
6×4	0.843	0.789	0.728	0.673	0.566	0.451
6×5	0.827	0.761	0.703	0.636	0.537	0.424
6×6	0.814	0.748	0.687	0.617	0.520	0.403
7×6	0.808	0.743	0.688	0.605	0.505	0.398

Chunk Statistics with All Quality Levels

Now, we consider the time to decode all tiles of a chunk sequentially, considering all videos and quality levels altogether. Figure 6 shows the chunk's average decoding time and average bitrate for each tiling pattern. Error bars indicate the standard deviation. As expected, the bitrate increases as the number of tiles per frame increases, since the segmentation in tiles limits the search space of the encoder's motion prediction. Surprisingly, however, the average chunk decoding time benefits from tiling segmentation, as their average values are generally smaller than for the 1×1 case. In fact, the average decoding time decreases as the number of tiles per frame increases, until it reaches a minimum at the 6×3 pattern that is 10.8% smaller than the 1×1 pattern (at the

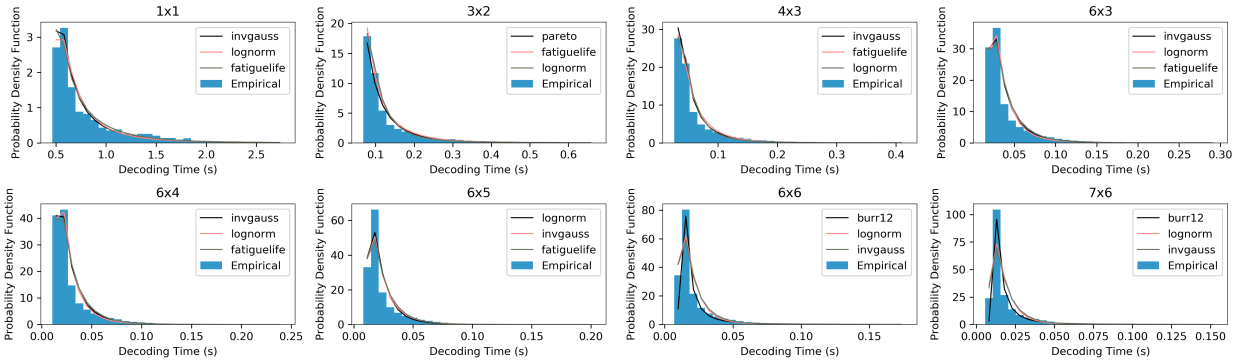


Figure 4. *Probability density functions fitted to measured tile decoding times according to tiling pattern. This case considers video chunks with all quality levels.*

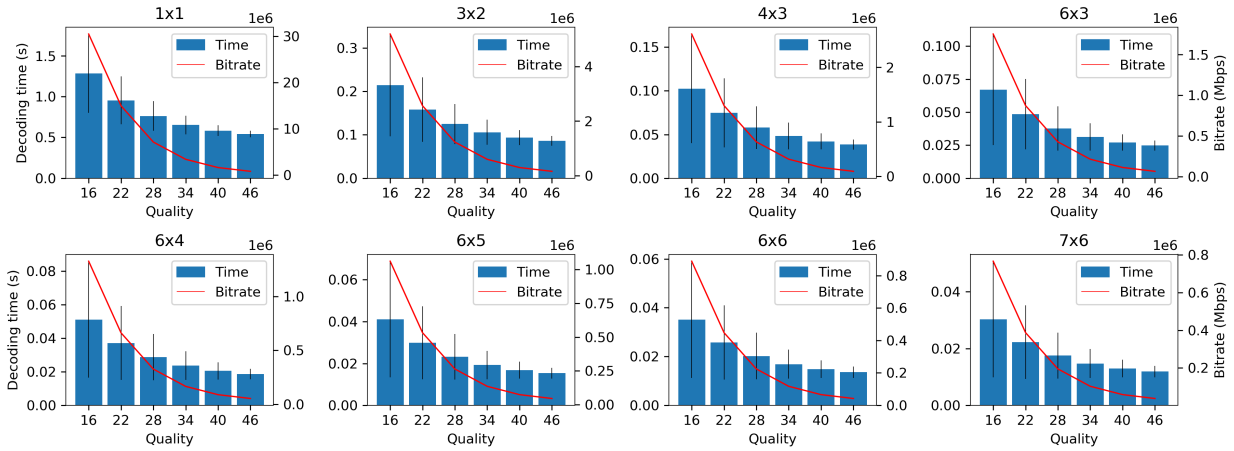


Figure 5. *Tile average decoding time (blue bars) and average bitrate (red lines) according to tiling pattern and tile quality level.*

expense of an increase of just 6.26% in bitrate). From this point on, decoding time values start increasing again. The explanation of this specific behavior deserves further investigation that is out of scope of this work.

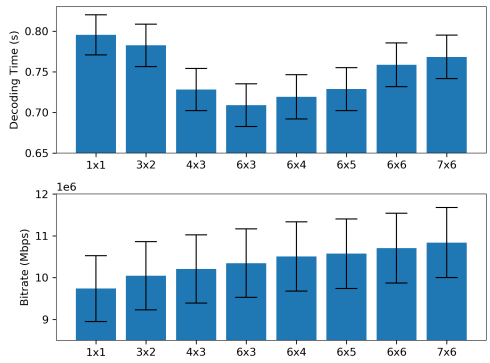


Figure 6. *Chunk decoding time and average bitrate by tiling pattern.*

From our measurements, the chunk's average decoding time is highly correlated with its average bitrate across all tiling patterns: the measured correlation coefficient is above 0.96 in all cases (not shown here due to lack of space). Figure 7 depicts the three best probability distributions fitted to the gathered data for each tiling pattern. The Inverse Gaussian and Birnbaum-Saunders distributions are among the top 3 in 100% of the patterns, while the Log-Normal is among the top 3 in 87.5% of the patterns. The Burr Type XVII distribution is a candidate for the 1×1 case. The values of distribution parameters for this case are not shown due to lack of space.

Conclusions

This paper presented a statistical characterization of tile decoding time of HEVC-encoded 360° videos that considered different tiling patterns over a significant range of video quality levels (i.e., bitrates) and SI/TI properties. From the results, the distributions Log-Normal, Inverse Gaussian, and Birnbaum-Saunders best fitted experimental data in most cases. Such distributions are very flexible and interesting for mathematical modeling purposes (e.g., application to queueing models). Tile decoding time (at chunk level) is strongly correlated with chunk bitrate only if the video quality is high, and the degree of correlation decreases if the number of tiles per frame increases (i.e., high tiling segmentation). The 6×3 tiling pattern delivered the best trade-off between tiled decoding time and average bitrate, while presenting good correlation properties between both metrics if high quality levels are used. Such an information may possibly be used by DASH-based ABR algorithms to infer tile decoding time.

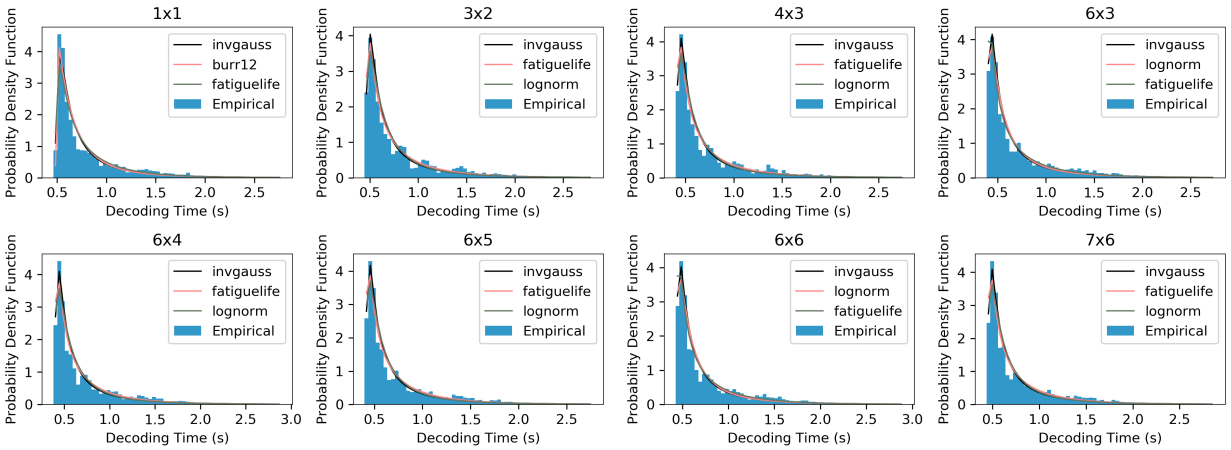


Figure 7. *The histogram of density of the decoding time for all tile pattern and the three best distribution with the minor RMSE, for the complete tiled video.*

Acknowledgments

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Brazil - Finance Code 001).

References

[1] Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," Cisco, White paper, 2018.

[2] S. Afzal, J. Chen, and K. K. Ramakrishnan, "Characterization of 360-degree Videos," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, Los Angeles, CA, Aug 2017, pp. 1–6.

[3] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications," in *Proceedings of the ACM on Multimedia Conference*, Amsterdam, Netherlands, Oct 2016, pp. 601–605.

[4] M. Graf, C. Timmerer, and C. Mueller, "Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP," in *Proceedings of the ACM on Multimedia Systems Conference*, Taipei, Taiwan, Jun 2017, pp. 261–271.

[5] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming based on MPEG-DASH SRD," in *Proceedings of IEEE International Symposium on Multimedia*, San Jose, CA, Jan 2017, pp. 407–408.

[6] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen, "BAS-360°: Exploring Spatial and Temporal Adaptability in 360-degree Videos over HTTP/2," in *Proceedings of the IEEE International Conference on Computer Communications*, Honolulu, HI, Apr 2018, pp. 953–961.

[7] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices," in *Proceedings of the International Conference on Mobile Computing and Networking*, New Deli, India, 2018, pp. 99–114.

[8] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *Proceedings of the ACM conference on SIGCOMM*, Chicago, Illinois, Aug 2014, pp. 187–198.

[9] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proceedings of the ACM Conference on SIGCOMM*, London, UK, Aug 2015, pp. 325–338.

[10] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proceedings of the Annual IEEE International Conference on Computer Communications*, Apr 2016, pp. 1–9.

[11] P. K. Yadav, A. Shafiei, and W. T. Ooi, "QUETRA: A queuing theory approach to DASH rate adaptation," in *Proceedings of the ACM international conference on Multimedia*, Mountain View, CA, Oct 2017, pp. 1130–1138.

[12] ITU-T, "Recomendation P.910 - Subjective video quality assessment methods for multimedia applications." International Telecommunication Union, Geneva, Standard, Apr 2008.

[13] F. Qian, L. Ji, B. Han, V. Gopalakrishnan, L. Ji, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the Workshop on All Things Cellular Operations, Applications and Challenges*, New York, USA, Oct 2016, pp. 1–6.

Appendix

Table 4. Fitted distribution parameters for each tiling pattern.

Format	Distribution	RMSE	Parameters	Loc	Scale
1x1	Inverse Gaussian	0.581	$\mu = 1.577$	0.461	0.212
	Log-Normal	0.660	$s = 1.035$	0.465	0.196
	Birnbaum-Saunders	0.809	$c = 1.115$	0.462	0.207
3x2	Pareto	6.880	$c = 3.108$	-0.072	0.140
	Birnbaum-Saunders	7.322	$c = 1.142$	0.068	0.038
	Log-Normal	7.355	$s = 1.037$	0.068	0.035
4x3	Inverse Gaussian	23.38	$\mu = 1.198$	0.026	0.029
	Birnbaum-Saunders	24.37	$c = 0.983$	0.026	0.024
	Log-Normal	24.87	$s = 0.907$	0.026	0.022
6x3	Inverse Gaussian	67.28	$\mu = 0.891$	0.015	0.028
	Log-Normal	75.70	$s = 0.809$	0.015	0.017
	Birnbaum-Saunders	97.38	$c = 0.867$	0.015	0.019
6x4	Inverse Gaussian	84.10	$\mu = 0.888$	0.011	0.022
	Log-Normal	94.83	$s = 0.804$	0.011	0.013
	Birnbaum-Saunders	124.8	$c = 0.865$	0.011	0.015
6x5	Log-Normal	372.3	$s = 0.750$	0.008	0.012
	Inverse Gaussian	404.2	$\mu = 0.749$	0.008	0.022
	Birnbaum-Saunders	508.4	$c = 0.804$	0.008	0.013
6x6	Burr Type XII	597.1	$c = 8.358$ $d = 0.179$	0.006	0.006
	Log-Normal	638.7	$s = 0.737$	0.007	0.010
	Inverse Gaussian	707.7	$\mu = 0.711$	0.007	0.020
7x6	Burr Type XII	546.9	$c = 14.095$ $d = 0.126$	0.457	0.209
	Log-Normal	1518	$s = 0.695$	0.006	0.009
	Inverse Gaussian	1586	$\mu = 0.624$	0.006	0.020

Table 5. Distribution parameters for 1×1 pattern by quality

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Generalized Pareto	0.350	$c = -0.480$	0.503	1.101
	Half-Normal	0.361	–	0.503	0.919
	Gama	0.365	$a = 2.65E+02$	-6.514	0.029
22	Half-Normal	0.434	–	0.516	0.527
	Exponential	0.476	–	0.516	0.438
	Generalized Pareto	0.483	$c = 1.66E+07$	-7.05E+06	7.05E+06
28	Burr Type XII	0.634	$c = 1.554$ $d = 6.22E+01$	0.483	4.417
	Gama	0.649	$a = 2.001$	0.482	0.140
	Half-Normal	0.658	–	0.484	0.332
34	Burr Type XII	0.737	$c = 9.18E+08$ $d = 0.017$	-2.20E+06	2.20E+06
	Gama	0.879	$a = 1.671$	0.495	0.094
	Birnbaum-Saunders	0.884	$c = 0.748$	0.477	0.136
40	Burr Type XII	0.971	$c = 5.47E+08$ $d = 0.046$	-2.00E+06	2.00E+06
	Birnbaum-Saunders	1.098	$c = 0.554$	0.459	0.107
	Inverse Gaussian	1.108	$\mu = 0.313$	0.457	0.401
46	Burr Type XII	1.260	$c = 5.81E+08$ $d = 0.152$	-3.83E+06	3.83E+06
	Inverse Gaussian	1.483	$\mu = 0.190$	0.448	0.494
	Birnbaum-Saunders	1.484	$c = 0.432$	0.449	0.085

Table 6. Distribution parameters for 3×2 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Birnbaum-Saunders	0.854	$c = 1.080$	0.069	0.091
	Log Normal	0.972	$s = 1.035$	0.071	0.092
	Inverse Gaussian	0.980	$\mu = 1.146$	0.063	0.132
22	Birnbaum-Saunders	1.395	$c = 1.043$	0.069	0.057
	Inverse Gaussian	1.441	$\mu = 1.290$	0.068	0.070
	Log Normal	1.578	$s = 0.996$	0.070	0.057
28	Inverse Gaussian	1.887	$\mu = 1.112$	0.069	0.051
	Birnbaum-Saunders	1.923	$c = 0.982$	0.069	0.037
	Log Normal	2.129	$s = 0.949$	0.070	0.037
34	Burr Type XII	2.473	$c = 4.42E+06$ $d = 0.047$	-6.16E+03	6.16E+03
	Inverse Gaussian	2.501	$\mu = 0.805$	0.068	0.046
	Birnbaum-Saunders	2.617	$c = 0.837$	0.069	0.028
40	Inverse Gaussian	4.301	$\mu = 0.488$	0.067	0.053
	Birnbaum-Saunders	4.527	$c = 0.667$	0.067	0.021
	Log Normal	4.553	$s = 0.660$	0.068	0.020
46	Log Normal	6.111	$s = 0.576$	0.068	0.015
	Inverse Gaussian	6.154	$\mu = 0.341$	0.067	0.055
	Birnbaum-Saunders	6.363	$c = 0.569$	0.067	0.016

Table 7. Distribution parameters for 4×3 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Burr Type XII	1.145	$c = 0.996$ $d = 5.75E+07$	0.030	4.49E+06
	Exponential	1.151	–	0.030	0.072
	Generalized Pareto	1.252	$c = 2.510$	-0.105	0.135
22	Birnbaum-Saunders	1.855	$c = 1.046$	0.028	0.030
	Inverse Gaussian	2.095	$\mu = 1.196$	0.027	0.040
	Log Normal	2.139	$s = 1.003$	0.029	0.030
28	Birnbaum-Saunders	2.940	$c = 0.886$	0.027	0.022
	Inverse Gaussian	2.960	$\mu = 0.907$	0.027	0.034
	Log Normal	3.481	$s = 0.852$	0.028	0.022
34	Burr Type XII	2.432	$c = 1.15E+02$ $d = 0.071$	-0.084	0.117
	Inverse Gaussian	3.396	$\mu = 0.613$	0.027	0.035
	Birnbaum-Saunders	3.526	$c = 0.744$	0.027	0.017
40	Inverse Gaussian	4.191	$\mu = 0.380$	0.027	0.039
	Log Normal	4.255	$s = 0.604$	0.028	0.012
	Birnbaum-Saunders	4.428	$c = 0.600$	0.027	0.012
46	Log Normal	6.185	$s = 0.416$	0.026	0.012
	Inverse Gaussian	6.449	$\mu = 0.171$	0.025	0.078
	Birnbaum-Saunders	6.613	$c = 0.407$	0.025	0.012

Table 8. Distribution parameters for 6×3 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Exponential	1.597	–	0.018	0.049
	Generalized Pareto	1.607	$c = 6.11E+02$	-2.93E+01	2.94E+01
	Generalized Pareto	1.661	$c = -0.181$	0.018	0.058
22	Birnbaum-Saunders	3.341	$c = 0.902$	0.015	0.024
	Log Normal	3.832	$s = 0.865$	0.016	0.023
	Gama	4.742	$a = 1.582$	0.016	0.021
28	Birnbaum-Saunders	4.401	$c = 0.891$	0.017	0.015
	Inverse Gaussian	4.438	$\mu = 0.880$	0.017	0.024
	Log Normal	4.962	$s = 0.869$	0.017	0.014
34	Inverse Gaussian	5.989	$\mu = 0.408$	0.015	0.040
	Birnbaum-Saunders	6.299	$c = 0.613$	0.015	0.014
	Log Normal	6.434	$s = 0.604$	0.015	0.013
40	Log Normal	5.642	$s = 0.543$	0.017	0.009
	Inverse Gaussian	5.725	$\mu = 0.303$	0.016	0.036
	Birnbaum-Saunders	6.073	$c = 0.537$	0.016	0.010
46	Burr Type XII	3.900	$c = 6.081$ $d = 0.660$	0.015	0.008
	Log Normal	7.804	$s = 0.334$	0.014	0.010
	Inverse Gaussian	8.261	$\mu = 0.108$	0.014	0.101

Table 9. Distribution parameters for 6×4 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Birnbaum-Saunders	1.652	$c = 1.066$	0.012	0.025
	Inverse Gaussian	1.825	$\mu = 1.293$	0.012	0.031
	Log Normal	1.917	$s = 1.004$	0.013	0.025
22	Birnbaum-Saunders	3.121	$c = 0.999$	0.012	0.016
	Log Normal	3.490	$s = 0.945$	0.013	0.016
	Burr Type XII	4.207	$c = 1.504$ $d = 1.698$	0.013	0.027
28	Burr Type XII	4.689	$c = 4.13E+02$ $d = 0.040$	-0.191	0.206
	Inverse Gaussian	5.291	$\mu = 0.858$	0.012	0.019
	Birnbaum-Saunders	5.713	$c = 0.864$	0.012	0.012
34	Burr Type XII	8.025	$c = 6.76E+03$ $d = 0.144$	-7.585	7.601
	Inverse Gaussian	8.758	$\mu = 0.435$	0.011	0.029
	Log Normal	9.199	$s = 0.617$	0.011	0.010
40	Inverse Gaussian	10.45	$\mu = 0.265$	0.011	0.035
	Birnbaum-Saunders	10.87	$c = 0.501$	0.011	0.008
	Burr Type XII	18.98	$c = 3.85E+08$ $d = 0.614$	-9.17E+05	9.17E+05
46	Log Normal	10.00	$s = 0.342$	0.010	0.008
	Inverse Gaussian	10.76	$\mu = 0.114$	0.010	0.076
	Birnbaum-Saunders	10.95	$c = 0.333$	0.010	0.008

Table 10. Distribution parameters for 6×5 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Birnbaum-Saunders	2.314	$c = 1.062$	0.010	0.020
	Log Normal	2.656	$s = 1.004$	0.010	0.020
	Exponential	2.752	–	0.010	0.031
22	Burr Type XII	4.619	$c = 1.46E+03$ $d = 0.027$	-0.655	0.668
	Birnbaum-Saunders	5.521	$c = 0.851$	0.008	0.016
	Log Normal	6.106	$s = 0.816$	0.009	0.015
28	Inverse Gaussian	7.967	$\mu = 0.548$	0.008	0.028
	Birnbaum-Saunders	8.316	$c = 0.701$	0.008	0.012
	Burr Type XII	9.747	$c = 5.24E+08$ $d = 0.191$	-1.03E+06	1.03E+06
34	Burr Type XII	5.805	$c = 3.43E+02$ $d = 0.105$	-0.215	0.228
	Inverse Gaussian	9.599	$\mu = 0.363$	0.008	0.030
	Log Normal	9.913	$s = 0.573$	0.009	0.009
40	Burr Type XII	4.650	$c = 4.46E+02$ $d = 0.165$	-0.267	0.280
	Log Normal	11.71	$s = 0.407$	0.008	0.008
	Inverse Gaussian	12.23	$\mu = 0.167$	0.008	0.056
46	Log Normal	10.71	$s = 0.294$	0.007	0.008
	Inverse Gaussian	11.32	$\mu = 0.081$	0.007	0.106
	Birnbaum-Saunders	11.42	$c = 0.283$	0.007	0.008

Table 11. Distribution parameters for 6×6 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Inverse Gaussian	2.991	$\mu = 1.107$	0.008	0.025
	Birnbaum-Saunders	3.105	$c = 0.959$	0.008	0.019
	Log Normal	3.533	$s = 0.908$	0.008	0.019
22	Birnbaum-Saunders	5.657	$c = 0.798$	0.007	0.014
	Log Normal	6.151	$s = 0.771$	0.007	0.014
	Burr Type XII	7.617	$c = 1.842$ $d = 1.578$	0.007	0.020
28	Burr Type XII	6.987	$c = 2.76E+02$ $d = 0.055$	-0.119	0.130
	Inverse Gaussian	7.920	$\mu = 0.621$	0.008	0.020
	Birnbaum-Saunders	8.373	$c = 0.743$	0.008	0.010
34	Log Normal	13.64	$s = 0.528$	0.007	0.009
	Birnbaum-Saunders	13.93	$c = 0.533$	0.007	0.009
	Rayleigh	20.68	–	0.007	0.008
40	Log Normal	13.42	$s = 0.405$	0.007	0.007
	Inverse Gaussian	13.92	$\mu = 0.165$	0.007	0.050
	Birnbaum-Saunders	14.21	$c = 0.399$	0.007	0.008
46	Burr Type XII	8.363	$c = 1.44E+08$ $d = 0.326$	-9.47E+04	9.47E+04
	Log Normal	11.68	$s = 0.294$	0.006	0.007
	Inverse Gaussian	12.41	$\mu = 0.081$	0.006	0.095

Table 12. Distribution parameters for 7×6 pattern by quality.

CRF	Distribution	RMSE	Parameters	Loc	Scale
16	Inverse Gaussian	3.249	$\mu = 1.175$	0.007	0.020
	Birnbaum-Saunders	3.302	$c = 0.991$	0.007	0.016
	Log Normal	3.764	$s = 0.935$	0.007	0.015
22	Inverse Gaussian	6.717	$\mu = 0.886$	0.007	0.017
	Birnbaum-Saunders	7.237	$c = 0.868$	0.007	0.011
	Log Normal	7.921	$s = 0.829$	0.007	0.011
28	Inverse Gaussian	11.12	$\mu = 0.468$	0.006	0.025
	Birnbaum-Saunders	11.58	$c = 0.651$	0.006	0.010
	Log Normal	11.60	$s = 0.635$	0.006	0.009
34	Burr Type XII	11.56	$c = 2.18E+08$ $d = 0.147$	-1.52E+05	1.52E+05
	Log Normal	16.15	$s = 0.509$	0.006	0.008
	Birnbaum-Saunders	16.45	$c = 0.513$	0.006	0.008
40	Log Normal	16.36	$s = 0.435$	0.006	0.006
	Inverse Gaussian	17.13	$\mu = 0.188$	0.006	0.036
	Birnbaum-Saunders	17.48	$c = 0.426$	0.006	0.006
46	Log Normal	22.61	$s = 0.272$	0.005	0.007
	Inverse Gaussian	23.15	$\mu = 0.070$	0.005	0.104
	Birnbaum-Saunders	23.20	$c = 0.262$	0.005	0.007

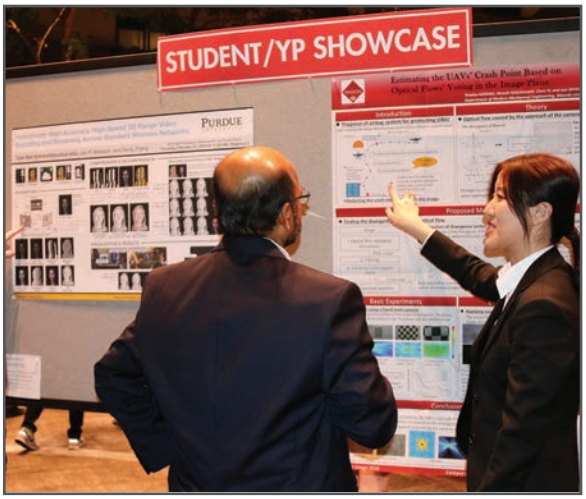
JOIN US AT THE NEXT EI!

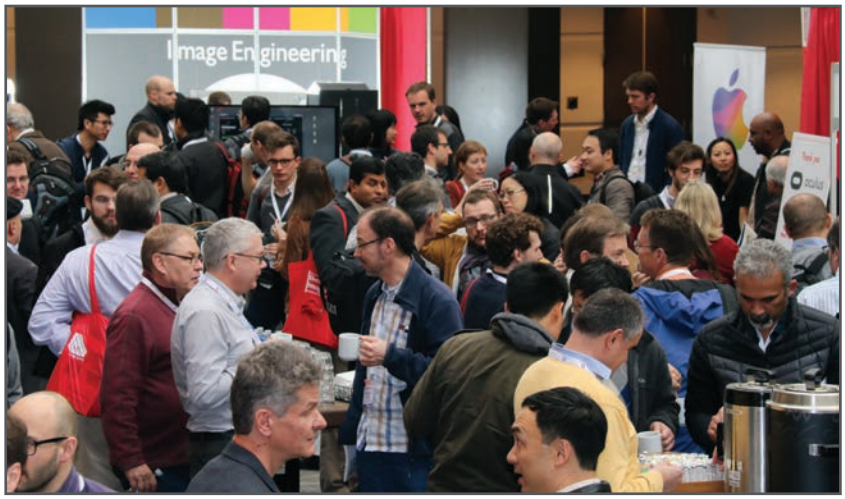
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org