MPEG–4 and H.263 Video Traces for Network Performance Evaluation

Frank H.P. Fitzek
Technical University Berlin
fitzek@ee.tu-berlin.de

Martin Reisslein *†
Arizona State University
reisslein@asu.edu

Received: October 2000
Revised: May 2001

Abstract

MPEG–4 and H.263 encoded video is expected to account for a large portion of the traffic in future wireline and wireless networks. However, due to a lack of sufficiently long frame size traces of MPEG–4 and H.263 encoded videos, most network performance evaluations currently use MPEG–1 encodings. In this article we present and study a publicly available library of frame size traces of long MPEG–4 and H.263 encoded videos, which we have generated at the Technical University Berlin. The frame size traces have been generated from MPEG–4 and H.263 encodings of over 10 video sequences of 60 minutes length each. We conduct a thorough statistical analysis of the traces.

Keywords: Frame Size Traces, H.263, Long Range Dependence, MPEG–4, Statistical Analysis, Video Encoding.

1 Introduction

MPEG–4 and H.263 encoded video is expected to account for large portions of the traffic in future wireline and wireless networks. To date the statistical analysis of MPEG–4 and H.263 encoded video has received only little attention in the literature. Similarly, there are only few studies that evaluate networking protocols and resource management schemes with MPEG–4 and H.263 encoded video. This is partly due to a lack of sufficiently long frame size traces of MPEG–4 and H.263 encoded videos. In fact, most researches currently use the MPEG–1 encodings of Garret [1], Rose [2], Krunz *et al.* [3], or Feng [4, 5]. These frame size traces

*Parts of this work were conducted while Martin Reisslein was with GMD Fokus, Berlin, Germany.

†Corresponding author: Martin Reisslein, Arizona State University, Dept. of Electrical Eng., Telecommunications Research Center, P.O. Box 877206, Tempe, AZ 85287–7206, phone: (480)965–8593, fax: (480)965–8325, www.eas.asu.edu/~mre.

give the sizes (in bit or byte) for each encoded video frame. Networking researchers use video frame size traces for video traffic studies [6], and video traffic modeling [7, 8], as well as for the development and evaluation of protocols and mechanisms for packet–switched networks [9, 10, 11, 12, 13, 14], wireless networks [15], and optical networks [16]. The cited works are just a small sample of the hundreds of works that have made use of video traces over the last couple of years.

In this article we report on a new publicly available library of frame size traces of long MPEG–4 and H.263 encoded videos, which we have generated in the Telecommunication Networks (TKN) Group at the Technical University Berlin. (The trace library is available at `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` and `http://www.eas.asu.edu/trace`.) The frame size traces have been generated from MPEG–4 and H.263 encodings of over 10 video sequences of 60 minutes length each. We present a thorough statistical analysis of the frame size traces. We study moments and autocorrelations as well as the long range dependence characteristics. We estimate the Hurst parameter of the traces with the R/S statistic.

This article is structured as follows. In Section 2 we give an overview of digital video as well as MPEG–4 and H.263 compression. In Section 3 we describe the generation of the frame size traces. We give an overview of the encoded video sequences and discuss the capturing of the uncompressed video information. We describe our MPEG–4 and H.263 encoding procedures in detail. In Section 4 we conduct a thorough statistical analysis of the generated MPEG–4 and H.263 frame size traces. We summarize our contributions in Section 5. In the Appendix we review the statistical methods used in the analysis of the traces.

2 Overview of Digital Video

First, we give a brief overview of digital video (we refer the interested reader to [17, 18] for a more detailed discussion). Let us start with an analog video signal generated by an analog video camera. The analog video signal consists of a sequence of video frames. The video frames are generated at a fixed frame rate (30 frames per second in the National Television Standards Committee (NTSC) format). For each video frame, the video camera scans the frame line by line (with 455 lines in NTSC). To obtain a digital video signal the analog video signal is passed to a digitizer. The digitizer samples and quantizes the analog video signal. Each sample corresponds to a picture element (pel). The most common digital frame formats are Common

Intermediate Format (CIF) with 352x288 pels (i.e., 352 pels in the horizontal direction and 288 pels in the vertical direction), Source Intermediate Format (SIF) with 352x240 pels, and Quarter CIF (QCIF) with 176x144 pels. In all three frame formats, each video frame is divided into three components. These are the luminance component (Y), and the two chrominance components: hue (U) and intensity (saturation) (V). Since the human eye is less sensitive to the color information, than to the luminance information, the chrominance components are sampled at a lower resolution. Typically, each chrominance component is sampled at half the resolution of the luminance component in both the horizontal and the vertical direction. (This is referred to as 4:1:1 chroma sub sampling.) In the QCIF frame format, for instance, there are 176x144 luminance samples, 88x72 hue samples, and 88x72 intensity samples in each video frame, when 4:1:1 chroma subsampling is used. Finally, each sample is quantized; typically, 8 bits are used per sample.

As an aside we note that the YUV video format was introduced to make color TV signals backward compatible with black–and–white TV sets, which can only display the luminance (brightness) components. Computer monitors, on the other hand, use typically the RGB video format, which contains red, green, and blue components for each pel.

Before we discuss the specific features of MPEG–4 and H.263 we briefly outline some of their common aspects. Both encoding standards employ the Discrete Cosine Transform (DCT) [17] to reduce the spatial redundancy in the individual video frames. Each video frame is divided into Macro Blocks (MBs). A macro block consists of 16x16 samples of the luminance component and the corresponding 8x8 samples of the two chrominance components. The 16x16 samples of the luminance component are divided into four blocks of 8x8 samples each. The DCT is applied to each of the six blocks (i.e., four luminance blocks and two chrominance blocks) in the macro block. For each block the resulting DCT coefficients are quantized using an 8x8 quantization matrix, which contains the quantization step size for each DCT coefficient. The quantization matrix is obtained by multiplying a base matrix by a quantization parameter. This quantization parameter is typically used to tune the video encoding. A larger quantization parameter results in coarser quantization, which in turn results in a lower quality as well as a smaller size (in bit) of the encoded video frame. The quantized DCT coefficients are finally variable–length–coded, for a more compact representation.

Both, MPEG–4 and H.263 employ predictive encoding to reduce the temporal redundancy, that is, the temporal correlation between successive video frames. A given macroblock is either intracoded (i.e., without reference to another frame) or intercoded (i.e., with reference to a preceding or succeeding frame). To intercode a given macroblock, a motion search is conducted to find the best matching 16x16 sample area in the preceding (or succeeding) frame. The difference between the macroblock and the best matching area is DCT coded, quantized, and variable–length–coded, and then transmitted along with a motion vector to the matching area.

2.1 Overview of MPEG–4 Video Compression

In this section we provide a brief overview of MPEG–4 video coding; we refer the reader to [19, 20, 21, 17, 18] for details. MPEG–4 provides very efficient video coding covering the range from the very low bit rates of wireless communication to bit rates and quality levels beyond high definition television (HDTV). In contrast to the "frame–based" video coding of MPEG–1 and H.263, MPEG–4 is object based. Each scene is composed of Video Objects (VOs) that are coded individually. (If scene segmentation is not available or not useful, e.g., in very simple wireless video communication, the standard defines the entire scene as one VO.) Each VO may have several scalability layers (i.e., one base layer and one or several enhancement layers), which are referred to as Video Object Layers (VOLs) in MPEG–4 terminology. Each VOL in turn consists of an ordered sequence of snapshots in time, referred to as Video Object Planes (VOPs). For each VOP the encoder processes the shape, motion, and texture characteristics.

The shape information is encoded by bounding the VO with a rectangular box and then dividing the bounding box into Macro Blocks (MBs). Each MB is classified as lying (*i*) inside the object, (*ii*) on the object's border, or (*iii*) outside the object (but inside the bounding box). The "border" MBs are then shape coded. The texture coding is done on a per–block basis similar to the "frame–based" standards, such as MPEG–1 and H.263. In an Intracoded (I) VOP the absolute texture values in each MB are Discrete Cosine Transform (DCT) coded. The DCT coefficients are then quantized and variable–length–coded. In forward Predicted (P) VOPs each MB is predicted from the closest match in the preceding I (or P) VOP using motion vectors. In Bi–directionally predicted (B) VOPs each MB is predicted from the preceding I (or P) VOP and the succeeding P (or I) VOP. The prediction errors are DCT coded, quantized, and variable–length–coded. The I, P, and B VOPs are arranged in a periodic pattern referred to as Group of Pictures (GoP). A typical GoP structure is IBBPBBPBBPBB. For the transmission the shape, motion, and texture information is multiplexed at the MB level, i.e., for a given MB the shape information is transmitted first, then the motion information, and then the texture information, then the shape information of the next MB, and so on. To combat the frequent transmission errors typical for wireless communication, MPEG–4 provides a number of error resilience and error concealment features; we refer the reader to [19, 20, 21, 17, 18] for details.

2.2 Overview of H.263 Video Compression

The basic structure of the H.263 video source coding algorithm [22, 23, 17, 18] has been adopted from ITU–T Recommendation H.261 [24]. It uses (1) inter picture prediction to reduce the temporal redundancy, and (2) Discrete Cosine Transform (DCT) coding of the residual prediction error to reduce the spatial redundancy. After the DCT coding, the prediction error is quantized and the resulting symbols are variable–length–coded and transmitted. For the interpicture prediction each video frame is divided into macro blocks and one motion vector is transmitted per macro block. In contrast to H.261, half pixel prediction is used for the motion vectors in H.263. The bit rate of the compressed video stream is controlled by adjusting several encoder parameters, such as quantizer scales and the frame rate. H.263 provides four advanced coding options. Unrestricted motion vectors, advanced prediction, and PB–frames are options that improve the inter–picture prediction. The fourth option is to use the more efficient arithmetic coding instead of variable–length–coding. These four options improve the video quality at the expense of increased video codec complexity. We refer the reader to [22, 23, 18] for details. Roughly speaking, the unrestricted motion vector option allows motion vectors to point outside the video frame. The edge pels are used instead of prediction pels that lie outside the frame. This allows for more efficient compression, especially when there is motion near the frame border and the frame format is small. With advanced prediction the motion predicted blocks overlap and a pel is interpreted as the weighted average of the overlapping blocks. This reduces artifacts in the decoded video frames and increases the perceived video quality. The PB–frames option increases the frame rate without significantly increasing the bit rate. A PB–frame consists of two consecutive frames that are encoded as one entity. Specifically, a PB–frame consists of a P–frame, which is predicted from the preceding P–frame, and a B–frame, which is bi–directionally predicted from the preceding P–frame and the P–frame being part of the PB entity. When the reconstruction of the PB–frame is complete, the B–frame is displayed first and then the P–frame.

3 Video Trace Generation

In this section we describe the generation of the video frame size traces. This process is illustrated in Figure 1, which we refer to throughout this section.

3.1 Overview and Capturing of Video Sequences

We played the videos [1] listed in Table 1 from VHS tapes using a Video Cassette Recorder (VCR). For ease of comparison with the existing MPEG–1 traces we included *Star Wars IV*,

Table 1: Overview of encoded video sequences.

Movies (rental tapes, German/English movie versions)
Jurassic Park I (G) *Silence of the Lambs* (E) *Star Wars IV* (E) *Mr. Bean* (G) *Star Trek: First Contact* (G) *Form Dusk Till Dawn* (G) *The Firm* (G)
Sports Events (recorded from German cable TV)
Formula 1: Formula 1 car race *Soccer*: Soccer game (European championship 1996)
Other TV sequences (recorded from German cable TV)
ARD News: German news (Tagesschau) *ARD Talk*: German Sunday morning talk show (Presseclub) *N3 Talk*: German late night show (Herman und Tietjen)
Set–top
Office–Cam: Office camera observing person in front of terminal

which has been MPEG–1 encoded by Garrett [1], and several of the movies that have been MPEG–1 encoded by Rose [2] in our video selection. (For a given movie, the German and English releases are sometimes edited according to slightly different criteria and may therefore differ slightly in the scene content. For this reason we indicate whether we encoded the German or English version.) For each video we captured the (uncompressed) YUV information using a PC video capture card and the `bttvgrab` (Version 0.15.10) software [25]. We stored the YUV information on disk. The YUV information was grabbed at a frame rate of 25 frames/sec

[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 researches.

in the QCIF format with 4:1:1 chrominance subsampling and quantization into 8 bits. We chose the QCIF format because we are particularly interested in generating traces for the evaluation of wireless networking systems. We expect that hand–held wireless devices of next–generation wireless systems will typically have a screen size that corresponds to the QCIF video format. We note that `bttvgrab` is a high–quality grabbing software. It is designed to not leave out a single video frame and to overcome temporal delays by buffering several frames. To avoid potential hardware problems due to buffer build–up when grabbing long video sequences, we grabbed the 60 minutes video sequences in segments of 22,501 frames ($\approx$ 15 minutes of video runtime). Any 22,501 frame segment of any video gave exactly 855,398,016 Bytes of uncompressed YUV information (= 38,016 Bytes/frame). (Note that in the QCIF format there are 176x144 + 2 $\cdot$ 88x72 = 38,016 pels per frame. With 8 bit quantization and 25 frames per second the bit rate of uncompressed QCIF video is 38,016 pels/frame $\cdot$ 8 bit/pel $\cdot$ 25 frames/sec = 7,603,200 bit/sec.) The stored YUV frame sequences were used as input for both the MPEG–4 encoder and the H.263 encoder. We emphasize that we did not encode in real–time; thus there was no encoder bottleneck.

3.2 Encoding Approach for MPEG–4

For each video we encoded the YUV information into an MPEG–4 bit stream with the MOMUSYS MPEG–4 video software [26], which has been adopted by MPEG in the MPEG–4 standard, Part 5 — Reference Software. We set the number of video objects to one, i.e., the entire scene is one video object. The width of the display is set to 176 pels, the height is set to 144 pels. We used a pel depth of 8 bits per pel. We did not use rate control in the encoding. The single video object was encoded into a single video object layer. We set the video object layer frame rate, i.e., the rate at which video object planes are generated, to 25 frames/sec. The Group of Pictures (GoP) pattern was set to IBBPBBPBBPBB. We encoded each video at three different quality levels: *low*, *medium*, and *high*. For the low quality encoding the quantization parameters were fixed at 10 for I frames (VOPs), 14 for P frames, and 18 for B frames. For the medium quality encoding the quantization parameters for all three frame types were fixed at 10. For the high quality encoding the quantization parameters for all three frame types were fixed at 4. We refer the interested reader to the technical report [27] for a complete listing of the parameters settings used in the encodings.

We note that the MOMUSYS MPEG–4 encoder is limited to encoding segments with a length of at most 1,000 video frames. Therefore, we encoded the YUV frame sequences in segments of 960 frames (= 80 GoPs) each. When encoding a given 80 GoP segment, the last two B frames of the 80th GoP are bi–directionally predicted from the third P frame of the 80th GoP and the I frame of the 81st GoP. Since the 81st GoP is not encoded, the last two B frames of the 80th GoP are not encoded either. As a consequence our frame size files were missing two B frames per 960 encoded video frames (= 38.4 seconds of video runtime). As a remedy we inserted two B frames at the end of each segment of 958 (actually encoded) frames. We set the size of the inserted B frames to the average size of the B frames in the 958 frame segment. We believe that this error, that is due to the limitations of the MOMUSYS MPEG–4 Reference Software, can be neglected.

3.3 Encoding Approach for H.263

We encoded the uncompressed YUV information into an H.263 bit stream with the `tmn` encoder (Version 2.0) [28]. (We did not use the H.263 encoder of `bttvgrab` because it is not fully compliant with the H.263 standard; it inserts additional sequencing and synchronization information into the H.263 bit stream.) We emphasize that we did not encode in real–time; thus there was no encoder bottleneck. We set the `tmn` encoder parameters to encode in the QCIF (176x144 pel) video format at a fixed reference frame rate of 25 frames/sec. We did not enable unrestricted motion vectors, syntax–based arithmetic coding, and advanced prediction, since we observed that these features bring only little improvement in the video quality while slowing down the encoder dramatically. We did enable PB–frames. We encoded each video at four different target bit rates: (1) 16 kbit/sec, (2) 64 kbit/sec, (3) 256 kbit/sec, and (4) Variable Bit Rate (VBR), i.e., without setting a target bit rate. We refer the interested reader to the technical report [27] for a more detailed description of the encoder parameter settings.

3.4 Extracting Frame Sizes

Finally, we obtained the frame sizes (in bytes) of the individual encoded video frames by directly parsing the encoded MPEG–4 and H.263 bit streams.

We note that Ryu [29] encoded and analyzed a one hour CNN news video and a one our C–SPAN video using the MBONE video tool `vic` [30]. The `vic` tool employs an encoding scheme that is roughly equivalent to H.261. Similarly, Dolzer and Payer [31] encoded and analyzed a political talk show using the `vic` tool. In both works the `vic` packet stream was sent over a packet–switched network and the video's frame sizes were extracted from the packet time stamps using `tcpdump`. It may be argued that this indirect measurement of the frame sizes is less accurate due to the influence of the underlying packet–switched network. For this reason we extract the frame sizes directly from the encoded bit streams.

4 Statistical Analysis

4.1 Statistical Analysis of MPEG–4 Traces

In this section we conduct a thorough statistical analysis of the generated MPEG–4 frame size traces. For the analysis we introduce the following notation. Let N denote the number of video frames in a given trace. Let t denote the frame period (display time) of a given frame. Note that for almost all our MPEG–4 traces approximately $N = 90.000$ and $t = 40$ msec, which corresponds to a video runtime of about 60 minutes. Let $X_n,\ n = 1, \ldots, N$, denote the number of bits in frame n, that is, the frame size of frame n. Let G denote the number of frames per Group of Pictures (GoP). Let $Y_m,\ m = 1, \ldots, N/G$, denote the number of bits in GoP m, that is, the size of GoP m. Clearly, $Y_m = \sum_{n=(m-1)G+1}^{mG} X_n$.

Frame Sizes and Bit Rates

Tables 2 and 3 give an overview of the statistical properties of the generated MPEG–4 traces. (To conserve space we give the statistics of all three quality levels only for the *Jurassic Park I*, *Silence of the Lambs*, and *Star Wars IV* videos. We refer the interested reader to the technical report [27] for the omitted results.) The compression ratio is defined as the ratio of the size of the entire uncompressed YUV video sequence (in bit) to the size of the entire MPEG–4 compressed video sequence (in bit). The Mean $\bar{X}$ gives the average frame size. The Coefficient of Variation (defined as the standard deviation S_X of the frame size divided by the average frame size $\bar{X}$) is a typical metric for the variability of the frame sizes; the larger the coefficient of variation the more variable are the frame sizes. (We refer the reader to the Appendix for the formal definitions of the mean and the coefficient of variation.) Comparing encodings at different quality levels we observe that lower quality encoding achieves higher compression ratios as well as smaller mean frame sizes and smaller mean bit rates, as is to be expected. The lower quality encodings, however, have significantly larger coefficients of variation and peak–to-mean ratios of the frame sizes, that is, they are much more variable (bursty). We observe that relatively high compression ratios are achieved for the *Star Wars IV* movie even for high quality encodings. This is probably due to the long scenes with dark backdrops and little contrast in this movie. For the *Formula 1* and *Soccer* videos, on the other hand, only

Table 2: Overview of frame statistics of MPEG–4 traces

		Compr.	Frame Size			Bit Rate	
		ratio	Mean	CoV	Peak/Mean	Mean	Peak
Quality	Trace	YUV:MP4	$\bar{X}$ [kbyte]	$S_X/\bar{X}$	$X_{\max}/\bar{X}$	$\bar{X}/t$ [Mbps]	$X_{\max}/t$ [Mbps]
High	*Jurassic Park I*	9.92	3.8	0.59	4.37	0.77	3.3
	Silence of the Lambs	13.22	2.9	0.80	7,73	0.58	4.4
	Star Wars IV	27.62	1.4	0.66	6.81	0.28	1.9
	Mr. Bean	13.06	2.9	0.62	5.24	0.58	3.1
	First Contact	23.11	1.6	0.73	7.59	0.33	2.5
	From Dusk Till Dawn	11.16	3.4	0.58	4.62	0.68	3.1
	The Firm	24.53	1.5	0.75	6.69	0.31	2.1
	Formula 1	9.10	4.2	0.42	3.45	0.84	2.9
	Soccer	6.87	5.5	0.41	3.24	1.10	3.6
	ARD News	10.52	3.6	0.70	4.72	0.72	3.4
	ARD Talk	13.95	2.7	0.63	5.72	0.54	3.1
	N3 Talk	13.76	2.8	0.60	6.17	0.55	3.4
	Office–Cam	19.16	2.0	1.09	4.99	0.40	2.0
Medium	*Jurassic Park I*	28.4	1.3	0.84	6.36	0.27	1.7
	Silence of the Lambs	43.43	0.88	1.21	13.6	0.18	2.4
	Star Wars IV	97.83	0.39	1.17	12.1	0.08	0.94
Low	*Jurassic Park I*	49.46	0.77	1.39	10.61	0.15	1.6
	Silence of the Lambs	72.01	0.53	1.66	21.39	0.11	2.3
	Star Wars IV	142.52	0.27	1.68	17.57	0.053	0.94

relatively small compression ratios are achieved. These videos feature many small objects that move rapidly. This results in high mean bit rates and relatively small peak–to–mean ratios of the encoded frame sizes. Comparing the frame statistics and the GoP statistics we observe that smoothing the videos over one GoP (= 0.48 sec of video runtime) is quite effective in reducing the variability and the peak rate. Nevertheless, the GoP smoothed video traffic is quite bursty with peak–to–mean ratios of the GoP sizes of three and larger (which is in contrast to the assumption of non–bursty streaming traffic in the current 3rd Generation Wireless System technical specification [32]).

In the following we provide plots to illustrate the statistical properties of the following three MPEG–4 traces: (a) *Star Wars IV* encoded at high quality, (b) *Jurassic Park I* encoded at medium quality, and (c) *Silence of the Lambs* encoded at low quality. Figure 2 gives the frame size traces, i.e., the frame size X_n (in bytes) as a function of the frame number n. We observe from the plots that the *Star Wars IV* encoding at high quality is relatively smooth. The *Silence of the Lambs* encoding at low quality, on the other hand, exhibits extreme changes in the frame sizes. Inspecting this trace closely, we are able to identify periods during which the frame sizes stay roughly at a fixed level; these periods appear to correspond to distinct

Table 3: Overview of GoP statistics of MPEG–4 traces

		GoP Size			Bit Rate	
		Mean	CoV	Peak/Mean	Mean	Peak
Quality	Trace	$\bar{Y}$ [kbyte]	$S_Y/\bar{Y}$	$Y_{\max}/\bar{Y}$	$\bar{Y}/(Gt)$ [Mbps]	$Y_{\max}/(Gt)$ [Mbps]
High	*Jurassic Park I*	46	0.47	3.15	0.77	2.4
	Silence of the Lambs	35	0.71	6.22	0.58	3.6
	Star Wars IV	17	0.38	4.29	0.28	1.2
Medium	*Jurassic Park I*	16	0.57	3.92	0.27	1.0
	Silence of the Lambs	11.0	0.99	10.07	0.18	1.8
	Star Wars IV	4.7	0.52	6.29	0.08	0.49
Low	*Jurassic Park I*	9.20	0.53	4.05	0.15	0.62
	Silence of the Lambs	6.30	0.92	10.48	0.11	1.10
	Star Wars IV	3.20	0.46	5.31	0.053	0.28

scenes in the movie.

Figure 3 gives the histograms of the frame size X_n. The histogram plots reflect again the general tendency that lower quality encodings (i.e., higher compression ratios) result in more variability of the encoded video stream. The irregular shapes of the histograms illustrate the difficulty in modeling the frame size distributions.

Correlations and Long Range Dependence

Figure 4 gives the autocorrelation coefficient $\rho_X(k)$ (see Appendix for the formal definition) of the frame size sequence $X_n,\ n = 1, \ldots, N$, as a function of the lag k (in frames). The frame size correlations exhibit a periodic spike pattern that is superimposed on a decaying slope. The periodic spike pattern reflects the repetitive GoP pattern. The large positive spikes are due to (the typically large) I frames. An I frame is followed by two (typically small) B frames, which appear as small negative spikes. The subsequent P frame (typically of mid–size) shows up as a small positive spike. The decaying slope is characteristic of the long term correlations in the encoded video. To get a clearer picture of these long term correlations we show in Figure 5 the autocorrelation coefficient $\rho_Y(k)$ of the GoP size sequence $Y_m,\ m = 1, \ldots, N/G$, as a function of the lag k (in GoPs). We observe from the figure that the GoP autocorrelation function of the *Jurassic Park I* encoding at medium quality decays roughly exponentially. This indicates that the GoP size process is memoryless. The other two curves clearly decay slower than an exponential function. This slow decay of the GoP autocorrelation is particularly pronounced for the *Silence of the Lambs* encoding at low quality, which has an autocorrelation coefficient of roughly 0.2 for a lag of 230 GoPs (approximately 110 sec).

The time–dependent statistics are important for network and traffic engineering since correlations in the video traffic can have a significant impact on the performance of packet–switched networks. Several studies [33, 34, 35, 36, 37] have found that the losses and/or delays of queuing systems are considerably larger for positively correlated input traffic than uncorrelated input traffic. It has also been demonstrated [38] that carefully designed VBR traffic models are able to capture the relevant range of correlations and predict the system performance accurately. For these reasons it is important to analyze the long range correlations of the video traces. These long range correlations are formally characterized as self–similarity or long–range dependence (LRD) [39, 37]. Intuitively, long–range dependent traffic is bursty (highly variable) over a wide range of time scales. The cumulative effect of the correlations for large lags is significant and gives rise to the large losses and/or delays found for long–range dependent traffic (even though the correlations for large lags may be individually small).

The Hurst parameter is a succinct metric for the long–range dependence (i.e., the degree of self–similarity). Generally speaking, time series without long range dependence have a Hurst parameter of 0.5. Hurst parameters between 0.5 and 1 indicate long range dependence. Additionally, larger Hurst parameters indicate a higher degree of long range dependence.

We estimated the Hurst parameters of the frame size traces from pox plots of the R/S statistic, as outlined in the Appendix. For each frame size trace we generated pox plots of R/S for different aggregation levels a, that is, we averaged the frame size traces over non–overlapping blocks of a frames and then plotted the pox diagram of R/S according to the algorithm given in Table 9. Figure 6 gives some pox plots of R/S for an aggregation level of $a = 1$. The Hurst parameter is estimated from the slope of the "street of points" in the pox plot. Table 4 gives the Hurst parameters of the MPEG–4 frame size traces as a function of the aggregation level a. Generally, Hurst parameters larger than 0.5 for all aggregation levels are a strong indication of long range dependence. We observe from the table that the encodings of *Silence of the Lambs*, *Star Wars IV*, *Mr. Bean*, *First Contact*, *From Dusk Till Dawn*, and *The Firm* have Hurst parameters larger than 0.72 for all aggregation levels. This indicates a high degree of long range dependence. The *Formula 1* and *ARD news* encodings have large Hurst parameters for aggregation levels of 50 frames and less; for aggregation levels of 200 frames and larger, however, the Hurst parameters are around 0.5. These results are thus not a strong indication of long range dependence. It is also interesting to note that the Hurst parameters for the *Jurassic Park I* encodings do not give a strong indication of long range dependence properties. This corroborates the observation that the GoP autocorrelation functions decay almost exponentially; thus indicating the memoryless property.

Table 4: Hurst parameters of MPEG–4 traces estimated from pox diagram of R/S as a function of the aggregation level a.

		Aggregation level a [frames]										
Quality	Trace	1	12	50	100	200	300	400	500	600	700	800
High	*Jurassic Park I*	0.973	0.830	0.795	0.774	0.737	0.753	0.666	0.653	0.705	0.591	0.622
	Silence of the Lambs	1.007	0.894	0.872	0.868	0.894	0.852	0.819	0.741	0.765	0.728	0.771
	Star Wars IV	0.903	0.838	0.808	0.785	0.776	0.765	0.756	0.758	0.752	0.727	0.722
	Mr. Bean	0.933	0.866	0.866	0.861	0.824	0.740	0.792	0.819	0.765	0.793	0.810
	First Contact	0.931	0.831	0.807	0.791	0.763	0.760	0.726	0.747	0.777	0.776	0.784
	Form Dusk Till Dawn	0.909	0.829	0.806	0.781	0.754	0.726	0.733	0.773	0.728	0.786	0.795
	The Firm	0.969	0.889	0.864	0.859	0.862	0.805	0.785	0.817	0.764	0.760	0.790
	Formula 1	0.867	0.732	0.682	0.600	0.571	0.515	0.601	0.646	0.497	0.445	0.527
	Soccer	0.837	0.701	0.642	0.639	0.652	0.610	0.672	0.632	0.651	0.666	0.694
	ARD News	0.967	0.852	0.709	0.602	0.565	0.567	0.535	0.405	0.409	0.333	0.362
	ARD Talk	0.903	0.863	0.796	0.768	0.772	0.719	0.663	0.684	0.634	0.580	0.532
	N3 Talk	0.882	0.840	0.846	0.868	0.878	0.931	0.919	0.943	0.950	1.012	0.985
	Office–Cam	0.607	0.886	0.850	0.858	0.858	0.884	0.927	0.859	0.940	0.973	0.960
Medium	*Jurassic Park I*	0.948	0.821	0.776	0.756	0.722	0.732	0.630	0.633	0.664	0.549	0.579
	Silence of the Lambs	0.997	0.891	0.867	0.866	0.896	0.864	0.849	0.765	0.781	0.736	0.760
	Star Wars IV	0.847	0.846	0.822	0.798	0.787	0.786	0.770	0.785	0.823	0.815	0.803
Low	*Jurassic Park I*	0.881	0.824	0.771	0.752	0.729	0.726	0.628	0.636	0.642	0.538	0.580
	Silence of the Lambs	0.935	0.887	0.863	0.858	0.882	0.867	0.842	0.769	0.777	0.744	0.764
	Star Wars IV	0.770	0.844	0.814	0.795	0.785	0.790	0.769	0.787	0.833	0.817	0.805

4.2 Statistical Analysis of H.263 Traces

In this section we conduct a thorough statistical analysis of the generated H.263 frame size traces. Let N denote the number of frames in a given video trace. Let $X_n,\ n = 1,\ldots,N$, denote the number of bits in frame n, i.e., the frame size of frame n. Let $t_n,\ n = 1,\ldots,N$, denote the frame period (display time) of frame n in msec. Let $T_n,\ n = 1,\ldots,N$, denote the cumulative display time up to (and including) frame n, i.e., $T_n = \sum_{k=1}^{n} t_k$ (define $T_0 = 0$). For illustration Table 5 gives the first ten lines of the trace of the *Silence of the Lambs* encoding with a target bit rate of 256 kbit/s. The trace gives on line $n,\ n = 1,\ldots,N$, the cumulative display time T_{n-1} (up to frame $n-1$), the type (I, P or PB) of frame n, and the frame size X_n in bytes.

As illustrated by the trace file, the T_n's are integer multiples of the basic (reference) frame period $\Delta = 40$ msec of the H.263 encoder. Notice, however, that some frames are skipped by the encoder striving to meet the specified target bit rate [18, p. 67]. This results in variable frame periods. Figure 7 gives the probability mass functions $P(t_n = l \cdot \Delta),\ l = 1,2,\ldots$, of the frame periods of three generated H.263 traces. We observe from the plots the general tendency that smaller target bit rates result in larger frame periods, i.e., more frames are skipped. On

Table 5: Excerpt of H.263 trace file of *Silence of the Lambs* encoding.

0	I	12539
360	P	3981
600	PB	6203
760	PB	5884
1000	PB	6749
1160	PB	6425
1400	PB	7849
1640	PB	5983
1800	PB	6183
2040	PB	7052

the other hand, for VBR encodings, i.e., without specified target bit rate, the H.263 encoder typically does not skip any frames. Nevertheless, as we observe from Figure 7 a), most encoded frames have a frame period of 2Δ. This is because the encoder produces mostly PB frames, i.e., two consecutive frames are encoded as one entity. If no frame is skipped, the PB frame has a frame period of 2Δ when emitted by the encoder; at the decoder, however, the B frame is displayed first for a period of Δ and then the P frame for a period of Δ.

Frame Sizes and Bit Rates

Table 6 gives an overview of the statistics of the frame sizes X_n. (To conserve space we give the statistics for all four target bit rate settings only for the *Jurassic Park I*, *Silence of the Lambs*, and *Star Wars IV* videos. We refer the interested reader to the technical report [27] and for the omitted results.) First, we note that the H.263 encoder meets a given target for the average bit rate of the encoded video stream. To see this recall that the uncompressed YUV video stream has a bit rate of 7,603,200 bit/sec. Also, recall that the compression ratio is defined as the ratio of the sum of the sizes of all unencoded YUV frames of the video to the sum of the sizes of all encoded frames emitted by the encoder. (Keep in mind that the H.263 encoder may (*i*) skip frames and (*ii*) encode two frames into one PB frame; therefore the number of encoded frames N may be smaller than the number of unencoded YUV frames.) To achieve a given target for the average bit rate of the encoded video stream the encoder enforces the same compression ratio for all videos. Even though, for a given target rate all encoded videos have the same average bit rate, their average frame sizes are different. For the 16 kbps target rate, for instance, the *Soccer* encoding has an average size of 655 bytes, while the *Silence of the Lambs* encoding has an average frame size of 370 bytes. Nevertheless, the

Table 6: Overview of frame size statistics of H.263 traces.

Rate	Trace	Comp. ratio YUV:H.263	Mean $\bar{X}$ [byte]	CoV $S_X/\bar{X}$	Peak/Mean $X_{\max}/\bar{X}$
16 kbps	*Jurassic Park I*	476.36	476.36	0.67	20.83
	Silence of the Lambs	476.43	369.85	0.67	33.90
	Star Wars IV	476.43	326.02	0.61	11.32
	Soccer	476.30	654.56	0.60	7.10
64 kbps	*Jurassic Park I*	118.96	1132.02	0.36	7.89
	Silence of the Lambs	118.95	1129.94	0.41	11.10
	Star Wars IV	118.95	1153.32	0.43	7.11
256 kbps	*Jurassic Park I*	29.73	4533.67	0.35	2.61
	Silence of the Lambs	29.73	4453.81	0.39	5.00
	Star Wars IV	29.73	4563.53	0.33	3.58
VBR	*Jurassic Park I*	17.08	3993.31	0.64	4.55
	Silence of the Lambs	25.19	2703.46	0.99	10.27
	Star Wars IV	65.79	1048.21	0.66	8.58
	Mr. Bean	25.00	2662.61	0.60	6.09
	First Contact	44.64	1512.82	0.78	7.68
	From Dusk Till Dawn	19.30	3378.48	0.58	4.81
	The Firm	57.17	1241.80	0.80	7.39
	Formula 1	14.25	3826.30	0.47	3.69
	Soccer	10.18	5583.62	0.50	4.05
	ARD News	20.17	3442.46	0.77	4.45
	ARD Talk	30.70	2374.17	0.55	5.59
	N3 Talk	27.96	2545.62	0.57	5.48
	Office–Cam	84.01	903.78	0.36	5.74

encoder meets the target bit rate by skipping more frames of the *Soccer* video; i.e., the average frame period of the *Soccer* encoding is larger.

Comparing the 256 kbps target rate encodings with the VBR encodings we observe that some VBR encodings have higher compression ratios than the corresponding 256 kbps target rate encodings. The VBR encoding of *Star Wars IV*, for instance, has a compression ratio of 65, while the 256 kbps encoding has a compression ratio of 29.7. The more efficient VBR encoding, however, has a larger variability of the frame sizes. It is important to note that for variable frame period H.263 encoded video, the frame sizes are only one component of the video stream statistics. For the complete picture we need to consider the frame sizes in conjunction with their associated frame periods. Clearly, if the larger frame sizes of the VBR H.263 encodings were associated with larger frame periods, and vice versa, then the larger frame periods could be used to smooth out the larger frames. We shall see shortly that this is to a limited extend possible.

Figure 8 gives the histograms of the frame size X_n for three traces. We observe that the 256 kbps target rate encoding of *Jurassic Park I* has a pronounced bi–modal distribution of the frame sizes. This is because the encoder typically produces (*i*) P frames with an average size of roughly 3 kbytes, and (*ii*) PB frames with an average size of 6 kbytes. Similar observations hold for the depicted frame size histogram of the 16 kbps target rate encoding of *Silence of the Lambs*, as well as the other encodings.

To get the complete picture of the H.263 video stream statistics we define for a given H.263 frame size trace, two different traces that associate the frame sizes X_n with the frame periods t_n. (This will also facilitate the analysis of the H.263 video correlations and long range dependence characteristics.) First, we consider a "stuffed" frame size trace $F_m,\ m = 1, \ldots, T_N/\Delta$, obtained by "stuffing" zeros for the skipped frames into the generated frame size trace $X_n,\ n = 1, \ldots, N$. Formally,

$$F_m = \begin{cases} X_n & \text{for } m = \frac{T_n}{\Delta},\quad n = 1, \ldots, N \\ 0 & \text{for } m \notin \{\frac{T_1}{\Delta}, \ldots, \frac{T_N}{\Delta}\}\ . \end{cases}$$

The stuffed frame size trace reflects the traffic characteristics at the encoder output, where the frames of sizes X_n are emitted at the discrete instants $T_n,\ n = 1, \ldots, N$.

Secondly, we introduce the rate trace $r(t),\ 0 \le t \le T_N$. We convert the discrete frame size trace $X_n,\ n = 1, \ldots, N$, to a fluid flow by transmitting the frame of size X_n at the constant rate X_n/t_n over its frame period, i.e.,

$$r(t) = \frac{X_n}{t_n} \quad \text{for } T_{n-1} < t \le T_n,\ n = 1, \ldots, N.$$

The fluid flow characterization is an approximation of a system that transmits the frame of size X_n in many small packets that are equally spaced over the frame period of length t_n. For infinitesimally small packets this approximation gives a fluid flow of rate X_n/t_n over the frame period of frame n. This fluid flow approximation is popular in teletraffic studies since it significantly simplifies mathematical analyses [40]. Note that $r(t)$ changes its value only at integer multiples of the reference frame period Δ. A more convenient representation of $r(t)$ is thus obtained by "sampling" at Δ–spaced intervals. We define the sampled rate trace as

$$R_m = r(m \cdot \Delta),\ m = 1, \ldots, T_N/\Delta.$$

In the following we study the statistical properties of the "stuffed" frame size traces $F_m,\ m = 1, \ldots, T_N/\Delta$, and the sampled rate traces $V_m,\ m = 1, \ldots, T_N/\Delta$, obtained from the generated H.263 frame size traces. Table 7 gives an overview of the statistics of the "stuffed" frame size traces and the sampled rate traces. First, we compare the frame size statistics from Table 6 with the sampled rate trace statistics in Table 7. We observe that for the target bit

Table 7: Overview of statistics of H.263 "stuffed" frame size traces and sampled rate traces.

		"Stuffed" Frame Size Trace			Sampled Rate Trace		
		Mean	CoV	Peak/Mean	Mean	CoV	Peak/Mean
Rate	Trace	$\bar{F}$ [byte]	$S_F/\bar{F}$	$F_{\max}/\bar{F}$	$\bar{R}$ [kbit/s]	$S_R/\bar{R}$	$R_{\max}/\bar{R}$
16 kbps	*Jurassic Park I*	79.81	2.48	111.96	16	0.35	5.8
	Silence of the Lambs	79.80	2.38	157.14	16	0.37	6.3
	Star Wars IV	79.81	2.15	46.24	16	0.42	6.1
	Soccer	79.82	3.18	58.22	16	0.17	4.6
64 kbps	*Jurassic Park I*	319.59	1.73	27.96	64	0.42	5.7
	Silence of the Lambs	319.60	1.77	39.23	64	0.42	5.7
	Star Wars IV	319.62	1.81	25.66	64	0.45	5.2
256 kbps	*Jurassic Park I*	1278.72	1.72	9.24	256	0.42	5.5
	Silence of the Lambs	1278.61	1.73	17.4	256	0.42	5.9
	Star Wars IV	1278.80	1.72	12.76	256	0.47	5.3
VBR	*Jurassic Park I*	2225.26	1.23	8.16	450	0.69	7.7
	Silence of the Lambs	1509.14	1.60	18.41	300	1.10	17.0
	Star Wars IV	589.63	1.24	15.25	120	0.76	11.0
	Mr. Bean	1520.68	1.17	10.67	300	0.70	11.0
	First Contact	851.63	1.36	13.65	170	0.84	13.0
	From Dusk Till Dawn	1969.69	1.13	8.25	390	0.68	8.2
	The Firm	665.00	1.44	13.79	130	0.89	13.0
	Formula 1	2667.84	0.86	5.29	530	0.47	5.3
	Soccer	3733.26	0.93	6.06	750	0.53	6.1
	ARD News	1884.48	1.38	8.12	380	0.88	7.8
	ARD Talk	1238.43	1.22	10.72	250	0.61	9.4
	N3 Talk	1359.79	1.22	10.26	270	0.62	9.0
	Office–Cam	452.52	1.12	11.47	91	0.38	11.0

rate encodings transmitting each encoded frame at a constant rate (fluid rate) over its frame period significantly reduces the variability of the encoder output. This is because some extremely large frames are associated with large frame periods. Nevertheless, the peak–to–mean ratios of the rate traces with fixed target bit rates are typically five and larger. On the other hand, for VBR encodings the rate traces have larger variability than the frame size traces (see Table 6). This indicates that in the VBR encodings the larger frames are typically associated with the shorter frame periods. We also observe from the statistics of the "stuffed" frame size traces that transmitting each frame at a constant rate over one reference period of length Δ gives extremely variable encoder output for small target bit rates, especially for the 16 kbps and 64 kbps encodings. This is because the encoder skips many frames in these encodings. Thus many zeros are "stuffed" into the traces. As a result the average size of the elements F_m of the stuffed frame size trace decreases, while their variability increases.

As alluded to above, some VBR encodings have significantly smaller average bit rates than the 256 kbps target rate; see, for instance, the *Star Wars IV* encoding. The more efficient VBR encoding, however, entails more variability in the encoded video stream. Loosely speaking, with VBR encoding the encoder produces high output rates (i.e., large frame sizes and short frame periods) when they are needed to encode complex scenes without reducing the video quality. We note, however, that a detailed study of the video stream statistics in conjunction with the perceived video quality is beyond the scope of this article. (For a study of the traffic characteristics in conjunction with the video quality characteristics of encoded video see, for instance, [41]; the work [41] analyzes the traffic statistics and perceived quality of H.261 and MPEG–1 encodings of a 9 minute *StarTrek* sequence, a 4 minute *Raiders* sequence, and a 4 minute *Terminator 2* sequence.)

Figure 9 gives the sampled rate trace R_m as a function of the index m (in reference frame periods of length Δ) for the generated H.263 traces. We observe from Figure 9a) that the rate trace of the VBR encoding of *Star Wars IV* exhibits fast time scale fluctuations that are superimposed on underlying slow time scale fluctuations. We observed that this is a typical characteristic of H.263 encodings without a specified target bit rate. We see from Figure 9b) and c) that the target bit rate encodings do exhibit the fast time scale fluctuations as well, but they do not exhibit the underlying slow time scale fluctuations. However, there are occasional "spikes" in the rate trace that are roughly four to six times higher than the average bit rate. These spikes can be effectively smoothed out by averaging the trace over (non–overlapping) blocks of roughly ten or more reference frame periods. Smoothing is also highly effective in reducing the fast time scale fluctuations. For the 256 kbps *Jurassic Park I* encoding, for instance, the fast time scale fluctuations are approximately between 100 kbps and 650 kbps without smoothing. As depicted in Figure 10b), with smoothing over 12 reference frame periods (i.e., with an aggregation level of $a = 12$) the fast time scale fluctuations are approximately between 210 kbps and 340 kbps, that is, the range of the fluctuations is roughly four times smaller. We observe from Figure 10c) that with smoothing over 50 reference frame periods the fast time scale fluctuations are approximately between 245 kbps and 275 kbps. The rate trace settles down with smaller fluctuations around the target bit rate. This trend continues for larger smoothing intervals; the trace settles with smaller and smaller fluctuations around the target bit rate. Similar observations hold for the other H.263 encodings with a specified target bit rate. The H.263 encodings without a specified target bit rate, on the other hand, behave very differently. They exhibit significant fluctuations even when smoothed over long intervals. This is illustrated for the VBR encoding of *Star Wars IV* smoothed over 500 reference frame periods in Figure 10a).

Correlations and Long Range Dependence

Figure 11 gives the autocorrelation coefficient of the "stuffed" frame size trace $\rho_F(k)$ and the autocorrelation coefficient of the sampled rate trace $\rho_R(k)$ as a function of the lag k (in reference frame periods of length Δ) for the generated H.263 traces over 14 reference frame periods of length Δ. We observe that the "stuffed" frame size traces have rather "jerky" autocorrelation functions. This is because the zeros in the "stuffed" traces give negative spikes. The sampled rate traces, on the other hand, have smooth autocorrelation functions. To get a better picture of the long term correlations we give in Figure 12 the autocorrelation functions over 500 reference frame periods. We observe that the autocorrelation function of the VBR encoding (i.e., without specified target rate) of *Star Wars IV* decays very slowly; for a lag of $d = 500\Delta$ the correlation coefficient of the sampled rate trace is roughly 0.18. (The "jerky" autocorrelation function of the "stuffed" frame size trace gives rise to the light gray shading in this plot.) The autocorrelations of the depicted target bit rate encodings, on the other hand, decay quickly to zero.

Table 8 gives the Hurst parameters of the sampled rate traces as a function of the aggregation level a. Figure 13 gives pox plots of R/S for aggregation levels of $a = 1$ and $a = 12$. We notice from the pox plots given here and in the technical report [27] that two problems arise when applying the R/S statistic to the H.263 traces. First, some pox plots for the aggregation level $a = 1$ have outliers for small lags d. One strategy could have been to remove those outliers; this would have given larger estimates for the Hurst parameter for the aggregation level $a = 1$. We chose not to do so in order to keep the least–squares fit estimation simple and automated. In interpreting the results in Table 8 we ignore the column $a = 1$ and focus on the larger aggregation levels instead. Secondly, we observed that the pox plots for aggregation levels of $a = 200$ and larger (not shown here because of space constraints) for encodings with a specified target bit rate, typically do not settle down around a straight "street". We suspect that this is due to the fact that the H.263 encoder typically skips many frames to meet a specified rate target. As a result these traces might not have a sufficiently large number of values to estimate the Hurst parameter for large aggregation levels.

Nevertheless, we observe that all VBR encodings have Hurst parameters above 0.7 for all aggregation levels of $a = 12$ and higher. This indicates a high degree of long range dependence in the VBR traces. We also observe from the table that the encodings with a specified target bit rate have Hurst parameter above 0.7 for the aggregation levels $a = 12$, $a = 50$, and $a = 100$. For aggregations levels of $a = 200$ and larger, however, the estimated Hurst parameter

Table 8: Hurst parameters of H.263 sampled rate traces estimated from pox diagram of R/S as a function of the aggregation level a.

		Aggregation level a [reference frame periods Δ]										
Rate	Trace	1	12	50	100	200	300	400	500	600	700	800
16 kbps	*Jurassic Park I*	0.945	0.930	0.843	0.770	0.630	0.409	0.581	0.413	0.083	0.352	0.503
	Silence o/t Lambs	0.262	0.657	0.835	0.721	0.594	0.405	0.488	0.420	0.091	0.215	0.361
	Star Wars IV	0.947	0.944	0.872	0.762	0.621	0.380	0.534	0.391	0.157	0.384	0.511
64 kbps	*Jurassic Park I*	1.005	0.963	0.832	0.748	0.608	0.365	0.503	0.445	0.209	0.323	0.527
	Silence o/t Lambs	0.459	0.985	0.886	0.790	0.608	0.400	0.465	0.383	0.101	0.272	0.467
	Star Wars IV	0.963	0.952	0.903	0.802	0.620	0.348	0.509	0.400	0.150	0.388	0.465
256 kbps	*Jurassic Park I*	0.815	0.960	0.883	0.770	0.575	0.353	0.449	0.360	0.129	0.318	0.497
	Silence o/t Lambs	0.961	0.950	0.867	0.757	0.598	0.407	0.509	0.422	0.112	0.226	0.435
	Star Wars IV	0.777	0.960	0.872	0.743	0.585	0.326	0.495	0.418	0.111	0.328	0.497
VBR	*Jurassic Park I*	0.863	0.862	0.902	0.900	0.867	0.816	0.770	0.785	0.805	0.817	0.867
	Silence o/t Lambs	0.575	0.758	0.778	0.815	0.793	0.730	0.753	0.803	0.727	0.738	0.755
	Star Wars IV	0.633	0.893	0.882	0.882	0.882	0.845	0.859	0.864	0.856	0.876	0.865
	Mr. Bean	0.860	0.811	0.803	0.816	0.836	0.842	0.816	0.802	0.753	0.765	0.748
	First Contact	0.650	0.855	0.874	0.870	0.871	0.877	0.888	0.909	0.896	0.944	0.875
	Dusk Till Dawn	0.671	0.852	0.841	0.867	0.915	0.899	0.915	0.877	0.875	0.893	0.891
	The Firm	0.803	0.812	0.854	0.847	0.878	0.831	0.849	0.848	0.844	0.879	0.820
	Formula 1	0.663	0.877	0.840	0.876	0.897	0.834	0.899	0.882	0.856	0.980	1.036
	Soccer	0.726	0.853	0.935	0.925	0.910	0.911	0.934	0.912	0.882	0.897	0.925
	ARD News	0.675	0.702	0.805	0.846	0.914	0.887	1.010	0.876	0.942	0.931	0.931
	ARD Talk	0.791	0.823	0.877	0.908	0.861	0.801	0.813	0.870	0.859	0.974	0.923
	N3 Talk	0.642	0.894	0.821	0.776	0.714	0.705	0.710	0.692	0.619	0.674	0.687
	Office–Cam	0.720	0.732	0.758	0.784	0.783	0.775	0.778	0.753	0.832	0.841	0.820

is typically around 0.5 or smaller. This appears to corroborate our earlier observation that the rate traces of H.263 encodings with a specified target bit rate settle down around the target bit rate when smoothed over long intervals (see Figure 10b) and c)). (Similarly, it has been observed in [42] that rate control may eliminate long range dependence in encoded video streams.) However, more studies on the long range dependence properties of H.263 traces are needed.

5 Conclusion

In this article we have presented and studied a publicly available library of frame size traces of MPEG–4 and H.263 encoded videos. We have encoded over ten videos of 60 minutes length each. For each video we have generated MPEG–4 and H.263 encodings at several different quality levels. All in all, we have generated and studied over 70 hours worth of video traces.

We have conducted a detailed statistical analysis of the generated traces. For the analysis of the H.263 encodings, which have variable frame periods, we have introduced the notion of a *rate trace.* The rate trace facilitates the analysis of the H.263 frame sizes in conjunction with their associated frame periods. We have found that the traces are typically highly variable in their frame sizes and bit rates; especially the traces of low quality encodings. Also, many of the traces show clear indications of long range dependence properties.

In our ongoing work we are expanding our video trace study by producing and analyzing MPEG–4 and H.263 encodings of more videos. We are also generating and studying MPEG–4 encodings with multiple Video Objects and multiple Video Object Layers. Moreover, we are encoding videos using the H.263+ encoder, which incorporates advanced motion prediction and enhanced PB frames.

Acknowledgment: We are grateful to Prof. Adam Wolisz for providing the environment that allowed us to pursue the work presented in this article. We are grateful to Moncef Abda Ben, Thomas Kroener, Mohammad Kandil, and Ahmed Salih for assisting in the recording and encoding of the videos during the student project at TKN in summer 2000. We gratefully acknowledge insightful discussions, borrowed video cassettes, and encouragement from Jean–Pierre Ebert, Enno Ewers, Andreas Festag, Andreas Koepsel, Rolf Morich, and Stephan Rein. We are grateful to Guido Heising for explaining the intricacies of the MOMUSYS MPEG–4 software. Sethuraman Panchanathan provided valuable background on digital video and video compression. We are grateful to three anonymous reviewers whose comments help greatly in improving the presentation of this article.

Appendix

In this appendix we review the statistical definitions and methods used in the analysis of the generated frame size traces. Recall that N denotes the number of frames in a given trace. Also recall that $X_n,\ n = 1, \ldots, N$, 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

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

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

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

A computationally more convenient expression for S_X^2 is

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

The coefficient of variation CoV is defined as

$$CoV = \frac{S_X}{\bar{X}}.$$

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

$$X_{\max} = \max_{1 \le n \le N} X_n.$$

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

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

These statistics are estimated in analogous fashion for the GoP size traces, the "stuffed" frame size traces, and the sampled rate traces. We refer the reader to [43] for more details on these definitions.

R/S Statistic

We use the R/S statistic [44, 39, 6] 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 9. The R/S statistic $R(t_i, d)/S(t_i, d)$ is computed for logarithmically spaced values of the lag k, starting with $d = 10$. 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 $(t_i - 1) + d \le N$, 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 $a \ge 1$. The aggregated trace $X_n^{(a)}$, $n = 1, \ldots, N/a$, is obtained from the original trace X_n, $n = 1, \ldots, N$, by averaging over non–overlapping blocks of length a, i.e.,

$$X_n^{(a)} = \frac{1}{a} \sum_{j=(n-1)a+1}^{na} X_j.$$

Table 9: Algorithm for pox diagram of R/S.

1.	**For** $d = 10,\ 20,\ 40,\ 80, \ldots$ **do**
2.	$\quad I = K + 1 - \lceil \frac{dK}{N} \rceil$
3.	$\quad$ **For** $i = 1, \ldots, I$ **do**
4.	$\quad\quad t_i = (i-1)\frac{N}{K} + 1$
5.	$\quad\quad \bar{X}(t_i, d) = \frac{1}{d}\sum_{j=1}^{d} X_{t_i+j}$
6.	$\quad\quad S^2(t_i, d) = \frac{1}{d}\sum_{j=1}^{d}[X_{t_i+j} - \bar{X}(t_i, d)]^2$
7.	$\quad\quad R(t_i, d) = \max\{0,\ \max_{1 \le k \le d} W(t_i, k)\} - \min\{0,\ \min_{1 \le k \le d} W(t_i, k)\}$
8.	$\quad\quad\quad W(t_i, k) = \left(\sum_{j=1}^{k} X_{t_i+j}\right) - k\bar{X}(t_i, d)$
9.	$\quad\quad$ plot point $\left(\log d,\ \log \frac{R(t_i,d)}{S(t_i,d)}\right)$

References

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

[2] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems. Technical Report 101, University of Wuerzburg, Institute of Computer Science, Am Hubland, 97074 Wuerzburg, Germany, February 1995.

[3] M. Krunz, R. Sass, and H. Hughes. Statistical characteristics and multiplexing of MPEG streams. In *Proceedings of IEEE Infocom '95*, pages 455–462, April 1995.

[4] W-C. Feng. *Video–on–Demand Services: Efficient Transportation and Decompression of Variable Bit Rate Video.* PhD thesis, University of Michigan, April 1996.

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

[6] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long–range dependence in variable–bit–rate video traffic. *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, February/March/April 1995.

[7] M. Frey and S. Nguyen-Quang. A gamma–based framework for modeling variable–rate MPEG video sources: The GOP GBAR model. *IEEE/ACM Transactions on Networking*, 8(6):710–719, December 2000.

[8] J.R. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Use of alpha–stable self-similar stochastic processes for modeling traffic in broadband networks. *Performance Evaluation*, 40(1–3):71–98, March 2000.

[9] N.G. Duffield, K.K. Ramakrishnan, and A.R. Reibman. SAVE: An algorithm for smoothed adaptive video over explicit rate networks. *IEEE/ACM Transactions on Networking*, 6(6):717–728, December 1998.

[10] I. Dalgic and F.A. Tobagi. Performance evaluation of ATM networks carrying constant and variable bit rate video traffic. *IEEE Journal of Selected Areas in Communications*, 15(6):1115–1131, 1997.

[11] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A simple and efficient service for multiple time–scale traffic. *IEEE/ACM Transactions on Networking*, 5(6):741–755, 1997.

[12] P. Jelenkovic, A. Lazar, and N. Semret. The effect of multiple time scales and subexponentiality in MPEG video streams on queueing behavior. *IEEE Journal of Selected Areas in Communications*, 15(6):1052–1071, 1997.

[13] J.D. Salehi, Z.L. Zhang, J. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end–to–end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking*, 6(4):397–410, August 1998.

[14] D.E. Wrege, E.W. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for VBR video in packet–switching networks: Fundamental limits and practical trade–offs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

[15] A. Baiocchi, F. Cuomo, and S. Bolognesi. IP QoS delivery in a broadband wireless local loop: MAC protocol definition and performance evaluation. *IEEE Journal on Selected Areas in Communications*, 18(9):1608–1622, September 2000.

[16] M. Ma and M. Hamdi. Providing deterministic quality–of–service guarantees on WDM optical networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2072–2083, October 2000.

[17] F. Halsall. *Multimedia Communications: Applications, Networks, Protocols, and Standards*. Addison–Wesley, 2001.

[18] A. Puri and T. Chen. *Multimedia Systems, Standards, and Networks*. Marcel Dekker, New York, 2000.

[19] R. Koenen (Editor). Overview of the MPEG–4 standard, ISO/IEC 14496, May/June 2000.

[20] R. Koenen. MPEG–4 multimedia for our time. *IEEE Spectrum*, 36(2):26–33, February 1999.

[21] L. D. Soares and F. Pereira. MPEG–4: A fexible coding standard for the emerging mobile multimedia applications. Technical report, MOMUSYS Project, 1999.

[22] ITU-T/SG15. Recommendation H.263, video coding for low bitrate communication, 1996.

[23] K. Rijkse. H.263: Video coding for low–bit–rate communication. *IEEE Communications Magazine*, 34(12):42–45, December 1996.

[24] ITU-T. Recommendation H.261, video codec for audio–visual services at 64 – 1920 kbit/s, 1993.

[25] J. Walter. bttvgrab. `http://www.garni.ch/bttvgrab/`.

[26] G. Heising and M. Wollborn. MPEG–4 version 2 video reference software package, ACTS AC098 mobile multimedia systems (MOMUSYS), December 1999.

[27] F. Fitzek and M. Reisslein. MPEG–4 and H.263 traces for network performance evaluation (extended version). Technical Report TKN–00–06, Technical University Berlin, Dept. of Electrical Eng., Germany, October 2000. Traces available at `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` and `http://www.eas.asu.edu/trace`.

[28] K. O. Lillevold. tmn, version 2.0, June 1996.

[29] B. Ryu. Modeling and simulation of broadband satellite networks — Part II: Traffic modeling. *IEEE Communications Magazine*, 37(7):48–56, July 1999.

[30] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. In *Proceedings of ACM Multimedia*, San Francisco, CA, April 1995.

[31] K. Dolzer and W. Payer. Two classes — sufficient QoS for multimedia traffic? In *Proceedings COST 257 Symposium*, Oslo, Norway, May 2000.

[32] 3rd Generation Partnership Project (3GPP). Technical Specification Group Services and System Aspects; QoS Concept and Architecture; TS 23.107; Release 4.

[33] A. Adas and A. Mukherjee. On resource management and QoS guarantees for long range dependent traffic. In *Proceedings of IEEE Infocom '95*, Boston, MA, April 1995.

[34] N. G. Duffield, J. T. Lewis, and N. O'Connell. Predicting quality of service for traffic with long–range fluctuations. In *Proceedings of IEEE ICC '95*, Seattle, WA, April 1995.

[35] N. Likhanov, B. Tsybakov, and N. D. Georganas. Analysis of an ATM buffer with self–similar (fractal) input traffic. In *Proceedings of IEEE Infocom '95*, Boston, MA, April 1995.

[36] M. Parulekar and A. M. Makowski. Buffer overflow probabilities for a multiplexer with self–similar input. In *Proceedings of IEEE Infocom '96*, San Francisco, CA, March 1996.

[37] K. Park and W. Willinger. *Self–Similar Network Traffic and Performance Evaluation.* Wiley, 2000.

[38] B. Ryu and A. Elwalid. The importance of long–range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proceedings in ACM SIGCOMM*, pages 3–14, Stanford, CA, August 1996.

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

[40] J. Roberts, U. Mocci, and J. Virtamo (Eds.). *Broadband Network Traffic: Performance Evaluation and Design of Broadband Multiservice Networks, Final Report of Action COST 242, (Lecture Notes in Computer Science, Vol. 1155).* Springer Verlag, 1996.

[41] I. Dalgic and F. A. Tobagi. Characterization of quality and traffic for various video encoding schemes and various encoder control schemes. Technical Report CSL–TR–96–701, Stanford University, Departments of Electrical Engineering and Computer Science, August 1996.

[42] M. Hamdi, J. W. Roberts, and P. Rolin. Rate control for VBR video coders in broad–band networks. *IEEE Journal on Selected Areas in Communications*, 15(6):1040–1051, August 1997.

[43] A. M. Law and W. D. Kelton. *Simulation, Modeling, and Analysis.* McGraw–Hill, second edition, 1991.

[44] 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*, pages 69–99, 1979.

Frank Fitzek is working towards his Ph.D. in Electrical Engineering in the Telecommunication Networks Group at the Technical University Berlin, Germany. He received his Dipl.–Ing. degree in electrical engineering for the University of Technology — Rheinisch–Westfälisch Technische Hochschule (RWTH) — Aachen, Germany, in 1997. His research interests are in the areas of multimedia streaming over wireless links and Quality of Service support in wireless CDMA systems.

Martin Reisslein is an Assistant Professor in the Department of Electrical Engineering at Arizona State University, Tempe. He is affiliated with ASU's Telecommunications Research Center. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994-1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin. While in Berlin he was teaching courses on performance evaluation and computer networking at the Technical University Berlin. He has served on the Technical Program Committees of IEEE Infocom, IEEE Globecom, and the IEEE International Symposium on Computer and Communications. He has organized sessions at the IEEE Computer Communications Workshop (CCW). His research interests are in the areas of Internet Quality of Service, wireless networking, and optical networking. He is particularly interested in traffic management for multimedia services with statistical Quality of Service in the Internet and wireless communication systems.

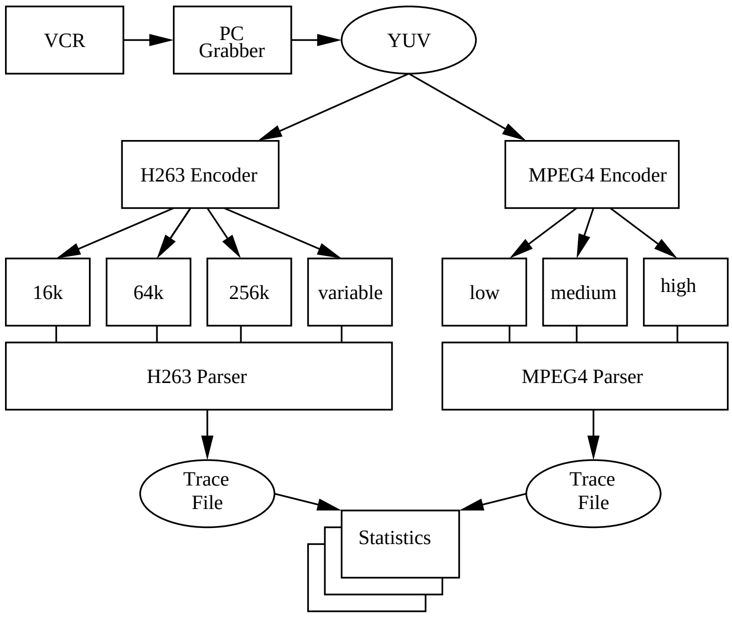
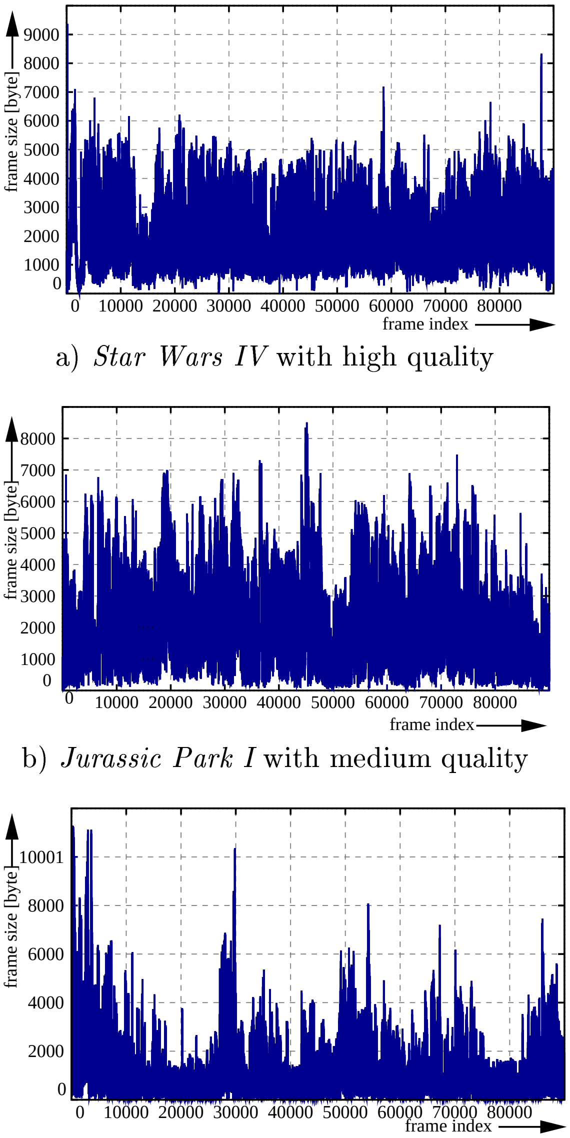


Figure 1: Generation of frame size traces.

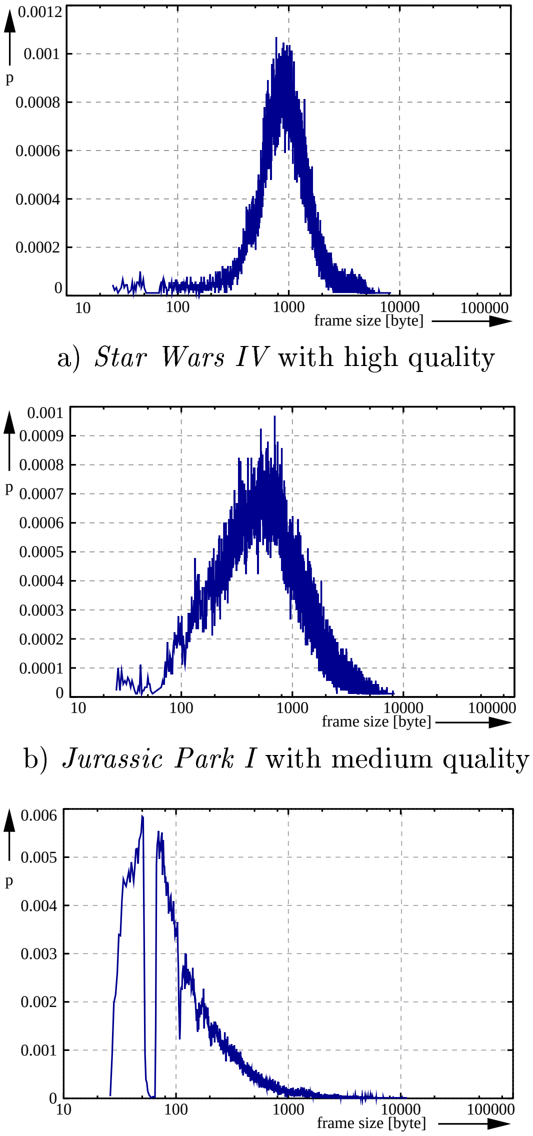


a) *Star Wars IV* with high quality

b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

Figure 2: MPEG–4 frame size traces.

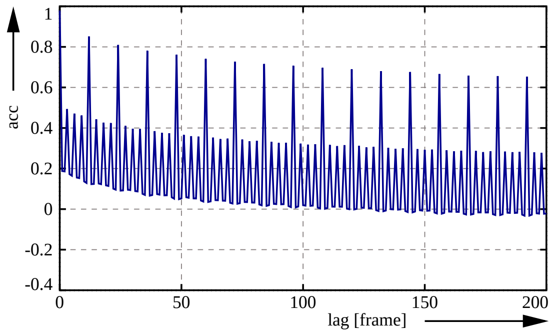


a) *Star Wars IV* with high quality

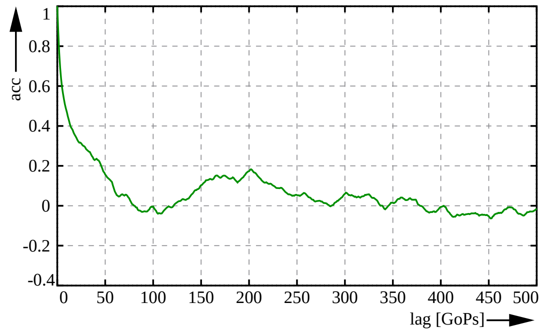
b) *Jurassic Park I* with medium quality

c) *Silence of the Lambs* with low quality

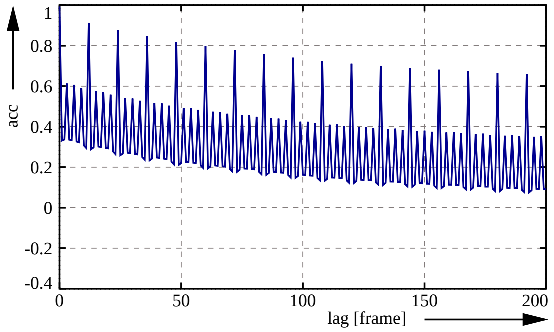
Figure 3: MPEG–4 frame size histograms.



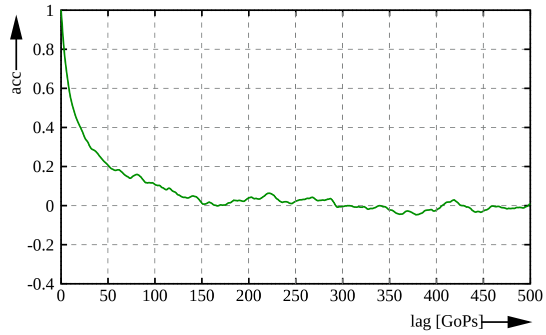
a) *Star Wars IV* with high quality



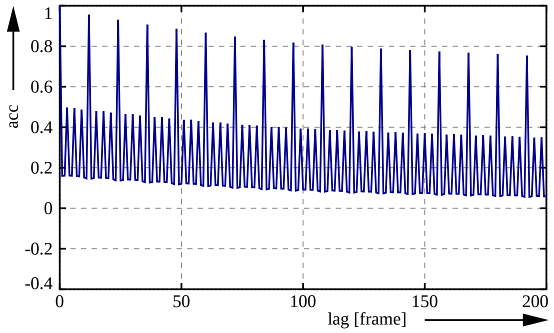
a) *Star Wars IV* with high quality



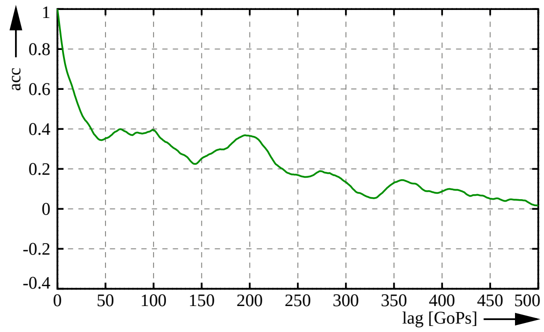
b) *Jurassic Park I* with medium quality



b) *Jurassic Park I* with medium quality



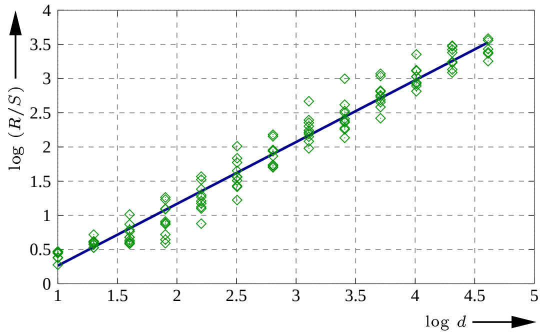
c) *Silence of the Lambs* with low quality



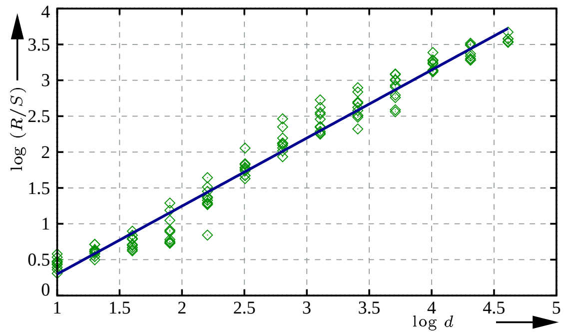
c) *Silence of the Lambs* with low quality

Figure 4: Autocorrelation of MPEG–4 frame size traces.

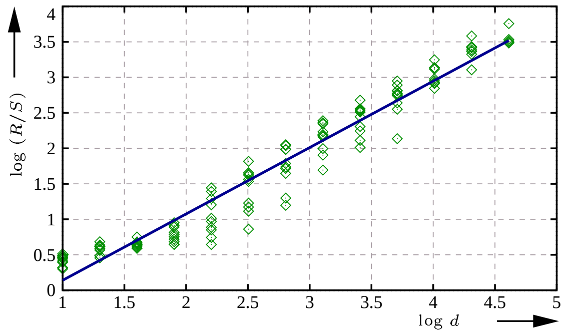
Figure 5: Autocorrelation of MPEG–4 GoP size traces.



a) *Star Wars IV* with high quality, $H = 0.903$

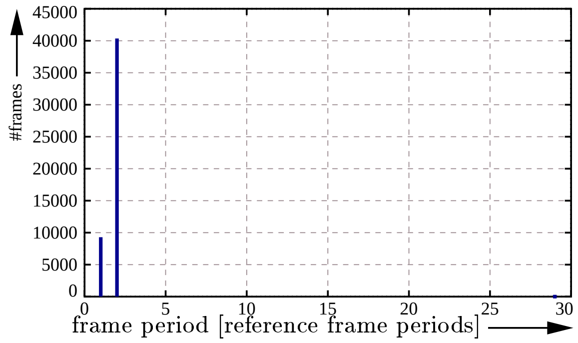


b) *Jurassic Park I* with medium quality, $H = 0.948$

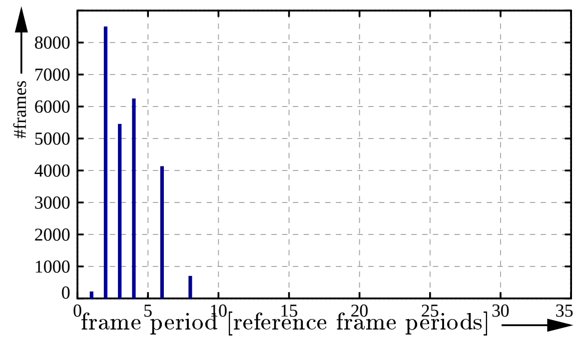


c) *Silence of the Lambs* with low quality, $H = 0.935$

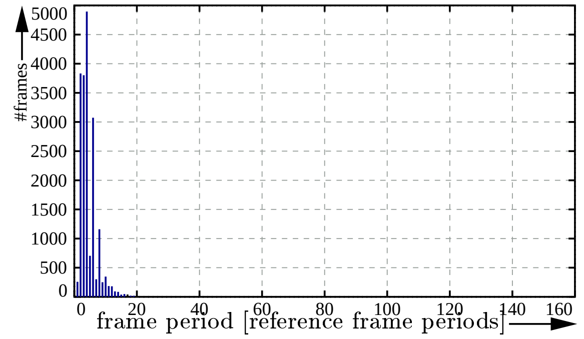
Figure 6: Pox plots of R/S showing street of slope H for MPEG–4 traces with aggregation level $a = 1$.



a) *Star Wars IV* without target rate

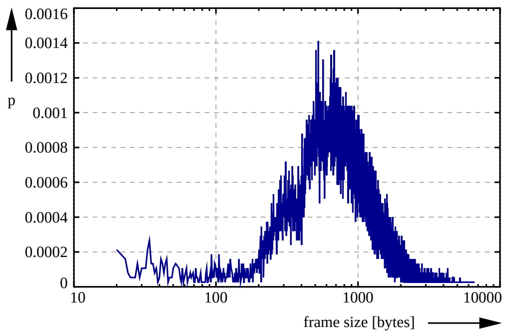


b) *Jurassic Park I* with target rate 256 kbps

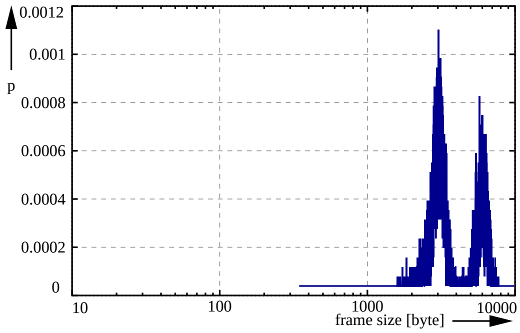


c) *Silence of the Lambs* with target rate 16 kbps

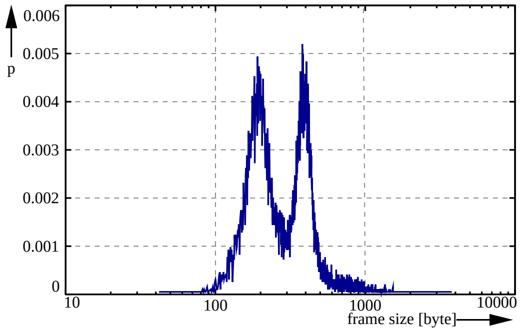
Figure 7: Probability mass functions of frame periods of H.263 traces.



a) *Star Wars IV* without target rate

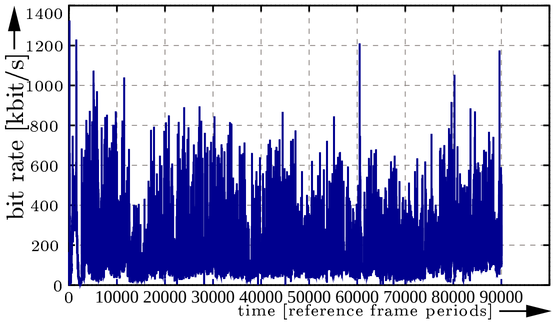


b) *Jurassic Park I* with target rate 256 kbps

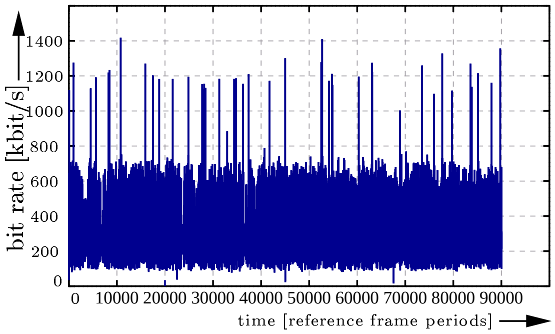


c) *Silence of the Lambs* with target rate 16 kbps

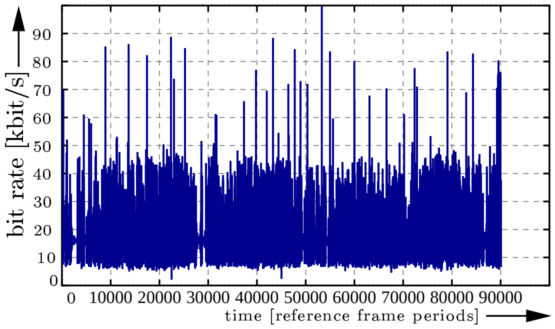
Figure 8: H.263 frame size histograms.



a) *Star Wars IV* without target rate

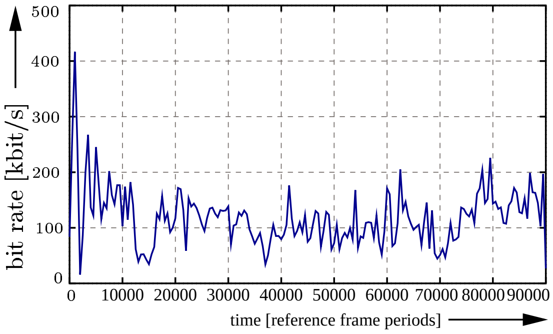


b) *Jurassic Park I* with 256 kbps target rate

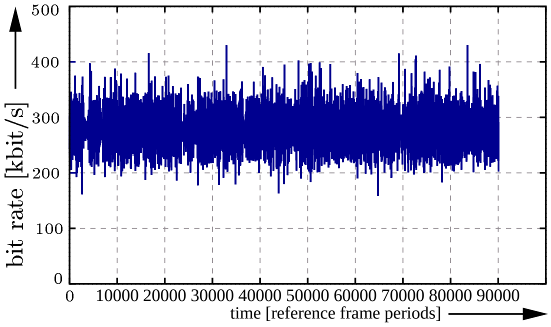


c) *Silence of the Lambs* with 16 kbps target rate

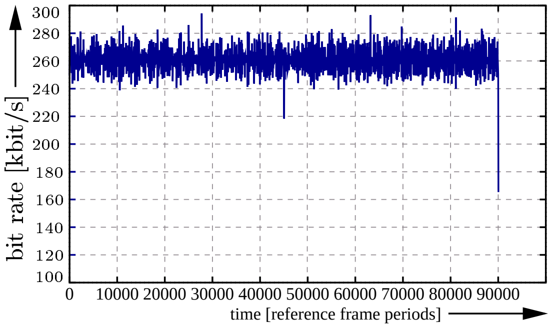
Figure 9: Sampled rate traces R_m of H.263 encodings (unsmoothed).



a) *Star Wars IV* without target rate
smoothed over 500 ref. frame periods

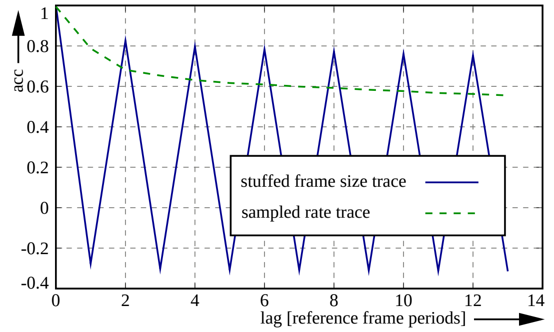


b) *Jurassic Park I* with 256 kbps target rate
smoothed over 12 ref. frame periods

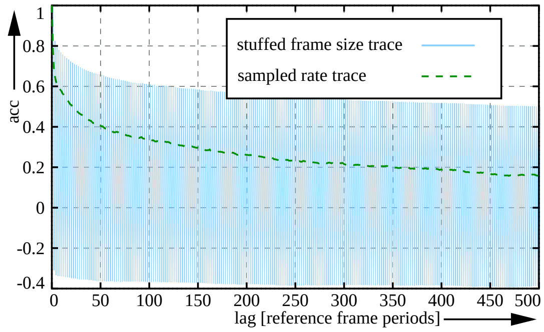


c) *Jurassic Park I* with 256 kbps target rate
smoothed over 50 ref. frame periods

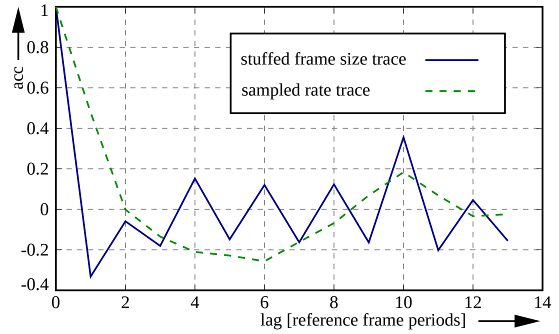
Figure 10: Smoothed sampled rate traces of H.263 encodings.



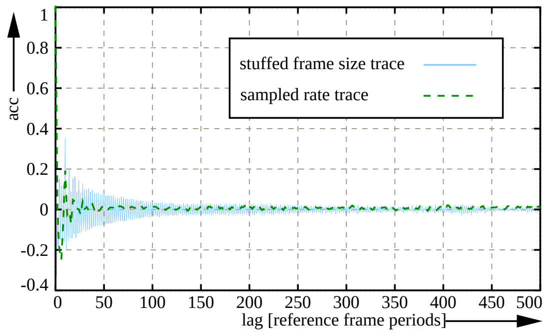
a) *Star Wars IV* without target rate



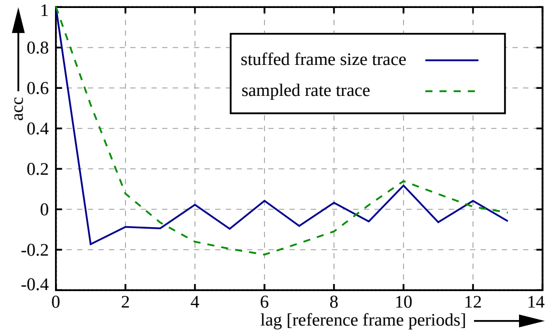
a) *Star Wars IV* without target rate



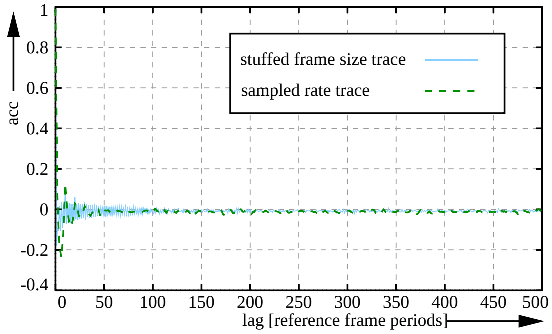
b) *Jurassic Park I* with 256 kbps target rate



b) *Jurassic Park I* with 256 kbps target rate



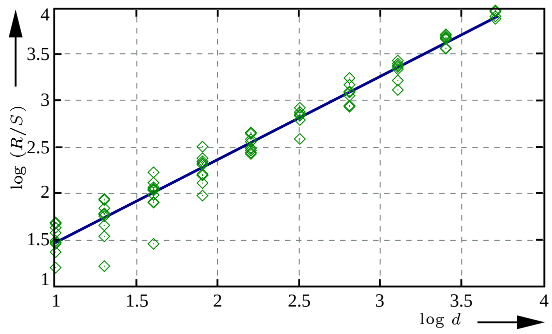
c) *Silence of the Lambs* with 16 kbps target rate



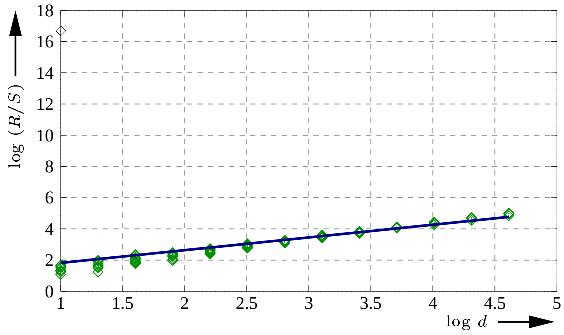
c) *Silence of the Lambs* with 16 kbps target rate

Figure 11: Autocorrelation of H.263 "stuffed" frame size traces and sampled rate traces over 14 reference frame periods.

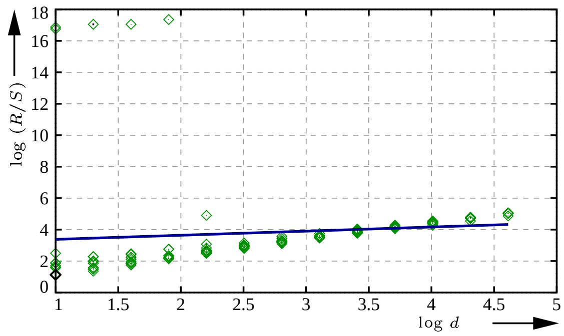
Figure 12: Autocorrelation of H.263 "stuffed" frame size traces and sampled rate traces over 500 reference frame periods.



a) *Star Wars IV* without target rate with $a = 12$, $H = 0.893$



b) *Jurassic Park I* with 256 kbps target rate with $a = 1$, $H = 0.815$



c) *Silence of the Lambs* with 16 kbps target rate with $a = 1$, $H = 0.262$

Figure 13: Pox plots of R/S showing street of slope H for H.263 sampled rate traces with aggregation levels of $a = 1$ and $a = 12$.