Traffic and Quality Characterization of Scalable Encoded Video: A Large-Scale Trace-Based Study

Part 1: Overview and Definitions [*][†]

Martin Reisslein[‡] Jeremy Lassetter Sampath Ratnam Osama Lotfallah
Frank H.P. Fitzek[§] Sethuraman Panchanathan
`http://www.eas.asu.edu/trace`

First Posted: June 2002
Revised: December 2002

Abstract

The Internet of the future and next generation wireless systems are expected to carry to a large extent video of heterogeneous quality and video that is scalable encoded (into multiple layers). However, due to a lack of long traces of heterogeneous and scalable encoded video, most video networking studies are currently conducted with traces of single layer (non–scalable) encoded video. In this technical report we present a publicly available library of traces of heterogeneous and scalable encoded video. The traces have been generated from over 15 videos of one hour each, which have been encoded into two layers using the temporal scalability and spatial scalability modes of MPEG–4. We provide both the frame sizes as well as the frame qualities (PSNR values) in the traces. We study the statistical characteristics of the traces, including their long–range–dependence and multi–fractal properties.

Keywords: Long Range Dependence; Multi–Fractal; Quality Statistics; Spatial Scalability; Temporal Scalability; Traffic Statistics; Video Traces;

1 Introduction

Video data is expected to account for a large portion of the traffic in the Internet of the future and next generation wireless systems. For the transport over networks, video is typically

---

[*] Supported in part by the National Science Foundation under Grant No. Career ANI-0133252 and Grant No. ANI-0136774. Supported in part by the State of Arizona through the IT301 initiative. Supported in part by a matching grant and a special pricing grant from Sun Microsystems.

[†] Please direct correspondence to M. Reisslein.

[‡] M. Reisslein, J. Lassetter, S. Ratnam, O. Lotfallah, and S. Panchanathan are with the Telecommunications Research Center, Dept. of Electrical Engineering, Arizona State University, Goldwater Center, MC 7206, Tempe AZ 85287–7206, Phone: (480)965–8593, Fax: (480)965–8325, (email: `{reisslein, jeremy.lassetter, sampath.ratnam, osama.lotfallah, panch}@asu.edu`, web: `http://www.eas.asu.edu/trace`).

[§] F. Fitzek is with acticom GmbH, Am Borsigturm 42, 13507 Berlin, Germany Phone: +49–30–4303–2510, Fax: +49–30–4303–2519, (email: `fitzek@acticom.de`, web: `http://www.acticom.de`).

encoded (i.e., compressed) to reduce the bandwidth requirements. Even compressed video, however, requires large bandwidths of the order of several hundred kbps or Mbps. In addition, compressed video streams typically exhibit highly variable bit rates (VBR) as well as long range dependence (LRD) properties. This, in conjunction with the stringent Quality of Service (QoS) requirements (loss and delay) of video traffic, makes the transport of video traffic over communication networks a challenging problem. As a consequence, in the last decade the networking research community has witnessed an explosion in the research on all aspects of video transport. The characteristics of video traffic, video traffic modeling, as well as protocols and mechanisms for the efficient transport of video streams have received a great deal of attention in the networking literature. The vast majority of this literature has considered single–layer MPEG–1 encoded video at a fixed quality level.

The video carried over the Internet of the future and the next generation wireless systems, however, is expected to be different from the extensively studied single–layer MPEG–1 video in several aspects. First, future networks will carry video coded using a wide variety of encoding schemes, such as H.263, MPEG–2, MPEG–4, and so on. Secondly, future networks will carry video of different quality levels, such as video coded with different spatial resolutions and/or signal to noise ratio (SNR). Thirdly, and perhaps most importantly, the video carried in future networks will be to a large extent scalable encoded video. Scalable encoded video will dominate because it facilitates heterogeneous multimedia services over heterogeneous wireline and wireless networks.

The fact that most existing video networking studies are restricted to video encoded into a single–layer (at a fixed quality level) using MPEG–1, is to a large degree due to the lack of traces of videos encoded with different encoders at different quality levels as well as the lack of traces of scalable encoded video. As a first step towards filling the need for a comprehensive video trace library we have generated traces of videos encoded at different quality levels as well as of videos encoded using the temporal and spatial scalability modes.

The traces have been generated from over 15 videos of one hour each. We have encoded the videos into two layers, i.e., a base layer and an enhancement layer, using the temporal scalability mode as well as the spatial scalability mode of MPEG–4. The base layer of the considered temporal scalable encoding gives a basic video quality by providing a frame rate of 10 frames per second. Adding the enhancement layer improves the video quality by providing the (original) frame rate of 30 frames per second. With the considered spatial scalable encoding, the base layer provides video frames that are one fourth of the original size (at the original frame rate), i.e., the number of pixels in the video frames is cut in half in both the horizontal and vertical direction. (These quarter size frames can be upsampled to give a coarse grained video with the original size.) Adding the enhancement layer to the base layer gives the video frames in the original size (format).

For each video and scalability mode we have generated traces for videos encoded without rate control and for videos encoded with rate control. For the encodings without rate control we keep the quantization parameters fixed, which produces nearly constant quality video (for both the base layer and the aggregate (base + enhancement layer) stream, respectively) but highly variable video traffic. For the encodings with rate control we employ the TM5 rate control, which strives to keep the bitrate around a target bit rate by varying the quantization parameters, and thus the video quality. We apply rate control only to the base layer of scalable encodings and encode the enhancement layer with fixed quantization parameters. Thus, the bit rate of the base layer is close to a constant bit rate, while the bit rate of the enhancement layer is highly variable. This approach is motivated by networking schemes that provide constant bit rate transport with very stringent quality of service for the base layer and variable bit rate transport with less stringent quality of service for the enhancement layer.

We also note that we have encoded all videos into a single–layer (non–scalable) for different sets of quantization parameters to obtain non–scalable encodings for different quality levels.

1.1 Organization

This technical report is organized into four parts as follows.

Part 1 gives an overview of the work and describes the generation and structure of the video traces. The video traffic metrics and the video quality metrics used for the statistical analysis of the generated traces are also defined in Part 1.

Part 2 gives the analysis of the video traffic and the video quality of the single layer (non–scalable) encoded video.

Part 3 gives the analysis of the traffic and the video quality of the temporal scalable encoded video. Both the base layer traffic as well as the enhancement layer traffic are analyzed. Also, the video quality provided by the base layer as well as the aggregate (base layer + enhancement layer) stream are studied.

Part 4 studies the video traffic as well as the video quality of the spatial scalable encoded video.

1.2 Related Work

Video traces of MPEG–1 encoded video have been generated and studied by Garret [1], Rose [2], Krunz *et al.* [3], and Feng [4]. These traces provide the size of each encoded video frame, and are therefore typically referred to as frame size traces. The studied frame size traces correspond to videos encoded with MPEG–1 with fixed sets of quantization parameters (i.e., without rate control) into a single layer.

Frame size traces of single–layer MPEG–4 and H.263 encoded video have been generated and studied by Fitzek and Reisslein [5]. Both traces of videos encoded without rate control and of videos encoded with rate control have been generated and studied. Also, different sets of quantization parameters for the encodings without rate control and different target bit rates for the encodings with rate control and thus different levels of video quality are considered.

Our work differs from the existing works on video traces in two fundamental aspects. First, we provide traces of scalable encoded video, i.e., videos encoded into a base layer and an enhancement layer, whereas the existing trace libraries provide only single layer (non–scalable) encoded videos. Secondly, we have broadened the notion of video traces by including not only the sizes of the individual video frames, but also the qualities (PSNR values) of the video frames. Our video traces thus allow for quantitative networking studies that involve the video traffic as well as the video quality.

We also note that studies of the video traffic (bit rate) in conjunction with the video quality (distortion) are very common in the video encoding (compression) field where the encoders are typically characterized by their rate–distortion performance [6, 7]. However, these studies are usually conducted with the publicly available MPEG test sequences (e.g., “foreman”, “coast guard”, etc), which are only 10 seconds (300 frames) in length and include one or two scenes. The rate–distortion characteristics collected for these relatively short sequences, however, are not suitable for typical networking studies. Networking studies typically require *long* sequences that extend over tens of minutes (several 10,000 frames) and include several distinct scenes. This is because the long range dependence phenomena and the rare event phenomena studied by networking researchers can only be observed with statistical confidence from long traces.

2 Video Trace Generation

In this section we describe the generation and the structure of the video traces. We first give a general overview of our experimental set–up and discuss the studied video sequences. We then discuss the different studied types of encoding, including the specific settings of the encoder parameters. Finally, we describe the structures of the video traces and define the quantities recorded in the traces.

2.1 Overview and Capturing of Video Sequences

Our experimental set–up is illustrated in Figure 1. We played each of the studied video

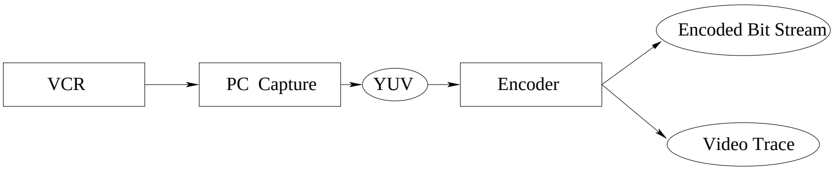

Figure 1: Overview of trace generation.

sequences (see Table 1 for an overview) from a VHS tape using a video cassette recorder (VCR). We captured the (uncompressed) YUV information using a PC video capture card and the `bttvgrab` (version 0.15.10) software [8]. We stored the YUV information on hard disk. We grabbed the YUV information at the National Television Standards Committee (NTSC) frame rate of 30 frames per second. We captured all studied video sequences in the QCIF (176x144 pels) format. In addition we captured some selected video sequences in the CIF (352x288 pels) format. All the video capturing was done with 4:2:0 chrominance subsampling and quantization into 8 bits. We note that the video capture was conducted on a high performance system (dual Intel Pentium III 933 MHz processors with 1 GB RAM and 18 GByte high–speed SCSI hard disc) and that `bttvgrab` is a high–quality video capture software. To avoid frame drops due to buffer build–up when capturing long video sequences we captured the 60 minute (108,000 frame) QCIF sequences in two segments of 30 minutes (54,000 frames) each. With this strategy we did not experience any frame drops when capturing video in the QCIF format. As noted in Table 1, we did experience a few frame drops, when capturing video in the larger CIF format. In order to have a full half hour (54,000 frames) of digital CIF video for our encoding experiments and statistical analyses we filled the gaps by duplicating the video frame preceding the dropped frame(s). We believe that the introduced error is negligible since the total number of dropped frames is small compared to the 54,000 frames in half an hour of video and the number of consecutive frame drops is typically less than 10–20.

We note that in the QCIF format with 4:2:0 chroma subsampling there are 176x144 + 2 · 88x72 = 38,016 pels per frame. With 8 bit quantization and 30 frames per second the bit rate of uncompressed QCIF video is 38,016 pels/frame · 8 bit/pel · 30 frames/sec = 9,123,840 bit/sec. The file size of 1 hour of uncompressed QCIF video is 4,105,728,000 Byte.

In the CIF format with 4:2:0 chroma subsampling, there are 352x288 + 2 · 176x144 = 152,064 pels per frame. The corresponding bit rate — with 8 bit quantization and 30 frames per second — is 152,064 pels/frame · 8 bit/pel · 30 frames/sec = 36,495,360 bit/sec. Because of the larger bit rate of the CIF video format, we restricted the length of the CIF format to 30 minutes. The size of the YUV file for 30 minutes of CIF video is 8,211,456,000 Byte.

The studied videos (see Table 1) cover a wide range of genres and include action movies, cartoons, sports, a variety of TV shows, as well as lecture videos[1]. Covering a wide range of video genres with a large variety in the semantic video content is important since the video traffic (and quality) characteristics typically depend strongly on the video content. To allow for a study of the effect of commercials on the traffic and quality characteristics of encoded video, we captured the *Basketball* video sequence and the talk shows sequences both with and without commercials. These videos were broadcasted with commercials (and recorded with one VCR). To obtain the commercial free sequence, a second VCR was used, which was manually paused during commercials. We acknowledge that this is a crude approach of extracting the commercials, but believe that this approach gives a reasonable approximation.

We note that all the other sports sequences (i.e., *Baseball*, *Football*, *Golf*, and *Snowboarding*) include commercials, as does the *Music* sequence. The *PBS News* sequence is commercial free. We also note that for all the movies and cartoons we commenced the video capture at the start of the feature presentation. (We did not include any previews, trailers, or commercials preceding the feature presentation.)

The lecture sequences are broadcast quality videos produced by ASU's Distance Learning Technology (DLT) department. These videos typically feature a head shot of the instructor lecturing to the class, or the instruction's hand writing on a writing pad or the blackboard.

2.2 Encoding Modes

In this section we describe in detail the studied types of video encoding (compression). All encodings were conducted with the Microsoft version of the MPEG–4 reference (software) encoder [9], which has been standardized by MPEG in Part5 — Reference Software of the standard. Using this standardized reference encoder, we study several different types of encodings which are controlled by the parameters of the encoder. We refer to a particular type of encoding as *encoding mode*. The studied encoding modes are illustrated in Figure 2. The three main categories of studied encoding modes are single–layer (non–scalable) encoding, temporal scalable encoding, and spatial scalable encoding. All studied encoding modes have in common that the number of video objects is set to one, i.e., we do not study object segmentation. We also note that we do not employ reversible variable length coding (RVLC), which achieves increased error resilience at the expense of slightly smaller compression ratios. We found that in the reference software RVLC is currently implemented only for single–layer encodings (as well as for the base layer of scalable encodings). To allow for a comparison of the traffic and

---

[1] To avoid any conflict with copyright laws, we emphasize that all image processing, encoding, and analysis was done for scientific purposes. The encoded video sequences have no audio stream and are not publicly available. We make only the frame size traces available to researchers.

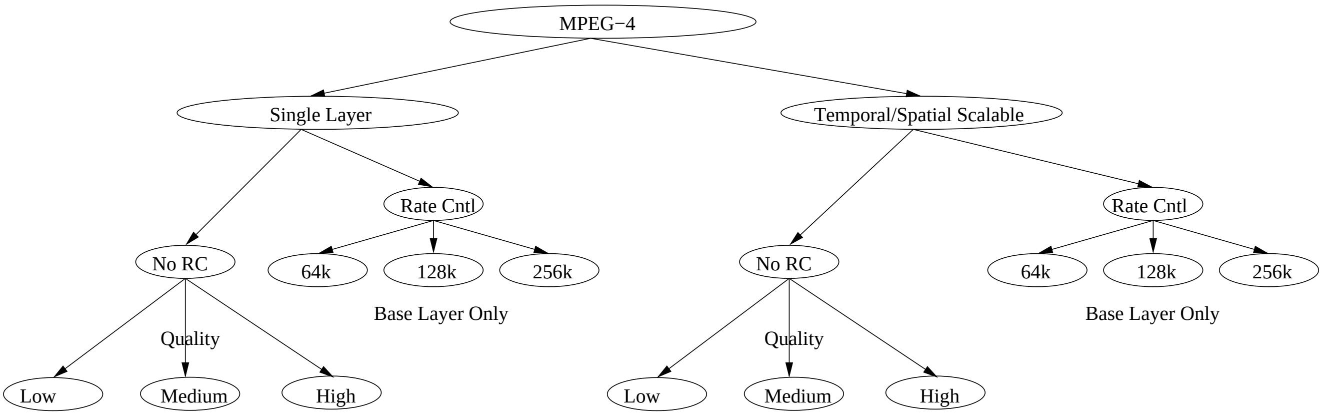

Figure 2: Overview of encoding modes.

quality characteristics of scalable encodings we conduct all encodings without RVLC. For similar reasons we consistently use the decoded frames (rather than the YUV source) for motion estimation (by setting `Motion.Use.Source.For.ME.Enable[0]` = 0). Also, throughout we employ the H.263 quantization matrix.

2.2.1 Single–Layer Encoding

The Group of Pictures (GoP) pattern for single layer encodings is set to IBBPBBPBBPBBIBBP. . ., i.e., there are 3 P frames between successive I frames and 2 B frames between successive P (I) frames. We conduct single–layer encodings both without rate control and with rate control. For the encodings without rate control, the quantization parameters are fixed throughout the encoding. We consider the five quality levels defined in Table 2.

The encodings with rate control employ the TM5 rate control scheme [10], which adjusts the quantization parameters on a macro block basis. We conduct encodings with the target bit rates 64 kbps, 128 kbps, and 256kbps.

2.2.2 Temporal Scalable Encoding

In the considered temporal scalable encoding type the I and P frames constitute the base layer while the B frames constitute the enhancement layer. We note that encoding types with different assignments of frames to the layers are possible (and are supported by the reference encoder). We chose the I and P frames in base layer, B frames is enhancement layer type to fix ideas. In this type the allocation of traffic to base layer and enhancement layer in controlled by varying the number of B frames between successive I(P) and P(I) frames. We initially conduct encodings with two B frames between successive I(P) and P(I) frames (i.e., in the MPEG terminology we set the source sampling rate to three for the base layer and to one for the enhancement layer). We again conduct encodings without rate control and with rate control. For the encodings without rate control we use the fixed sets of quantization parameter settings defined in Table 2. Note that with the adopted scalable encoding types, the quantization parameters of the I and P frames determine the size (in bits) and the quality of the frames in the base layer while the quantization parameter of the B frame determines the size and quality of the enhancement layer frames.

For the temporal scalable encodings with rate control we use the TM5 scheme to control the bit rate of the base layer to a prespecified target bit rate (64 kbps, 128 kbps, and 256 kbps are used). The B frames in the enhancement layer are open–loop encoded (i.e., without rate control); throughout we set the quantization parameter to 16 (which corresponds to the medium quality level; see Table 2). The temporal scalable encodings are conducted both for video in the QCIF format and for video in the CIF format.

2.2.3 Spatial Scalable Encoding

In our study on spatial scalable encoding we focus on video in the CIF format. Every encoded video frame has a base layer component and an enhancement layer component. Decoding the base layer gives the video in the QCIF format, whereas decoding both layers gives the video in the CIF format. We note that the base layer QCIF video may be up–sampled and displayed in the CIF format; this up–sampling results in a coarse–grained, low-quality CIF format video. For the spatial scalable encoding we set the GoP structure for the base layer to IBBPBBPBBPBBIBB.... The corresponding GoP structure for the enhancement layer is PBBBBBBBBBBBPBB..., where by the convention of spatial scalable encodings, each P frame in the enhancement layer is encoded with respect to the corresponding I frame in the base layer and each B frame in the enhancement layer is encoded with respect to the corresponding P frame in the base layer. Each P frame in the base layer is forward predicted from the preceding I(P) frame.

For the spatial scalable encoding without rate control the quantization parameters of the different frame types (I, P, and B) are fixed according to the quality levels defined in Table 2. For the encodings with rate control we use the TM5 scheme to keep the bitrate of the base layer at a prespecified target bitrate of 64 kbps, 128kbps, or 256kbps. The quantization parameters of the enhancement layer frames are fixed at the settings for the defined medium quality level (10 for I frames, 14 for P frames, and 16 for B frames).

2.3 Structure and Generation of Video Traces

In this section we describe the structure of the generated video traces. We first give an overview of the video trace structures and define the quantities recorded in the traces. We then discuss the trace structures for single–layer encoding, temporal scalable encoding, and spatial scalable encoding in detail. We also discuss how the quantities recorded in the traces were obtained for each of the three encoding types.

2.3.1 Overview

Let N denote the number of video frames in a given trace. Let $t_n,\ n = 0, \ldots, N-1$, denote the frame period (display time) of frame n. Let $T_n,\ n = 1, \ldots, N$, denote the cumulative display time up to (and including) frame $n-1$, i.e., $T_n = \sum_{k=0}^{n-1} t_k$ (and define $T_0 = 0$). Let X_n, $n = 0, \ldots, N-1$, denote the frame size (number of bit) of the encoded (compressed) video frame frame n. Let $Q_n^Y,\ n = 0, \ldots, N-1$, denote the quality (in terms of the Peak Signal to Noise Ratio (PSNR)) of the luminance component of the encoded (and subsequently decoded) video frame n (in dB). Similarly, let Q_n^U and $Q_n^V,\ n = 0, \ldots, N-1$, denote the qualities of the two chrominance components hue (U) and saturation (V) of the encoded video frame n (in dB).

We generate two types of video traces: *verbose* traces and *terse* traces. The verbose traces give the following quantities (in this order): frame number n, cummulative display time T_n, frame type (I, P, or B), frame size X_n (in bit), luminance quality Q_n^Y (in dB), hue quality Q_n^U (in dB), and saturation quality Q_n^V (in dB). These quantities are given in ASCII format with one video frame per line. Recall that in our single–layer (non–scalable) encodings and our temporal scalable encodings we use the GoP pattern with 3 P frames between 2 successive I frames, and 2 B frames between successive (I)P and P(I) frames. With this GoP pattern the decoder needs both the preceding I (or P) frame and the succeeding P (or I) frame for decoding a B frame. Therefore, the encoder emits the frames in the order IPBBPBBPBBPBBIBBP.... We also arrange the frames in this order in the verbose trace file. Note that due to this ordering, line 0 of the verbose trace gives the characteristics of frame number $n = 0$, line 1 gives frame number $n = 3$, lines 2 and 3 give frames 1 and 2, line 4 gives frame 6, lines 5 and 6 give frames 4 and 5, and so on.

In the terse traces, on the other hand, the video frames are ordered in strictly increasing frame numbers. Specifically, line $n,\ n = 0, \ldots, N-1$, of a given terse trace gives the frame size X_n and the luminance quality Q_n^Y.

We remark that for simplicity we do not provide the cummulative display time of frame number $N-1$, which would result in an additional line number N in the trace. We also note that for our encodings with spatial scalability, which use the GoP pattern with 11 P frames between successive I frames and no bi–directionally predicted (B) frames, the frames are ordered in strictly increasing order of the frame numbers in both the verbose and the terse trace files.

For the two–layer encodings with temporal and spatial scalability we generate verbose and terse traces for both the base layer and the enhancement layer. The base layer traces give the sizes and the PSNR values for the (decoded) base layer (see Sections 2.3.3 and 2.3.4 for details). The enhancement layer traces give the sizes of the encoded video frames in the enhancement layer and the *improvement* in the PSNR quality obtained by adding the enhancement layer to the base layer (i.e, the difference in quality between the aggregate (base + enhancement layer) video stream and base layer video stream). In summary, the base layer traces give the traffic and quality of the base layer video stream. The enhancement layer traces give the enhancement layer traffic and the quality improvement obtained by adding the enhancement layer to the base layer.

2.3.2 Trace Generation for Single–Layer Encoding

The frame sizes and frame qualities for the single–layer encoding are obtained directly from the software encoder. During the encoding the MPEG–4 encoding software computes internally the frame sizes and the PSNR values for the Y, U, and V components. We have augumented the encoding software such that it writes this data along with the frame numbers and frame types to a verbose trace. We have verified the accuracy of the internal computation of the frame sizes and the PSNR values by the software encoder. To verify the accuracy of the frame size computation we compared the sum of the frame sizes in the trace with the file size (in bit) of the encoded video (bit stream). We found that the file size of the encoded video is typically on the order of 100 Byte larger than the sum of the framesizes. This discrepancy is due to some MPEG–4 system headers, which are not captured in the frame sizes written to the trace. Given that the filesize of the encoded video is on the order of several Mbytes and that individual encoded frames are typically on the order of several kbytes, this discrepancy is negligible. To verify the accuracy of the PSNR computation we decoded the encoded video and computed the PSNR by comparing the original (uncompressed) video frames with the encoded and subsequently decoded video frames. We found that the PSNR values computed for the Y, U, and V components internally perfectly match the PSNR values obtained by comparing original and decoded video frames.

We note that the employed MPEG–4 software encoder is limited to encoding segments with a YUV file size no larger than about 2 GBytes. Therefore, we encoded the $108,000$ frame

QCIF sequences in two segments of 54,000 frames (4500 GoPs with 12 frames per GOP) each and the 54,000 CIF sequences in four segments of 13,500 frames each. The verbose traces for the individual segments were merged to obtain the 108,000 QCIF frame trace and the 54,000 CIF frame trace. When encoding the 4500th GoP of a segment, the last two B frames of the 4500 GOP are bi–directionally predicted from the third P frame of the 4500th GOP and the I frame of the 4501th GoP. Since the 4501th GoP is not encoded in the same run as the preceding GoPs, our traces were missing the last two B frames in a $54,000$ frame segment. To fix this we inserted two B frames at the end of each segment of 53,998 (actually encoded) frames. We set the size of the inserted B frames to the average size of the actually encoded B frames in the 4500th GoP. We belive that this procedure results in a negligible error.

We finally note that the terse traces are obtained from the verbose traces.

2.3.3 Trace Generation for Temporal Scalable Encoding

The frame size of both the encoded video frames in the base layer (I and P frames with the adopted encoding modes, see Section 2.2) and the encoded video frames in the enhancement layer (B frames) are obtained from the frame sizes computed internally in the encoder. Note that the base layer traces (both verbose and terse traces) give the sizes of the frames in the base layer and contain zero for a frame in the enhancement layer. The enhancement layer traces, on the other hand, give the sizes of the frames in the enhancement layer and contain zero for a frame in the base layer. Formally, we let $X_n^b,\ n = 0, \ldots, N-1$, denote the frame sizes in the base layer stream, and let $X_n^e,\ n = 0, \ldots, N-1$, denote the frame sizes in the enhancement layer stream. The video frame qualities (PSNR values) for the base layer, which we denote by $Q_n^{b,Y}$, $Q_n^{b,U}$, and $Q_n^{b,V},\ n = 0, \ldots, N-1$, are determined as follows. The qualities of frames that are in the base layer (I and P frames with our settings) are obtained by comparing the decoded base layer frames with the corresponding original (uncompressed) video frames. To determine the qualities of the frame in the enhancement layer, which are missing in the base layer, we adopt a simple interpolation policy (which is typically used in rate–distortion studies, see, e.g., [11]) With this interpolation policy, the "gaps" in the base layer are filled by repeating the last (decoded) base layer frame, that is, the base layer stream $I_1 \_ \_ P_1 \_ \_ P_2 \_ \_ P_3 \_ \_ I_2 \_ \_ P_4 \ldots$ is interpolated to $I_1\ I_1\ I_1\ P_1\ P_1\ P_1\ P_2\ P_2\ P_2\ P_3\ P_3\ P_3\ I_2\ I_2\ I_2\ P_4\ P_4\ P_4 \ldots$. The base layer PSNR values are then obtained by comparing this interpolated decoded frame sequence with the original YUV frame sequence. The improvements in the video quality (PSNR) achieved by adding the enhancement layer, which we denote by $Q_n^{e,Y}$, $Q_n^{e,U}$, and $Q_n^{e,V},\ n = 0, \ldots, N-1$, are determined as follows. For the base layer frames, which correspond to "gaps" in the enhancement layer, there is no improvement when adding the enhancement layer. Consequently, for the base layer frames, zeros are recorded for the quality improvement of the Y, U, and V components in the enhancement layer trace.

To determine the quality improvement for the enhancement layer frames, we obtain the PSNR of the aggregate (base + enhancement layer) stream from the encoder. We then record the differences between these PSNR values and the corresponding $Q_n^{b,Y}$, $Q_n^{b,U}$, and $Q_n^{b,V}$ values in the enhancement layer trace.

2.3.4 Trace Generation for Spatial Scalable Encoding

With spatial scalable encoding each encoded frame has both a base layer component and an enhancement layer component. We let X_n^b and X_n^e, $n = 0, \ldots, N-1$, denote the sizes (in bit) of the base layer component and the enhancement layer component of frame n. Both components are obtained from the framesizes computed internally by the encoder. The verbose base layer trace gives two different qualities for each video frame, these are the QCIF qualities $Q_n^{b,\mathrm{qcif},Y}$, $Q_n^{b,\mathrm{qcif},U}$, and $Q_n^{b,\mathrm{qcif},V}$ as well as the CIF qualities $Q_n^{b,\mathrm{cif},Y}$, $Q_n^{b,\mathrm{cif},U}$, and $Q_n^{b,\mathrm{cif},V}$. The QCIF qualities are obtained by comparing the decoded base layer stream with the downsampled (from CIF to QCIF) original video stream. The CIF qualities are obtained as follows. The base layer stream is decoded and upsampled (from QCIF to CIF). This CIF video stream is then compared with the original CIF video stream to obtain the CIF qualities. The terse base layer trace gives only the sizes (in bit) of the base layer component X_n^b and the luminance CIF quality $Q_n^{b,\mathrm{cif},Y}$ for each frame n, $n = 0, \ldots, N-1$.

The verbose enhancement layer trace gives the $Q_n^{b,Y}$, $Q_n^{b,U}$, and $Q_n^{b,V}$, $n = 0, \ldots, N-1$, the quality improvements achieved through the enhancement layer with respect to the base layer CIF qualities. These quality improvements are obtained as follows. The aggregate video stream is decoded (CIF format) and compared with the original CIF format video stream to obtain the PSNR values of the aggregate stream. The quality improvements are then obtained by subtracting the base layer CIF qualities $Q_n^{b,\mathrm{cif},Y}$, $Q_n^{b,\mathrm{cif},U}$, and $Q_n^{b,\mathrm{cif},V}$ from the corresponding PSNR values of the aggregate stream.

3 Navigation of Video Trace Website

In this section we give instructions for navigating the video trace website (as well as the video trace CDROM). Our focus is mainly on the `Trace File and Statistics` page for a given video, as the navigation of the other parts of the site is self–explanatory. The `Trace File and Statistics` page is used to navigate to the different encoding modes illustrated in Figure 2 for a given video. This navigation is organized into a tree structure. The tree is rooted at the name of the video, then branches out over several levels (which are discussed in detail below).

The leaves of the tree are the `view` buttons on the right, which link to the page for a particular encoding mode. (The `view` buttons are also duplicated on the left, for convenience.)

Proceeding from left to right we now explain the different levels where the tree branches.

`Format` The format level distinguishes the different video frame formats (dimensions), such as QCIF, CIF. For now, all single–layer (non–scalable) and temporal scalable encodings are in the QCIF format and all spatial scalable encodings are in the CIF format. Thus, there is for now no branching of the tree at this level.

`Scalab.` The scalability level distinguishes single–layer (non–scalable) encoding, temporal scalable encoding, and spatial scalable encoding.

`GoP` The GoP structure level distinguishes different GoP structures. For now, all single–layer (non–scalable) encodings and all temporal, and spatial scalable encodings have the IBBPBBPBBPBBIBBP… structure. Thus, for now, there is no branching of the tree at this level.

`RC` The rate control level distinguishes between encodings without rate control (i.e., rate control is `off`) and encodings with rate control (i.e., rate control is `on`).

`QL` This level distinguishes between the different quality levels (sets of quantization parameter settings) for encodings without rate control and the different target bit rates for encodings with rate control. For encodings without rate control the mappings from the digits 1,…, 5 to the quality levels (and quantization parameters) are given in Table 2, in particular, `1` corresponds to low quality, `3` corresponds to medium quality, and `5` corresponds to high quality. For the eoncodings with rate control, `1` corresponds to a target bit rate of 64 kbps, `2` to a target bit rate of 128 kbps, and `3` to a target bit rate of 256 kbps. Note that for single layer (non–scalable) encodings the target bit rate is for the single layer stream, whereas for scalable encodings the target bit rate is for the base layer.

`Layer` The layer level distinguishes the different encoding layers. For single layer (non–scalable) encodings there is no branching at this level. For scalable encodings we distinguish the base layer (`base`), the enhancement layer (`enh.`), and the aggregate (base + enhancement layer) (`agg.`) stream.

`Smooth.` The smoothing level distinguishes different levels of frame smoothing for temporal scalable encoded video, which has ”gaps” in the individual layers. For single layer encoded video and for spatial scalable encoded video there is no branching at this level. For the base layer of temporal scalable encoded video we distinguish no smoothing (which we denote here by `zero`) and smoothing over three frames i.e., the I (or P) frame and the subsequent two frame gaps (which we denote here by `one`). For the enhancement layer of temporal scalable encoded video we distingiush no smoothing (denoted by `zero`), two–frame smoothing as defined in Part 3 (denoted here by `one`), and three–frame smoothing (denoted here by `two`).

`Metric` The metric level distinguishes the frame sizes, the GoP sizes, and the quality level (PSNR).

A Appendix: Video Traffic Metrics

In this appendix we review the statistical definitions and methods used in the analysis of the generated frame size traces, we refer the interested reader to [12, 13] for details. Recall that N denotes the number of frames in a given trace. Also recall that $X_n,\ n = 0, \ldots, N-1$, denotes the size of frame n in bit.

Mean, Coefficient of Variation, and Autocorrelation

The (arithmetic) sample mean $\bar{X}$ of a frame size trace is estimated as

$$\bar{X} = \frac{1}{N} \sum_{n=0}^{N-1} X_n. \tag{1}$$

The sample variance S_X^2 of a frame size trace is estimated as

$$S_X^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} (X_n - \bar{X})^2. \tag{2}$$

A computationally more convenient expression for S_X^2 is

$$S_X^2 = \frac{1}{N-1} \left[\sum_{n=0}^{N-1} X_n^2 - \frac{1}{N} \left(\sum_{n=0}^{N-1} X_n \right)^2 \right]. \tag{3}$$

The coefficient of variation CoV_X of the frame size trace is defined as

$$CoV_X = \frac{S_X}{\bar{X}}. \tag{4}$$

The maximum frame size $X_{\max}$ is defined as

$$X_{\max} = \max_{0 \le n \le N-1} X_n. \tag{5}$$

The autocorrelation coefficient $\rho_X(k)$ for lag $k,\ k = 0, 1, \ldots, N-1$, is estimated as

$$\rho_X(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} \frac{(X_n - \bar{X})(X_{n+k} - \bar{X})}{S_X^2}. \tag{6}$$

We define the aggregated frame size trace with aggregation level a as

$$X_n^{(a)} = \frac{1}{a} \sum_{j=na}^{(n+1)a-1} X_j, \quad \text{for} \ \ n = 0, \ldots, N/a - 1, \tag{7}$$

i.e., the aggregate frame size trace is obtained by averaging the original frame size trace $X_n,\ n = 0, \ldots, N-1$, over non–overlapping blocks of length a.

We define the GoP size trace as

$$Y_m = \sum_{n=mG}^{(m+1)G-1} X_n, \quad \text{for} \ \ m = 0, \ldots, N/G - 1, \tag{8}$$

where G denotes the number of frames in a GoP (where typically $G = 12$). Note that $Y_m = G \cdot X_n^{(G)}$

Variance–Time Test

The variance time plot [14, 15, 16] is obtained by plotting the normalized variance of the aggregated trace $S_X^{2(a)}/S_X^2$ as a function of the aggregation level ("time") a in a log–log plot, as detailed in Table 3. Traces without long range dependence eventually (for large a) decrease linearly with a slope of -1 in the variance time plot. Traces with long range dependence, on the other hand, eventually decrease linearly with a flatter slope, i.e., a slope larger than -1. We consider aggregation levels that are multiples of the GoP size (12 frames) to avoid the effect of the intra–GoP correlations. For reference purposes we plot a line with slope -1 starting at the origin. For the estimation of the Hurst parameter we estimate the slope of the linear part of the variance time plot using a least squares fit. We consider the aggregation levels $a \geq 192$ in this estimation since our variance time plots are typically linear for these aggregation levels. The Hurst parameter is then estimated as $H = \text{slope}/2 + 1$.

R/S Statistic

We use the R/S statistic [17, 14, 15] to investigate the long range dependence characteristics of the generated traces. The R/S statistic provides an heuristic graphical approach for estimating the Hurst parameter H. Roughly speaking, for long range dependent stochastic processes the R/S statistic is characterized by $E[R(n)/S(n)] \sim cn^H$ as $n \to \infty$ (where c is some positive finite constant). The Hurst parameter H is estimated as the slope of a log–log plot of the R/S statistic.

More formally, the *rescaled adjusted range statistic* (for short *R/S statistic*) is plotted according to the algorithm given in Table 4. The R/S statistic $R(t_i, d)/S(t_i, d)$ is computed for logarithmically spaced values of the lag k, starting with $d = 12$ (to avoid the effect of intra–GoP correlations). For each lag value d as many as K samples of R/S are computed by considering different starting points t_i; we set $K = 10$ in our analysis. The starting points must satisfy $(t_i - 1) + d \leq N$, hence the actual number of samples I is less than K for large lags d. Plotting $\log[R(t_i, d)/S(t_i, d)]$ as a function of $\log d$ gives the *rescaled adjusted range plot* (also referred to as *pox diagram of R/S*). A typical pox diagram starts with a transient zone representing the short range dependence characteristics of the trace. The plot then settles down and fluctuates around a straight "street" of slope H. If the plot exhibits this asymptotic behavior, the *asymptotic Hurst exponent* H is estimated from the street's slope using a least squares fit.

To verify the robustness of the estimate we repeat this procedure for each trace for different aggregation levels $a \geq 1$.

Periodogram

We estimate the Hurst parameter H using the heuristic least squares regression in the spectral domain, see [14, Sec. 4.6] for details. This approach relies on the periodogram $I(\lambda)$ as approximation of the spectral density, which near the origin satisfies

$$\log I(\lambda) \approx \log c_f + (1 - 2H) \log \lambda_k + \log \xi_k. \tag{9}$$

To estimate the Hurst parameter H we plot the periodogram in a log–log plot, as detailed in Table 5. (Note that the expression inside the $| \cdot |$ corresponds to the Fourier transform coefficient at frequency λ_k, which can be efficiently evaluated using Fast Fourier Transform techniques.) For the Hurst parameter estimation we define

$$x_k = \log_{10} \lambda_k \qquad y_k = \log_{10} I(\lambda_k) \tag{10}$$

$$\beta_0 = \log_{10} c_f - 0.577215 \qquad \beta_1 = 1 - 2H \tag{11}$$

$$e_k = \log_{10} \xi_k + 0.577215 \tag{12}$$

With these definitions we can rewrite (9) as

$$y_k = \beta_0 + \beta_1 x_k + e_k. \tag{13}$$

We estimate β_0 and β_1 from the samples $(x_k,\ y_k),\ k = 1, 2, \ldots, \lfloor 0.7 \cdot (N/a - 2)/2 \rfloor := K$ using least squares regression, i.e.,

$$\beta_1 = \frac{K \sum_{k=1}^{K} x_k y_k - \left(\sum_{k=1}^{K} x_k\right)\left(\sum_{k=1}^{K} y_k\right)}{K\left(\sum_{k=1}^{K} x_k^2\right) - \left(\sum_{k=1}^{K} x_k\right)^2} \tag{14}$$

and

$$\beta_0 = \frac{\sum_{k=1}^{K} y_k - \beta_1 \sum_{k=1}^{K} x_k}{K} \tag{15}$$

The Hurst parameter is then estimated as $H = (1 - \beta_1)/2$. We plot the periodogram (along with the fitted line $y = \beta_0 + \beta_1 x$) and estimate the Hurst parameter in this fashion for the aggregation levels $a = 12$, 24, 48, 96, 192, 300, 396, 504, 600, 696, and 792.

Logscale Diagram

We jointly estimate the scaling parameters α and c_f using the wavelet based approach of Veitch and Abry [18], where α and c_f characterize the spectral density

$$f_X(\lambda) \sim c_f \mid \lambda \mid^{-\alpha}, \quad \mid \lambda \mid \rightarrow 0. \tag{16}$$

The estimation is based on the logscale diagram, which is a plot of $\log_2(\mu_j)$ as a function of $\log_2 j$, where

$$\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \mid d_X(j,\ k) \mid^2 \tag{17}$$

is the sample variance of the wavelet coefficient $d_X(j,\ k)$, $k = 1, \ldots, n_j$, at octave j. The number of available wavelet coefficients at octave j is essentially $n_j = N/2^j$.

We plot the logscale diagram for octaves 1 through 14 using the code provided by Veitch and Abry [18]. We use the daubechies 3 wavelet to eliminate linear and quadratic trends [19]. We use the automated `choosenewj1` approach [18] to determine the range of scales (octaves) for the estimation of the scaling parameters.

We report the estimated scaling parameter α, its equivalent representation $H = (1 + \alpha)/2$, as well as the normalized scaling parameter $\underline{c}_f = c_f/S_x^2$.

Multiscale Diagram

We investigate the multifractal scaling properties [20, 21, 22, 23, 19, 18, 24, 25, 26, 27] using the wavelet based framework [22]. In this framework the qth order scaling exponent α_q is estimated based on the qth order logscale diagram, i.e., a plot of

$$\log_2(\mu_j^{(q)}) = \log_2 \frac{1}{n_j} \sum_{k=1}^{n_j} \mid d_X(j,\ k) \mid^q \tag{18}$$

as a function of $\log_2 j$. The multiscale diagram is then obtained by plotting $\zeta(q) = \alpha_q - q/2$ as a function of q. A variation of the multiscale diagram, the so–called linear multiscale diagram is obtained by plotting $h_q = \alpha_q/q - 1/2$ as a function of q.

We employ the multiscaling Matlab code provided by Abry and Veitch [18]. We employ the daubechies 3 wavelet. We use the L1 norm, sigtype 1, the q vector [05, 1, 1.5, 2, 2.5, 3, 3.5, 4]. We use the automated `newchoosej1` approach form Abry and Veitch's logscale diagram Matlab code [18] to determine the range of scales (octaves) for the estimation of the scaling parameters.

B Appendix: Video Quality Metrics

Consider a video sequence with N frames (pictures), each of dimension $D_x \times D_y$ pixels. Let $I(n,\ x,\ y),\ n = 0, \ldots, N-1;\ x = 1, \ldots, D_x;\ y = 1, \ldots, D_y$, denote the luminance (gray–level, or Y component) value of the pixel at location $(x,\ y)$ in video frame n. The Mean Squared Error (MSE) is defined as the mean of the squared differences between the luminance values of the video frames in two video sequences I and $\tilde{I}$. Specifically, the MSE for an individual video frame n is defined as

$$M_n = \frac{1}{D_x \cdot D_y} \sum_{x=1}^{D_x} \sum_{y=1}^{D_y} \left[I(n,\ x,\ y) - \tilde{I}(n,\ x,\ y) \right]^2 . \tag{19}$$

The mean MSE for a sequence of N video frame is

$$\bar{M} \ = \ \frac{1}{N} \sum_{n=0}^{N-1} M_n. \tag{20}$$

The Peak Signal to Noise Ratio (PSNR) in decibels (dB) is generally defined as PSNR $= 10 \cdot \log_{10}(p^2/\text{MSE})$, where p denotes the maximum luminance value of a pixel (255 in 8–bit pictures). We define the *quality* (in dB) of a *video frame* n as

$$Q_n = 10 \cdot \log_{10} \frac{p^2}{M_n}. \tag{21}$$

We define the *average quality* (in dB) of a *video sequence* consisting of N frames as

$$\bar{Q} = 10 \cdot \log_{10} \frac{p^2}{\bar{M}}. \tag{22}$$

Note that in this definition of the average quality, the averaging is conducted with the MSE values and the video quality is given in terms of the PSNR (in dB).

We also define an *alternative average quality* (in dB) of a video sequence as

$$\bar{Q}' = \frac{1}{N} \sum_{n=0}^{N-1} Q_n, \tag{23}$$

where the averaging is conducted over the PSNR values directly.

We now define natural extensions of the above quality metrics. We define the MSE sample variance S_M^2 of a sequence of N video frames as

$$S_M^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} \left(M_n - \bar{M} \right)^2 , \tag{24}$$

and the MSE standard deviation S_M as

$$S_M = \sqrt{S_M^2}. \tag{25}$$

We define the *coefficient of quality variation $CoQV$* of a video sequence as

$$CoQV = \frac{S_M}{\bar{M}}. \tag{26}$$

We define an *alternative quality standard deviation* as

$$S'_Q = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} \left(Q_n - \bar{Q}'\right)^2}, \tag{27}$$

and the corresponding alternative coefficient of quality variation as

$$CoQV' = \frac{S'_Q}{\bar{Q}'}. \tag{28}$$

We define the *quality range* (in dB) of a video sequence as

$$Q_{\min}^{\max} = \max_{0 \le n \le N-1} Q_n - \min_{0 \le n \le N-1} Q_n. \tag{29}$$

We estimate the MSE autocorrelation coefficient $\rho_M(k)$ for lag $k,\ k = 0, \ldots, N-1$, as

$$\rho_M(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} \frac{(M_n - \bar{M})(M_{n+k} - \bar{M})}{S_M^2}. \tag{30}$$

While the above definitions focus on the qualities at the level of individual video frames, we also define, as extensions, qualities for aggregates (groups) of a frames (with the GoP being a special case of frame aggregation with $a = G$, where typically $G = 12$).

Let $M_m^{(a)},\ m = 0, \ldots, N/a - 1$, denote the MSE of the mth group of frames, defined as

$$M_m^{(a)} = \frac{1}{a} \sum_{n=ma}^{(m+1)a-1} M_n. \tag{31}$$

Let $Q_m^{(a)},\ m = 0, \ldots, N/a - 1$, denote the corresponding PSNR quality (in dB), defined as

$$Q_m^{(a)} = 10 \cdot \log_{10} \frac{p^2}{M_m^{(a)}}. \tag{32}$$

We define the MSE sample variance $S_M^{2(a)}$ of a sequence of groups of a frames each as

$$S_M^{2(a)} = \frac{1}{N/a - 1} \sum_{n=0}^{N/a-1} \left(M_n^{(a)} - \bar{M}\right)^2, \tag{33}$$

and the corresponding MSE standard deviation $S_M^{(a)}$ as

$$S_M^{(a)} = \sqrt{S_M^{2(a)}}. \tag{34}$$

We define the coefficient of quality variation $CoQV^{(a)}$ of a sequence of groups of a frames each as

$$CoQV^{(a)} = \frac{S_M^{(a)}}{\bar{M}}. \tag{35}$$

We define the *alternative quality standard deviation* for groups of a frames each as

$$S_Q'^{(a)} = \sqrt{\frac{1}{N/a-1} \sum_{n=0}^{N/a-1} \left(Q_n'^{(a)} - \bar{Q}' \right)^2}, \tag{36}$$

where $Q_n'^{(a)} = \frac{1}{a} \sum_{n=ma}^{(m+1)a-1} Q_n$. We define the corresponding alternative coefficient of quality variation as

$$CoQV'^{(a)} = \frac{S_Q'^{(a)}}{\bar{Q}'}. \tag{37}$$

We define the quality range (in dB) of a sequence of groups of a frames each as

$$Q_{\min}^{\max(a)} = \max_{0 \leq n \leq N/a-1} Q_n^{(a)} - \min_{0 \leq n \leq N/a-1} Q_n^{(a)}. \tag{38}$$

We estimate the MSE autocorrelation coefficient for groups of a frames $\rho_M^{(a)}$ for lag k, $k = 0,\ a,\ 2a, \ldots, N/a-1$ frames as

$$\rho_M^{(a)}(k) = \frac{1}{N/a-k} \sum_{n=0}^{N/a-k-1} \frac{(M_n^{(a)} - \bar{M})(M_{n+k}^{(a)} - \bar{M})}{S_M^{(a)}}. \tag{39}$$

C Appendix: Correlation between Frame Sizes and Qualities

We define the covariance between the frame size and the MSE frame quality as

$$S_{XM} = \frac{1}{N-1} \sum_{n=0}^{N-1} (X_n - \bar{X})(M_n - \bar{M}), \tag{40}$$

and the *size–MSE quality correlation coefficient* as

$$\rho_{XM} = \frac{S_{XM}}{S_X \cdot S_M}. \tag{41}$$

We define the covariance between the frame size and (PSNR) frame quality as

$$S_{XQ} = \frac{1}{N-1} \sum_{n=0}^{N-1} (X_n - \bar{X})(Q_n - \bar{Q}'), \tag{42}$$

and the *size–quality correlation coefficient* as

$$\rho_{XQ} = \frac{S_{XQ}}{S_X \cdot S_Q'}. \tag{43}$$

Similar to the above frame–level definitions, we define the covariance between the aggregated frame sizes $X_n^{(a)}$, $n = 0, \ldots, N/a - 1$, and the aggregated MSE qualities $M_n^{(a)}$, $n = 0, \ldots, N/a - 1$, as

$$S_{XM}^{(a)} = \frac{1}{N/a - 1} \sum_{n=0}^{N/a-1} (X_n^{(a)} - \bar{X})(M_n^{(a)} - \bar{M}), \tag{44}$$

and the corresponding correlation coefficient as

$$\rho_{XM}^{(a)} = \frac{S_{XM}^{(a)}}{S_X^{(a)} \cdot S_M^{(a)}}. \tag{45}$$

We define the covariance between aggregated frame size $X_n^{(a)}$, $n = 0, \ldots, N/a - 1$, and the aggregated (PSNR) qualities $Q_n'^{(a)}$, $n = 0, \ldots, N/a - 1$, as

$$S_{XQ}^{(a)} = \frac{1}{N/a - 1} \sum_{n=0}^{N/a-1} (X_n^{(a)} - \bar{X})(Q_n'^{(a)} - \bar{Q}'), \tag{46}$$

and the corresponding correlation coefficient as

$$\rho_{XQ}^{(a)} = \frac{S_{XQ}^{(a)}}{S_X^{(a)} \cdot S_Q'^{(a)}}. \tag{47}$$

References

[1] M. W. Garret, *Contributions toward Real-Time Services on Packet Networks*, Ph.D. thesis, Columbia University, May 1993.

[2] O. Rose, “Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems,” Tech. Rep. 101, University of Wuerzburg, Institute of Computer Science, Feb. 1995.

[3] M. Krunz, R. Sass, and H. Hughes, “Statistical characteristics and multiplexing of MPEG streams,” in *Proceedings of IEEE Infocom ’95*, April 1995, pp. 455–462.

[4] W.-C. Feng, *Buffering Techniques for Delivery of Compressed Video in Video–on–Demand Systems*, Kluwer Academic Publisher, 1997.

[5] F. Fitzek and M. Reisslein, “MPEG–4 and H.263 video traces for network performance evaluation,” *IEEE Network*, vol. 15, no. 6, pp. 40–54, November/December 2001, Video traces available at `http://www.eas.asu.edu/trace`.

[6] A. Ortega and K. Ramchandran, “Rate–distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–46, Nov. 1998.

[7] G. J. Sullivan and T. Wiegand, “Rate–distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[8] J. Walter, “bttvgrab,” `http://www.garni.ch/bttvgrab/`.

[9] ISO/IEC 14496, “Video Reference Software, Microsoft–FDAM1–2.3–001213,” .

[10] Test Model Editing Committee, “MPEG–2 Video Test Model 5, ISO/IEC JTC1/SC29WG11 MPEG93/457,” Apr. 1993.

[11] Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Resource allocation for multimedia streaming over the internet,” *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 339–355, Sept. 2001.

[12] A. M. Law and W. D. Kelton, *Simulation, Modeling and Analysis*, McGraw Hill, third edition, 2000.

[13] C. Chatfield, *The Analysis of Time Series: An Intoduction*, Chapman and Hall, fourth edition, 1989.

[14] J. Beran, *Statistics for long–memory processes*, Chapman and Hall, 1994.

[15] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, “Long–range dependence in variable–bit–rate video traffic,” *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1566–1579, February/March/April 1995.

[16] M. Krunz, “On the limitations of the variane–time test for inference of long–range dependence,” in *Proceedings of IEEE Infocom 2001*, Anchorage, Alaska, Apr. 2001, pp. 1254–11260.

[17] B. B. Mandelbrot and M. S. Taqqu, “Robust R/S analysis of long–run serial correlations,” in *Proceedings of 42nd Session ISI, Vol. XLVIII, Book 2*, 1979, pp. 69–99.

[18] D. Veitch and P. Abry, “A wavelet based joint estimator of the parameters of long–range dependence,” *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 878–897, Apr. 1999, Matlab code available at `http://www.emulab.ee.mu.oz.au/~darryl`.

[19] P. Abry and D. Veitch, “Wavelet analysis of long–range–dependent traffic,” *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 2–15, Jan. 1998.

[20] P. Abry, D. Veitch, and P. Flandrin, “Long–range–dependence: Revisiting aggregation with wavelets,” *Journal of Time Series Analysis*, vol. 19, no. 3, pp. 253–266, May 1998.

[21] M. Roughan, D. Veitch, and P. Abry, “Real–time estimation of the parameters of long–range dependence,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 467–478, Aug. 2000.

[22] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, “Wavelets for the analysis, estimation and synthesis of scaling data,” in *Self Similar Network Traffic Analysis and Performance Evaluation (Wiley)*, K. Park and W. Willinger, Eds., 2000.

[23] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, “Self–similarity and long–range dependence through the wavelet lens,” in *Long Range Dependence: Theory and Applications*, Doukhan, Oppenheim, and Taqqu, Eds., 2002.

[24] A. Feldmann, A. C. Gilbert, W. Willinger, and T.G. Kurtz, “The changing nature of network traffic: Scaling phenomena,” *Computer Communciation Review*, vol. 28, no. 2, Apr. 1998.

[25] J. Gao and I. Rubin, “Multiplicative multifractal modeling of long–range–dependent network traffic,” *International Journal of Communication Systems*, vol. 14, pp. 783–801, 2001.

[26] A. C. Gilbert, W. Willinger, , and A. Feldmann, “Scaling analysis of conservative cascades, with applications to network traffic,” *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 971–991, Apr. 1999.

[27] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk, “A multifractal wavelet model with applications to network traffic,” *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 992–1018, Apr. 1999.

Table 1: Overview of studied video sequences.

Movies (rental tapes)	Format	Length (min)	Frames Dropped
Citizen Kane	QCIF	60	0
Die Hard I	QCIF	60	0
Jurassic Park I	QCIF	60	0
Jurassic Park II	QCIF	60	0
Silence of the Lambs	QCIF	60	0
Silence of the Lambs	CIF	30	63
Star Wars IV	QCIF	60	0
Star Wars V	QCIF	60	0
The Firm	QCIF	60	0
The Terminator I	QCIF	60	0
The Terminator I	CIF	30	16
Total Recall	QCIF	60	0

Cartoons (rental tapes)	Format	Length (min)	Frames Dropped
Alladin	QCIF	60	0
Cinderella	QCIF	60	0
Toy Story I	QCIF	60	0
Toy Story I	CIF	30	52
Toy Story II	QCIF	60	0

Sports Events (recorded from Broadcast TV)	Format	Length (min)	Frames Dropped
Baseball Game 7 of the 2001 World Series	QCIF	60	0
Basketball /w comm. NBA Basketball Game	QCIF	60	0
Basketball w/o comm. NBA Basketball Game	QCIF	60	0
Football NFL Football Game	QCIF	60	0
Football NFL Football Game	CIF	30	89
Golf Seniors PGA Tournament	QCIF	60	0
Snowboarding Snowboarding Competition	QCIF	60	0

Other TV sequences (recorded from Broadcast TV)	Format	Length (min)	Frames Dropped
Music Clips VH1 Music Videos Clips	QCIF	60	0
Oprah American Talk Show (without Commercials)	QCIF	60	0
Oprah American Talk Show (without Commercials)	CIF	30	64
Oprah American Talk Show (with Commercials)	QCIF	60	0
PBS News News Hour with Jim Lehrer	QCIF	60	0
PBS News News Hour with Jim Lehrer	CIF	30	64
Tonight Show Late Night Show (without Commercials)	QCIF	60	0
Tonight Show Late Night Show (with Commercials)	QCIF	60	0

Lectures and Set–top	Format	Length (min)	Frames Dropped
Lecture MR EEE 554 Lecture by Prof. Martin Reisslein on 09/05/2001	QCIF	60	0
Lecture SG Lecture by Prof. Sandeep Gupta on 09/05/2001	QCIF	60	0
Lecture SG Lecture by Prof. Sandeep Gupta on 09/05/2001	CIF	30	61
Security–Cam Parking Lot Security Camera	QCIF	60	0

Table 2: Quantization parameter settings for defined quality levels.

		Quantization Parameter Setting		
Quality Level		I Frame	P Frame	B Frame
Low	1	30	30	30
Medium–Low	2	24	24	24
Medium	3	10	14	16
High–Medium	4	10	10	10
High	5	4	4	4

Table 3: Algorithm for variance time plot.

```
$S_X^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} (X_n - \bar{X})^2$
**For** $a = 12,\ 24,\ 48,\ 96, \ldots,\ 12288$ **do**
    $M = \lfloor N/a \rfloor$
    $X_n^{(a)} = \frac{1}{a} \sum_{j=na}^{(n+1)a-1} X_j, \quad n = 0, \ldots, M-1$
    $S_X^{2(a)} = \frac{1}{M-1} \sum_{n=0}^{M-1} (X_n^{(a)} - \bar{X})^2$
    plot point $\left(\log_{10} a,\ \log_{10}(S_X^{2(a)}/S_X^2)\right)$
```

Table 4: Algorithm for pox diagram of R/S.

```
**For** $d = 12,\ 24,\ 48,\ 96, \ldots$ **do**
    $I = K + 1 - \lceil \frac{dK}{N} \rceil$
    **For** $i = 1, \ldots, I$ **do**
        $t_i = (i-1)\frac{N}{K} + 1$
        $\bar{X}(t_i, d) = \frac{1}{d} \sum_{j=0}^{d-1} X_{t_i+j}^{(a)}$
        $S^2(t_i, d) = \frac{1}{d} \sum_{j=0}^{d-1} [X_{t_i+j}^{(a)} - \bar{X}(t_i, d)]^2$
        $R(t_i, d) = \max\{0,\ \max_{1 \le k \le d} W(t_i, k)\} - \min\{0,\ \min_{1 \le k \le d} W(t_i, k)\}$
            $W(t_i, k) = \left(\sum_{j=0}^{k-1} X_{t_i+j}^{(a)}\right) - k\bar{X}(t_i, d)$
        plot point $\left(\log d,\ \log \frac{R(t_i,d)}{S(t_i,d)}\right)$
```

Table 5: Algorithm for periodogram.

```
$M = \lfloor N/a \rfloor$
$X_n^{(a)} = \frac{1}{a} \sum_{j=na}^{(n+1)a-1} X_j,\ n = 0, \ldots, M-1$
$Z_n^{(a)} = \log_{10} X_n^{(a)},\ n = 0, \ldots, M-1$
**For** $k = 1,\ 2,\ 3,\ 4, \ldots, \lfloor \frac{M-1}{2} \rfloor$ **do**
    $\lambda_k = \frac{2\pi k}{M}$
    $I(\lambda_k) = \frac{1}{2\pi M} \mid \sum_{n=0}^{M-1} Z_n^{(a)} e^{-jn\lambda_k} \mid^2$
    plot point $(\log_{10} \lambda_k,\ \log_{10} I(\lambda_k))$
```