# H.26L Pre-Standard Evaluation

F. Fitzek, P. Seeling, M. Reisslein*

acticom GmbH – mobile networks
R & D Group
Germany
[fitzek|seeling]@acticom.de

Arizona State University
Department of Electrical Engineering
USA
reisslein@asu.edu

30. November 2002

Technical Report acticom-02-002

In this report we give an overview of our first evaluations of the upcoming H.26L video encoding standard of the ITU–T. Since this standard is still in the development phase, all of the results presented here are to be seen as preliminary, as is the introduction to the standard itself. The video encodings analyzed in this report have been generated with a preliminary version of the H.26L encoder. The key characteristics of the final H.26L encoder are expected to be very close to the preliminary coder used in our experiments. The traffic characterisations given in this report give therefore a very close approximation of the video traffic and quality produced by the final encoder. In this report we first outline the current state of the standard, our measurement setup, and give an introduction to the analyzed statistical measures. We then present and interpret the statistical characteristics of the H.26L encoded video. We conclude by stating the current problems and outline future work.

*The work of M. Reisslein is supported in part by the National Science Foundation through Grant No. Career ANI-0133252 and Grant No. ANI-0136774. Any opinions, findings, and conclusions or recommendations expressed in this material are these of the authors and do not necessarily reflect the views of the National Science Foundation.

# Contents



# List of Tables

# 1 Introduction

H.26L is currently in the development status and is due to become standard according to the ITU–T and ISO/MPEG groups in late 2002. It will then become part of the H.2xx, H.3xx and MPEG standard families. H.26L is expected to replace its predecessors H.261 and H.263, which were widely used in integrated circuits for telephone and video equipment for ISDN services over telephone networks. H.26L has been designed with packet-switched networks in mind and has in its current implementation a complete network adaptation layer. Due to the joint development of the ITU and ISO bodies, it is also known as H.264, to furthermore express these joint efforts. The development goal — to reach DVD quality video with data rates of about 1 MBit/s — is also referred to as "Advanced Video Coding" (AVC). Standardization bodies in Europe, such as the DVB-Consortium, as well as its American counterpart, the Advanced Television Systems Committee (ATSC), are considering to employ H.26L in their respective standards. H.26L is also widely viewed as a promising standard for wireless video streaming and is expected to largely replace MPEG–4 and H.263+.

Given the expected popularity and widespread use of the new H.26L video encoding standard, the bandwidth demands of H.26L encoded video need to be taken into consideration when designing future wired and wireless (e.g., wireless LAN and 3G) networks. It is therefore very important to understand the characteristics of the traffic produced by this new standard. In this report we examine the traffic (bit rate) characteristics of video encoded with the H.26L encoder.

We have generated traces which contain the sizes (in byte) of the encoded video frames. Our video traces serve as the basis for our statistical analysis of the H.26L video traffic. The traces may also be used by other researchers as a basis for the development of models of the H.26L video traffic. The traces may also be used to evaluate networking protocols and mechanisms with trace–driven simulations of the H.26L video traffic.

This report has four main parts. We first give a brief introduction to the latest proposal for the H.26L standard. We then describe the general setup of our video trace generation and give a brief statistical evaluation of the traces, followed by a description of the currently existing problems. We finish with an outlook of our future work.

# 2 Video Basics

The digitized video is generated by sampling the analog video signal as it is received by the A/D converter hardware. The rate of pictures per second (or frames per second, fps) that is generated is different for the two major standards, PAL (Phase Alternation by Line) has 25 fps and NTSC (National Television Standards Committee) has 30 fps. The main picture formats currently used for video compression studies are CIF and QCIF. The CIF picture size is 352 columns by 288 lines, the QCIF format is 176x144 (i.e., half the size of CIF in each dimension). The video signal is sampled according to the picture size and with respect to the sensivity characteristics of the human eye. In contrast to the RGB format — which generates any color by combining red, green and blue components — the YUV format combines the luminance component and the two chrominance components hue and intensity (saturation). Since the eye is far more sensitive to the luminance level than to coloring information, the YUV formats subsample the chrominance information. (Although YUV is often referred to as lossless (or raw) picture information, when sampling into YUV, some chrominance information is lost.) The two most common YUV sampling formats are 4:1:1 as illustrated in Figure 1 and 4:2:0 as illustrated in

Figure 2. Both formats store one set of hue and intensity samples for four luminance samples (pixels), i.e., 176x144 luminance samples and 88x72 samples for each hue and intensity in case of a QCIF format frame. The 4:1:1 subsampling format stores one set of hue and intensity samples for four luminance samples grouped in a row, whereas the 4:2:0 subsampling format stores one set of hue and intensity samples for four luminance samples grouped in a rectangle. The 4:2:0 format is most commonly used since is has proven to give the best trade–off between sampling efficiency and accuracy.

Each sample is typically stored in an 8–bit value. Thus, the size of one YUV frame with 4:2:0

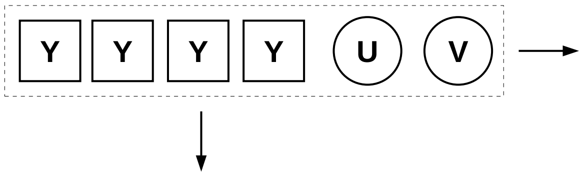


Figure 1: YUV 4:1:1 subsampling

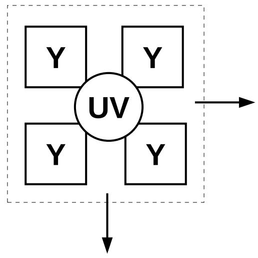


Figure 2: YUV 4:2:0 subsampling

(or 4:1:1) chrominance subsampling in the QCIF format is

$$176 \cdot 144 \cdot \left(8 \text{ bit} + \frac{2 \cdot 8 \text{ bit}}{4}\right) = 304128 \text{ bit} = 38016 \text{ byte}. \tag{1}$$

Similarly, the size of of one YUV frame in the CIF format is

$$352 \cdot 288 \cdot \left(8 \text{ bit} + \frac{2 \cdot 8 \text{ bit}}{4}\right) = 1216512 \text{ bit} = 152064 \text{ byte}. \tag{2}$$

The corresponding bit rates, with the NTSC frame rate of 30 frames per second are

$$T_{x,QCIF} = 30 \text{ Hz} \cdot 304128 \text{ bit} = 9123840 \text{ bps} \approx 9.12 \text{ Mbps} \tag{3}$$

for QCIF and

$$T_{x,CIF} \approx 35.5 \text{ Mbps} \tag{4}$$

for CIF if no encoding is applied. As is obvious by these rates, compression schemes have to be employed in order to achieve data rates suitable for transmission over wireless networks.

# 3 The H.26L Standard

We used a development version of the future H.26L standard implementation software, the JM rev. 2 (dated April 11th, 2002) in our experiments. Since the standard is not yet final and discussions concerning the header and other parts of the encoded video bitstream are currently ongoing, we will focus on the main coding algorithm. We leave out the other details (such as header formats) for further examination when the standard has been adopted. The upcoming H.26L standard differs from its predecessors (the ITU–T H.26x video standard family and the MPEG standards MPEG–2 and MPEG–4) in providing a high compression video coding layer (VCL) for storage optimization as well as a network adaption layer (NAL) for the packetization of

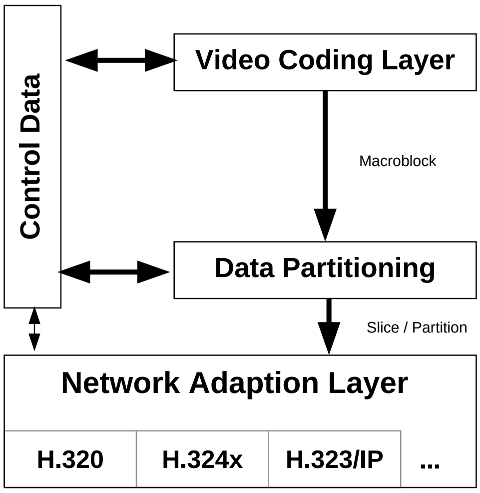


Figure 3: Block diagram of an H.26L coder.

the encoded bitstream according to transmission requirements [14]. An overview of these layers is given in Figure 3. The network adaption layer will be left out in the following discussion, since its functionality will vary according to the underlying network type (e.g. 802.3, 802.11x, UMTS, and others) and will not be a subject of our evaluations. The encoded bitstream will therefore not be sliced (partitioned) further but remain a single sequence. (Slices represent independent coding units that can be decoded without referencing other slices of the same frame. They consist typically of several consecutive macroblocks. Slicing can be utilized to achieve a higher error–robustness.)

The standard is based on a block–oriented and motion–compensating hybrid transformation process. Similar to other video coding standards [14, 15, 6], the standard will only specify the decoding process to allow for maximum customization possibilities in the encoding routines. The encoding is done on a macroblock level. Each CIF format picture is subdivided into 18 (lines) $\times$ 22 (rows) macroblocks, for a total of 396 macroblocks. As illustrated in Figure 4, each QCIF format picture is subdivided into 9 $\times$ 11 macroblocks for a total of 99 macroblocks.

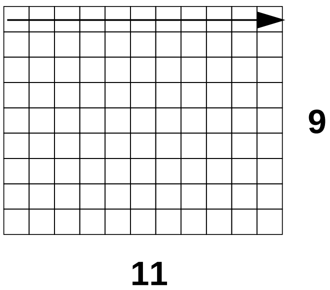


Figure 4: Sample QCIF image layout.

As noted above, compression is needed in order to enable efficient video transmission over data networks. Several types of redundancy can be exploited to achieve compression. The most commonly exploited redundancy is the temporal interdependence of consequtive video frames, which typically leads to the highest achievable compression gains. (Additional compression schemes are also in development, but not commonly applied up to now, such as the exploitation of object recognition techniques.) There are three methods for encoding the original pictures: I (Intra), P (Inter), and B (Bi–directional). These encoding methods are applied on the macroblock level. An intra–coded frame consists exclusively of intra-coded macroblocks. Thus, an intra–coded frame contains the compressed image information (without any prediction information), resulting in a large frame size (compared to the size of the inter– or bidirectional–coded frames). Intra–coding uses well-known compression schemes such as JPEG or wavelet–based approaches to compress the image information.

The inter–coded frames use a motion estimation relying on the previous inter– or intra–coded frame, whereas the bi–directional encoded frames rely on a previous as well as a following intra– oder inter–coded frame. This prediction information results in smaller frame sizes for the P–frames and even smaller frame sizes for the B–frames. The relationship between the encoding types and how frames rely on each other in a typical frame sequence [5] is illustrated in Figure 5. (Note that when B frames do not have any following I– or P–frames they can be referenced to, no encoding or decoding is possible, as illustrated in Figure 5.) The sequence of frames between

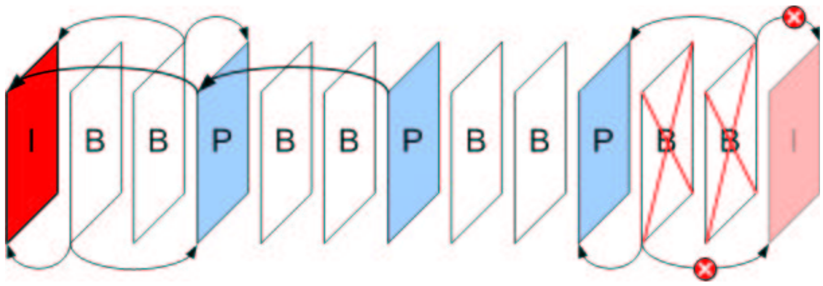


Figure 5: Typical frame sequence and dependencies for one GoP

the intra–coded frames is referred to as *Group of Pictures* (GoP). (Note that it is not necessary to have more than one I–frame at the beginning of the video sequence, in which case the entire frame sequence is a single GoP.)

To handle abrupt changes in the bitstream and the loss of parts of pictures or structures, the H.26L standard provides the possibility of refreshing the pictures on a macroblock level. Additionally, refresh frames (intra picture refresh) are used to stop the prediction process of frames that are referencing lost or errorneous frames. Furthermore, the standard will provide the possibility to switch between several different bitrate streams to avoid high computational effort (and thus high power consumption)for the encoding and decoding. In order to provide quantized values, an inverse discrete cosine transformation (IDCT) is utilized. The IDCT in H.26L is performed in the same manner as in H.263 [17]. The representation of the image in a finite numberspace done by the quantization based on the IDCT coefficients is the main reason for losses and compression. The quantization parameter defines the fidelity of the picture encoding, since the smaller the quantization parameter, the more values are available to express the value of each coefficient resulting from the transformation.

The H.26L standard takes the characteristics of wireless environments where the available bitrate may change often and over larger ranges into account. The value of the quantization parameter can be changed on a frame–by–frame basis as well as on a macroblock–by–macroblock basis. This — in addition to the stream switching functionality — allows for a fast response of the real–time encoding process to changing bandwidths. The stream switching functionality allows for non–realtime encoding and real–time, bandwidth–based selection of streams encoded with different quantization and/or GoP settings. In this paper we will not perform any evaluations of these advanced features. Instead, we focus on the non–real–time behavior of the standard and assume that the encoder has no information about the underlying channel charactersistics.

After quantizing the IDCT coefficients, the temporal redundancy of the picture information is removed by applying motion estimation. This is especially useful for high frame rates, where successive frames are highly correlated. The motion estimation is performed for multiple reference frames (see H.263++ standard, Annex U – *long term memory prediction*) and works beyond the picture boundaries as illustrated in Figure 6.

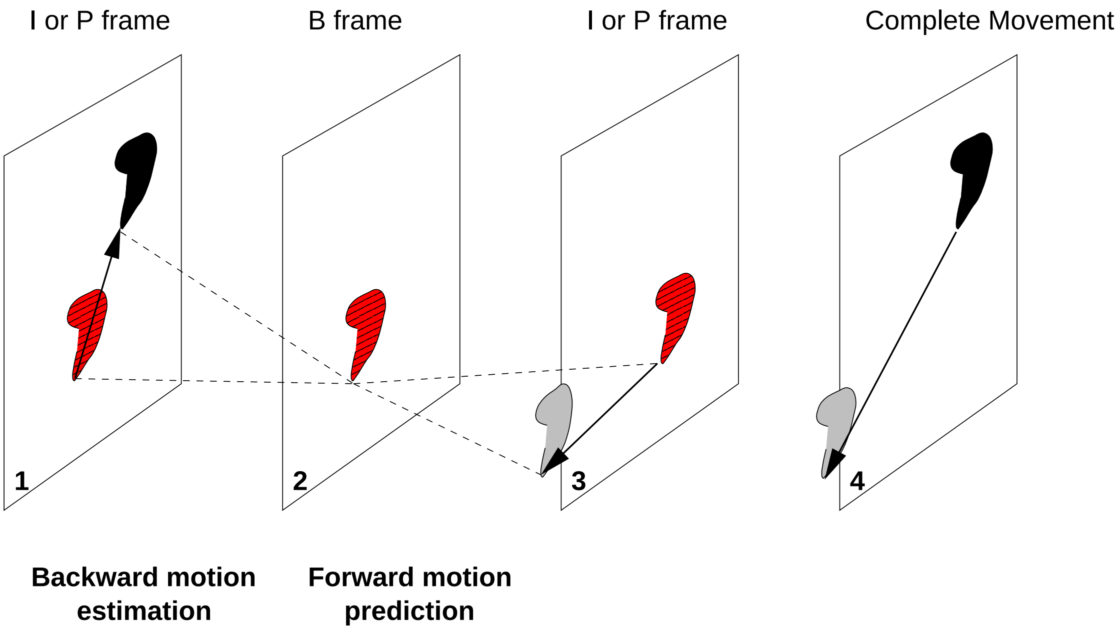


Figure 6: Illustration of motion estimation

Motion Compensation is utilizing so–called motion vectors to exploit the temporal redundancy more efficiently in order to gain higher compression. A motion vector uses reference frames (or fields of frames) in the past and/or future. This two–dimensional vector provides an offset from the coordinate in the current picture to the coordinates in the referenced frame. For illustration, the fourth frame in Figure 6 represents an already identified moving object with its full path. The first to third frames are illustrating the motion vector generation. For a backward prediction, an earlier reference is used to derive the coordinate change, whereas for forward prediction, a later reference is utilized. For the different encoding types, different prediction modes are implemented. The enhancement of normal motion vectors is the revocation of picture boundaries as limits for the validity of a vector's target, also known as unrestricted or extended motion vector mode. Frame three in Figure 6 gives such an example. Since there is no content and thus data available for the outside of a picture, the pixels at the border are simply replicated to fill the nonexistent values needed as references. Figure 7 illustrates this scheme.

Each macroblock can be subdivided into smaller fragments in order to provide a finer granu-

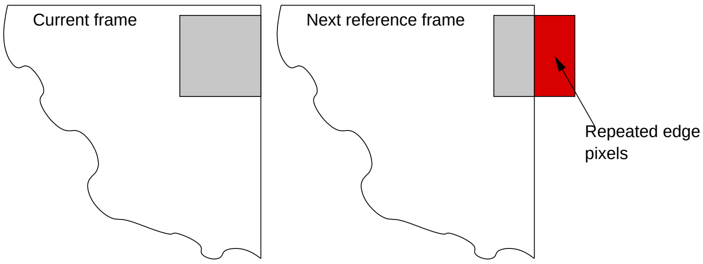


Figure 7: Illustration of unrestricted motion estimation.

larity and higher quality. The different subdivision formats are illustrated in Figure 8.

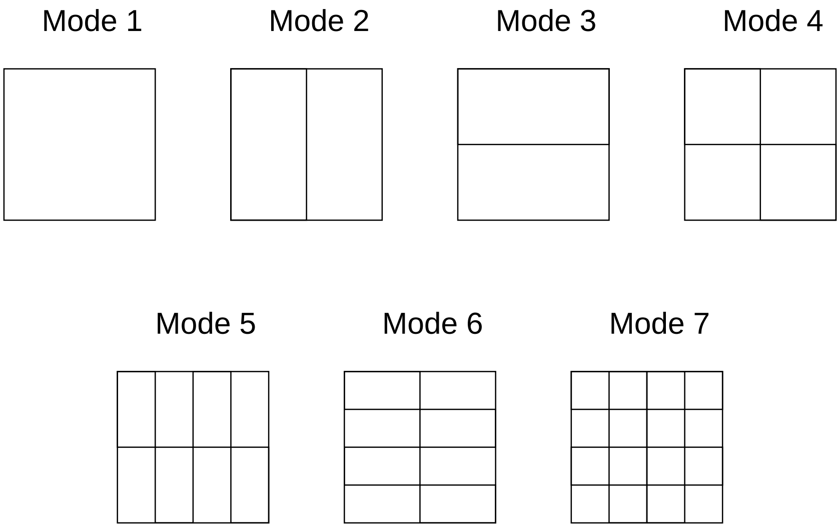


Figure 8: Different macroblock subdivision modes.

After the discrete cosine transformation and motion estimation are completed, some redundancy is typically still left in the video data. This remaining redundancy can be exploited by entropy–coding techniques such as the *universal variable length coding* (UVLC) or the *context–adaptive binary arithmetic coder* (CABAC) [17]. The latter approach uses probability distributions to further reduce the space needed to store the encoded frame. Shorter symbols are assigned to bit patterns with a high probability of occurence and longer symbols to bit patterns with a smaller probability of occurence. This mapping process achieves lossless compression. The UVLC uses an infinite set of code words and is applied only on the mapping of the symbols and thus reduces the necessity of redefining codewords [4]. The coding is based on a single, static table of codewords which results in a simple mapping process. As an alternative to the UVLC, the CABAC–technique can be used. This algorithm encodes the sequence of symbols into an interval of real numbers between 0 and 1 and is able to do this with respect to the symbol's probability at the source. It is therefore exploiting additional correlation of symbols at the encoding side for further reduction of data to be stored for each frame.

# 4 Measurement Setup

We used the reference JM2–encoder version 3.6 which is publicly available (for more recent releases refer to [13]). This reference encoder conforms to the current standard development and includes the currently proposed features. Nevertheless, as changes are ongoing, the software may lack the most recently adopted features. Since the purpose of our study is to generate and statistically evaluate the frame sizes of the encoded videostreams, we disabled some of the more advanced encoder features.

The disabled features included the slice mode that is providing error resilience features by coding fixed macroblocks or fixed bytes per slice. A slice is an individual entity no relying on other data inside a frame. We also used only the CABAC–technique to remove intersymbol correlation. The network adaption layer was also not used, as were restriction to the search range. We were therefore only using the basic features such as inter–, intra–, and bidirectional prediction and motion estimation. Additionally, we used a fixed GoP and motion–vector resolution setting for the prediction modes. The result is a setup being very close to the most basic encoding settings used in previous video trace file generation processes such as [5].

Overall, we believe that the differences between the encoder version used in our experiments and the final encoder version are negligible as far as the video traffic characterization is concerned for these basic settings.

We did not specify a target bit rate, since rate–adaptive encoding is not available at present. Instead, we used static quality levels (quantization parameters) which we set for all three frame types to 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 51. For ease of comparison with the already existing video trace files (H.261, H.263, and MPEG–4, see [5]) we used the GoP structure *IBBPBBPBBPBB*. Note that the encoder has to encode the referenced frames first, thus the resulting frame sequence is *IPBBPBBPBBIBBP....*

We initially used the freely available and widely used YUV testing sequences [16] in our experiments. These sequences are in the NTSC format and have a frame rate of 30 frames per second.

For each of the studied quality levels we encoded the YUV–files into the H.26L bitstream off–line (thus there was no frame drop during the encoding).

The encoder status output was parsed to generate the traces.

For each quantization level and test sequence we generated a *terse* and a *verbose* trace file, as illustrated in Figure 9. The traces were then used for the statistical analysis of the video traffic. (We found that working with the encoded file itself is yet not very advisable, see discussion on problems encountered) The verbose trace gives for each frame the type (I, P, or B), the playout time (= frame number/30) in msec, and the frame size in byte. The terse trace gives only the sequence of frame sizes in byte as generated by the encoder *IPBBPBBPBBIBBP.....*

We used bytes instead of bits, since every frame start code has to be byte–aligned in the H.26L bitstream. Note that in the encodings the last GoP is incomplete, since the last two B–frames are referencing a frame that is not available. The traces and the statistics are publicly available on our website [1] for viewing and downloading.

# 5 Statistical Results of Video Trace Files

The following results are only intended to give a first impression of the capabilities of the future H.26L standard. The longest testing sequence we were able to utilize has 1000 pictures. T he

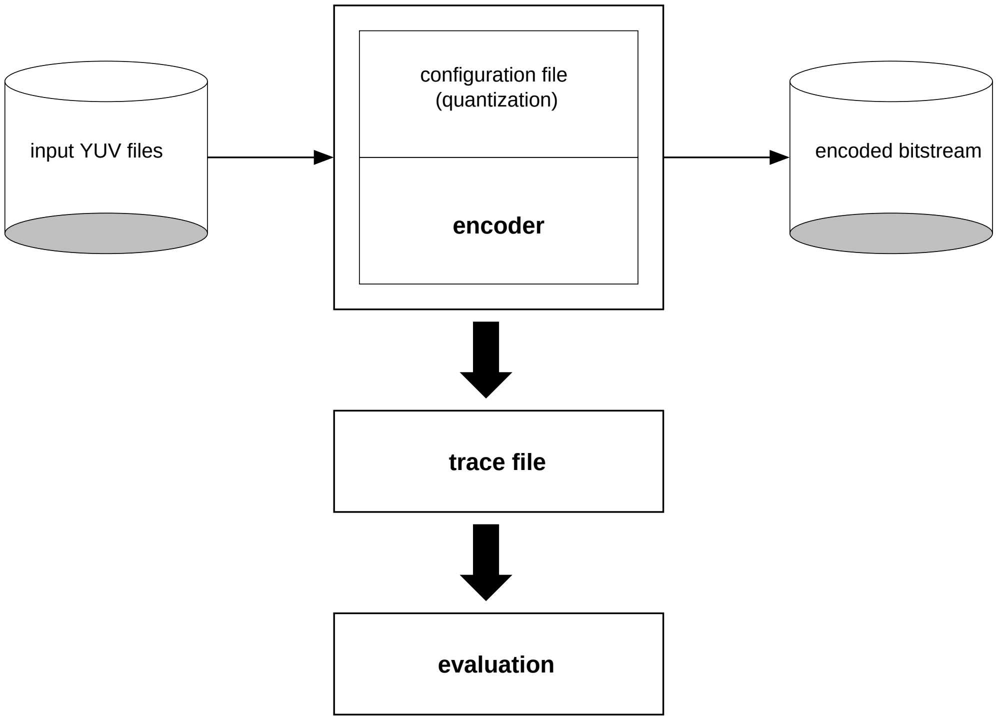


Figure 9: Setup of the evaluation process

following detailed discussion focuses on this video, called *Paris*, since the statistical evaluation of the shorter sequences is far more prone to residual errors and inconsistencies. A screenshot giving an impression of the content (discussion with some movements of bodies and ball) of the *Paris* seqeunce is given in Figure 10 We provide the results for the other video traces with a quantization parameter setting of 25 in abbreviated form in the corresponding appendeces. An overview of the evaluated sequences is given in Table 1. For the statistical evaluation of the traces we introduce the following notation.

Tables 2 and 3

Let $N$ denote the number of considered frames, in case of *Paris* this would be $N = 1000$. The individual frame sizes are denoted by $X_1, \ldots, X_N$. The mean frame size is estimated as

$$\overline{X} = \frac{1}{N} \cdot \sum_{i=1}^{N} X_i. \tag{5}$$

$$\tag{6}$$

The variance is estimated as

$$S_X^2 = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \left( X_i - \overline{X} \right)^2 \tag{7}$$

$$= \frac{1}{N-1} \left[ \sum_{i=1}^{N} X_i^2 - \frac{1}{N} \cdot \left( \sum_{i=1}^{N} X_i \right)^2 \right]. \tag{8}$$


Figure 10: Screenshot of *Paris* in CIF format

The Coefficient of Variation is given by

$$CoV = \frac{S_X}{\overline{X}}. \tag{9}$$

### 5.1 Frame–based Statistical Overview

Table 2 provides an overview of the basic statistics of the *Paris* traces for the different quantization parameter settings.

### 5.2 GoP–based Statistical Overview

We also evaluated the traces at an aggregation level of 12 frames, i.e., at the GoP level, see Table 3. This fixed–length moving average analysis gives a more stationary impression of the video trace since the frame type differences are smoothed out.

In the folowing sections, we give some graphical representations of the frame size traces, the distribution, the autocorrelation function, and the R/S plots. The R/S plots are used to find the Hurst parameters for the three quantization settings $ql = 1, 25, 51$ that are stated in each title as $QP_{ql}$. The main usage and conclusion of these plots will be described in the following part of the evaluation. These figures should give a graphical overview of some characteristical behavior (e.g. the long time dependency of the frame trace) which is regarded as a time series in statistical means.

Table 1: Overview of Evaluated Sequences

| Name of Video Sequence | Number of Frames | Format |
|---|---|---|
| Carphone | 382 | QCIF |
| Claire | 494 | QCIF |
| Container | 300 | QCIF |
| Foreman | 400 | QCIF |
| Grandma | 870 | QCIF |
| Mobile | 300 | CIF |
| Mother and Daughter | 961 | QCIF |
| News | 300 | QCIF |
| Paris | 1000 | CIF |
| Salesman | 449 | QCIF |
| Silent | 300 | QCIF |
| Tempete | 260 | CIF |

### 5.3 Frame Size Traces

The main purpose for evaluating the behavior of the frame sizes is to achieve a statistically sound base for the modeling and simulation of video traffic. The frame sizes reflect the video content and its dynamic behavior. With any block– and motionvector–based encoding process, the frame sizes are larger if the movie content is more dynamic and richer in texture. As can be seen in the frame traces of *Carphone*, the frame size is rising around frame 150. This is due to a shift of the landscape in the back, viewable through the car window. Before, the view is a clear sky, only occasionally interrupted by moving objects (e.g. lanterns, street signs) — after frame 150, the view is a forest, with a rich texture. Figure 11 gives an impression on the changing backgrounds and the resulting frame sizes for a GoP–aggregation.

Furthermore, the frame sizes are larger when smaller quantization parameters are used (which in turn give higher video quality). These factors are interdependent, i.e., high dynamics paired with finer quantization results in larger frame sizes, and vice versa. We observe from Figures 12, 13, and 14, that the range of frame sizes is extremely different within a given GoP.

The GoP–smoothed traces in Figures 15, 16, and 17 give a clearer impression of the traffic dynamics. We observe that the plots do not indicate any large dynamic change. This is because the used tesing sequences typically have only little dynamic change in their content. In fact, these testing sequences are typically employed to study video encoding at the time scale of a video frame or smaller. The study of the impact of dynamic changes of the video content on the video traffic requires longer test videos, which we will study in future work. A clear observation from the figures is that frame sizes are larger for smaller quantization parameters

### 5.4 Frame Size Distribution

The distribution of the frame sizes is needed in order to make any statistical modeling of the traffic possible. Frame size histograms or probability distributions allow us to make observations concerning the variability of the encoded data and the necessary requirements for the purpose of real–time transport of the data over a combination of wired and wireless networks. In the following we present the probability density function $p$ as a function of the frame size. For the

Table 2: Single frame statistics for different quality levels using *Paris*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 43390 | 29037 | 12061 | 3930 | 1288 | 418 | 119 | 67 | 29 | 25 | 16 |
| $X_{max}$ [byte] | 95525 | 81139 | 62578 | 46474 | 33824 | 23919 | 15746 | 9840 | 5763 | 3214 | 1448 |
| $\overline{X}$ [byte] | 54345.28 | 39572.05 | 22062.51 | 11395.94 | 6331.72 | 3827.69 | 2145.86 | 1182.10 | 647.40 | 348.34 | 161.30 |
| $\overline{X}_{I-frame}$ [byte] | 94414.24 | 80066.83 | 61447.85 | 45408.61 | 32699.18 | 22964.62 | 14945.32 | 9248.70 | 5445.10 | 3035.90 | 1377.35 |
| $\overline{X}_{P-frame}$ [byte] | 58793.27 | 43068.29 | 24494.70 | 11899.60 | 5928.88 | 3337.22 | 1748.51 | 854.76 | 421.38 | 208.58 | 64.75 |
| $\overline{X}_{B-frame}$ [byte] | 47621.87 | 33152.20 | 16182.01 | 6916.99 | 3157.31 | 1598.14 | 680.67 | 287.57 | 127.13 | 61.83 | 44.16 |
| $S_X^2$ | 174399635.34 | 172302917.00 | 158433906.59 | 112674047.03 | 65962417.27 | 34444608.41 | 15328149.06 | 6054760.37 | 2134816.65 | 668818.27 | 136146.83 |
| $CoV$ | 0.24 | 0.33 | 0.57 | 0.93 | 1.28 | 1.53 | 1.82 | 2.08 | 2.26 | 2.35 | 2.29 |
| **Mean bitrate** [bit/s] | 13042866.96 | 9497292.48 | 5295003.36 | 2735025.36 | 1519611.84 | 918646.32 | 515007.12 | 283704.00 | 155376.24 | 83601.84 | 38711.04 |
| **Peak bitrate** [bit/s] | 25473333.33 | 21637066.67 | 16687466.67 | 12393066.67 | 9019733.33 | 6378400.00 | 4198933.33 | 2624000.00 | 1536800.00 | 857066.67 | 386133.33 |
| **Peak to mean** | 1.95 | 2.28 | 3.15 | 4.53 | 5.94 | 6.94 | 8.15 | 9.25 | 9.89 | 10.25 | 9.97 |

Table 3: GoP statistics for different quality levels using *Paris*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 93578 | 79234 | 60610 | 44535 | 31944 | 22312 | 14461 | 8896 | 5273 | 2955 | 1348 |
| $X_{max,GoP}$ [byte] | 721737 | 539976 | 324552 | 179212 | 101868 | 61343 | 33712 | 17741 | 9496 | 5035 | 2362 |
| $\overline{X}_{GoP}$ [byte] | 650782.23 | 473844.23 | 264149.30 | 136385.82 | 75734.05 | 45763.58 | 25640.10 | 14112.82 | 7725.02 | 4154.18 | 1924.11 |
| $S^2_{X,GoP}$ | 512264567.35 | 432694471.18 | 366798439.65 | 199274339.93 | 74414914.80 | 26030350.12 | 6610073.94 | 1420858.69 | 319171.88 | 60188.64 | 10792.15 |
| $CoV_{GoP}$ | 0.03 | 0.04 | 0.07 | 0.10 | 0.11 | 0.11 | 0.10 | 0.08 | 0.07 | 0.06 | 0.05 |
| **Mean GoP rate** [bit/s] | 14461.83 | 10529.87 | 5869.98 | 3030.80 | 1682.98 | 1016.97 | 569.78 | 313.62 | 171.67 | 92.32 | 42.76 |
| **Peak GoP rate** [bit/s] | 16038.60 | 11999.47 | 7212.27 | 3982.49 | 2263.73 | 1363.18 | 749.16 | 394.24 | 211.02 | 111.89 | 52.49 |
| **Peak to mean** | 1.11 | 1.14 | 1.23 | 1.31 | 1.35 | 1.34 | 1.31 | 1.26 | 1.23 | 1.21 | 1.23 |

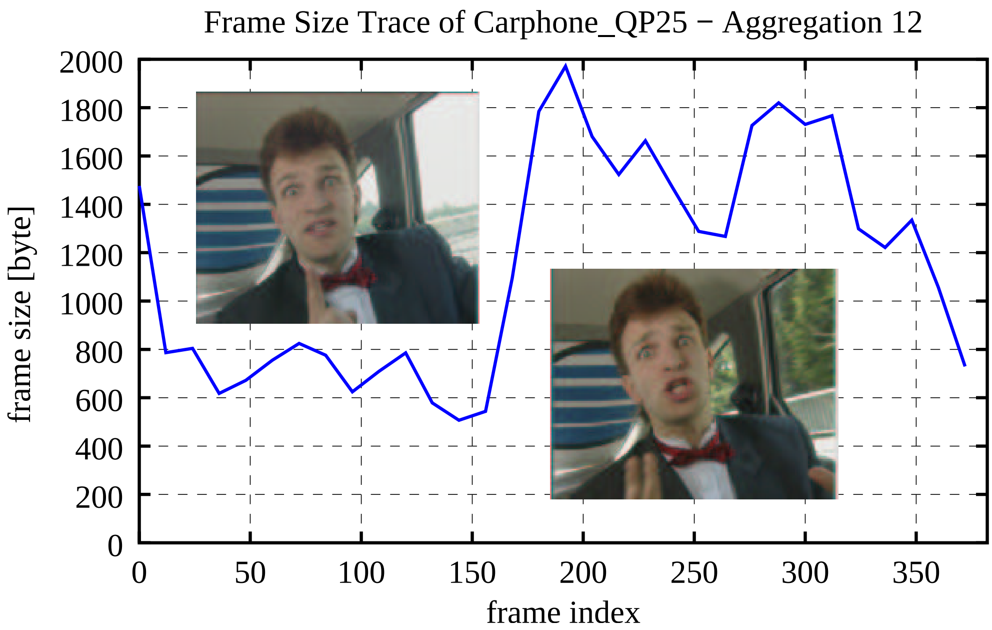


Figure 11: Impact of changing background dynamics on frame sizes.

probability distribution function as well as the inverse probalbility distribution function, we refer to our web page [1]. We observe for all the different quality levels a large spread of the frame sizes. We observe that the distribution is spreading out more for smaller quantization parameters.

This is expectedly derived by comparing the differences in the frame sizes for the different frame types (which normally tend to be high for I–frames, intermediate for P–frames, and low for B–frames). With lower fidelity (i.e. higher quantization), the differentiation between these types regarding the frame size is decreasing due to the more forcefully applied quantization. The viewable result is characterized by a total loss of clear differences between objects, colors and so forth. Figures 18, 19, and 20 give an overview of these quantization effects (please note that these images were scaled down to fit on a single page).

The overall distribution may very roughly be seen as normal or Gaussian, what should be easing future modeling efforts. A short warning here, again, with respect to the length of the traces evaluated up to now.

### 5.5 Autocorrelation Coefficient

The autocorrelation [3] function can be used for the detection of non–randomness in data or identification of an appropriate time series model if the data are not random. One basic assumption is that the observations are equi-spaced. The autocorrelation is a correlation coefficient and thus referred to as autocorrelation coefficient (acc). However, instead of the correlation between two different variables, the correlation is between two values of the same variable at times $X_t$ and $X_{t+k}$. When the autocorrelation is used to detect non-randomness, it is usually only the first (lag $k = 1$) autocorrelation that is of interest. When the autocorrelation is used to identify an appropriate time series model, the autocorrelations are usually plotted for a range of lags $k$.

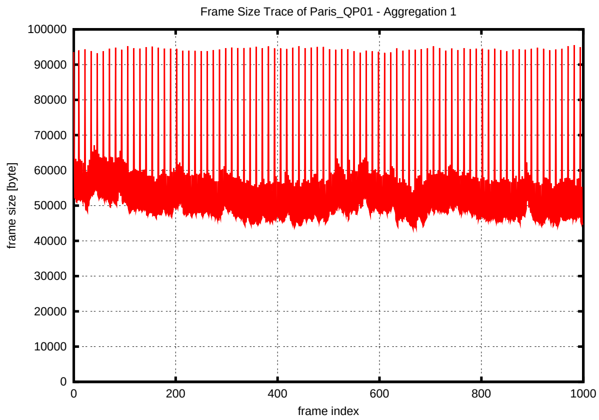


Figure 12: Frame size trace of *Paris (quantization 01)*

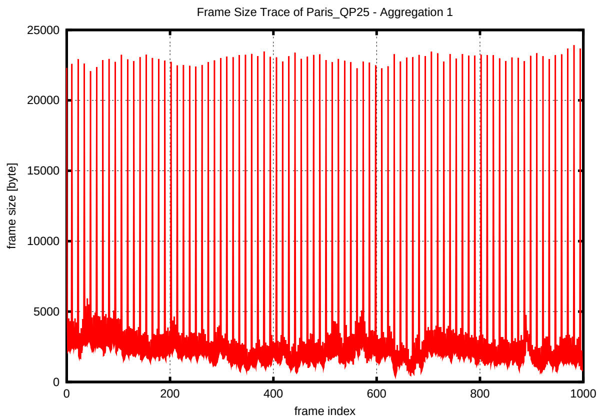


Figure 13: Frame size trace of *Paris (quantization 25)*

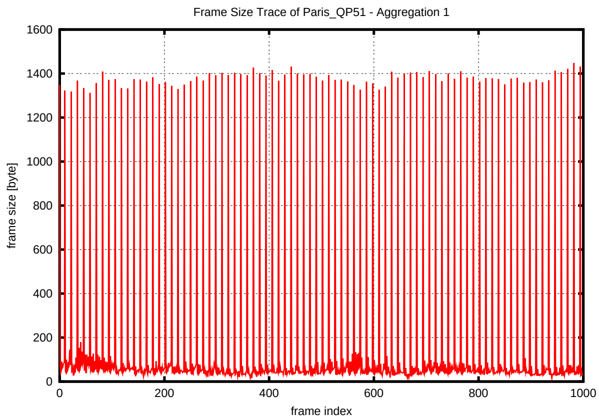


Figure 14: Frame size trace of *Paris (quantization 51)*

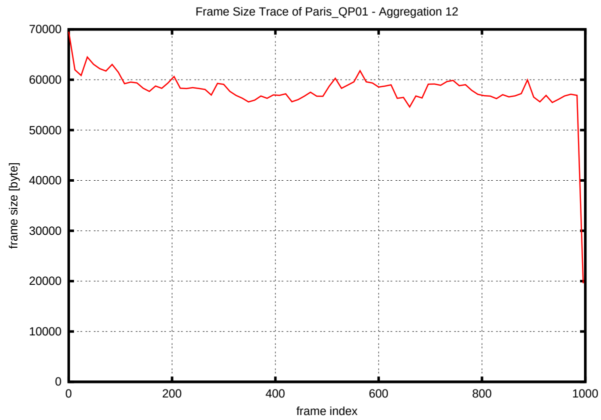


Figure 15: Frame size trace of *Paris* averaged over one GoP *(quantization 01)*

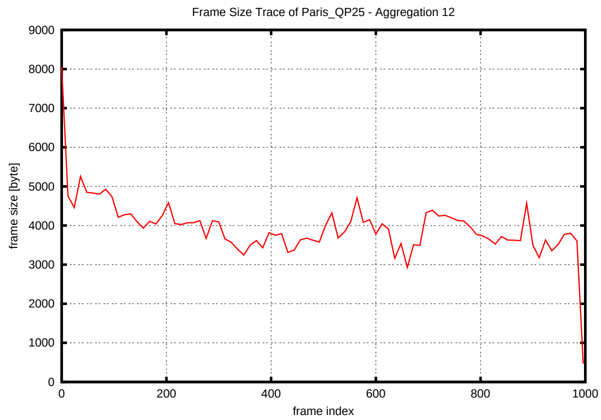


Figure 16: Frame size trace of *Paris* averaged over one GoP *(quantization 25)*

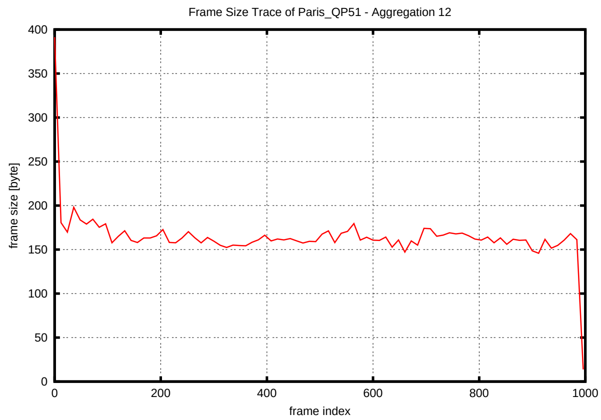


Figure 17: Frame size trace of *Paris* averaged over one GoP *(quantization 51)*

Figure 18: Quantization effect for *Paris* (quantization 40, PSNR for this frame: 27.4271)

Figure 19: Quantization effect for *Paris* (quantization 45, PSNR for this frame: 24.2853)

Figure 20: Quantization effect for *Paris* (quantization 51, PSNR for this frame: 20.3898)

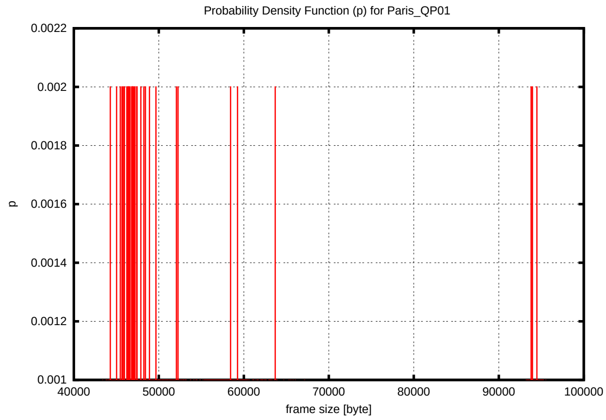


Figure 21: Frame size distribution for *Paris (quantization 01)*

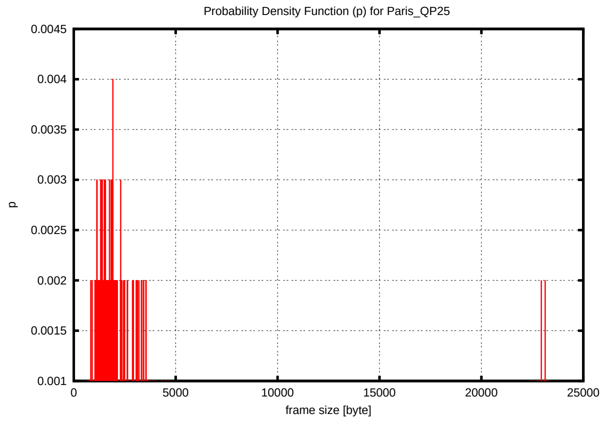


Figure 22: Frame size distribution for *Paris (quantization 25)*

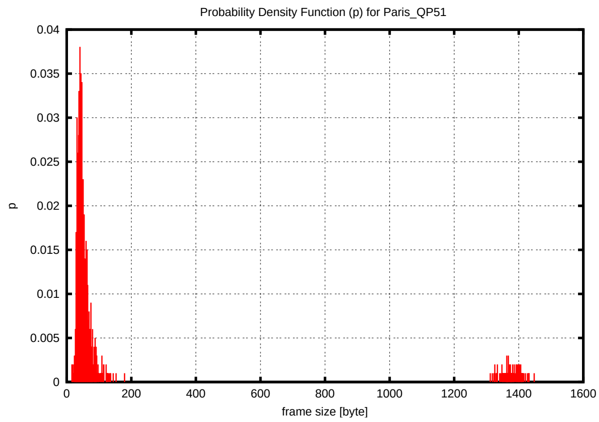


Figure 23: Frame size distribution for *Paris (quantization 51)*

With our notation the acc can be estimated by

$$\rho_X(k) = \frac{1}{N-k} \cdot \sum_{i=1}^{N-k} \frac{\left(X_i - \overline{X}\right) \cdot \left(X_{i+k} - \overline{X}\right)}{S_X^2} \tag{10}$$

In 10 the lag (i.e. for either single frames or aggregated for one or multiple GoPs) is denoted as $k$, with $k = 0, 1, \ldots, N$. The autocorrelation function for the single frame aggregation level shows the similarity within a GoP, whereas higher aggregation levels give an indication of the long–term self–similarity. We observe from Figures 24, 25, and 26, that there large spikes spaced 12 frames apart. These are due to repetive GoPs, which contain 12 frames each. Thus for a lag of 12 frames, I frames correlate with I frames, P frames with P frames, and B frames with B frames. The intermediate spikes that are spaced three frames apart are due to the correlations between I and P frames. We observe that the intermediate spikes are decreasing with the fidelity of the encoded bitstream. This appears to be due to the wider spread of the frame size distribution for larger quantization parameters.

We observe from Figures 27, 28, and 29 that the GoP–based autocorrelation tends to fall off slower than an exponential, suggesting the presence of long-range dependencies.

## 5.6 R/S Plots

The Hurst parameter, or self–similarity parameter, $H$, is a key measure of self-similarity [7, 8]. $H$ is a measure of the persistence of a statistical phenomenon and is a measure of the length of the long range dependence of a stochastic process. A Hurst parameter of $H = 0.5$ indicates absence of self-similarity whereas $H = 1$ indicates the degree of persistence or a present long–range dependence. The $H$ parameter can be estimated from a graphical interpolation of the so–called R/S plot. The R/S plot gives the graphical interpretation of the rescaled adjusted range statistic by utilizing the following method [2, 9].

The length of the complete series $N$ has to be subdivided into blocks with a length of $k$, for which the partial sums $Y(k)$ have to be calculated as in Equation 11. Following the variance of all these aggregations has to be calculated. The resulting R/S value is derived as shown in Equation 13 for a single block.

$$Y(k) = \sum_{i=1}^{k} X_i \tag{11}$$

$$S_X^2(k) = \frac{1}{k} \cdot \sum_{i=1}^{k} \left[ X_i^2 - \left(\frac{1}{k}\right)^2 \cdot Y(k)^2 \right] \tag{12}$$

$$\frac{R}{S}(N) = \frac{1}{S_X(k)} \left[ max_{0 \le t \le k} \left( Y(t) - \frac{t}{k} \cdot Y(k) \right) - min_{0 \le t \le k} \left( Y(t) - \frac{t}{k} \cdot Y(k) \right) \right] \tag{13}$$

If plotted on an log/log scale for R/S versus differently sized blocks, the result will be several different points. This plot is also called the *pox plot* for the R/S statistic. The Hurst parameter $H$ can then be estimated by fitting a line to the points of the plot, normally by using a least square fit, neglecting the residual values at the lower and upper borders ( since those are typically transient zones that represent the short rage dependencies, which exist on a GoP–level as studied earlier). The larger the resulting Hurst parameter, the higher the degree of long range dependency of the time series. As can be seen from the following Figures 30, 31, 31 on a single–frame basis,

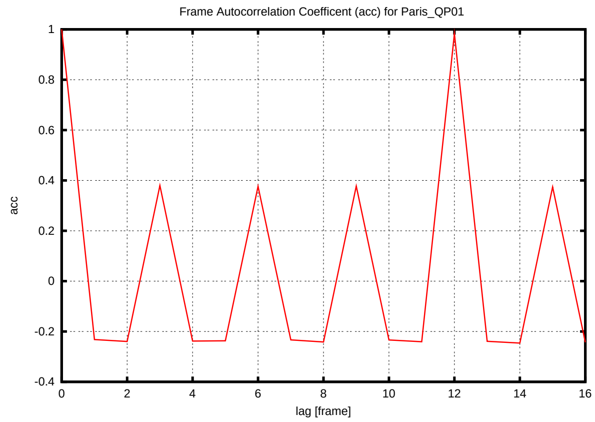


Figure 24: Autocorrelation coefficients for *Paris (quantization 01)*

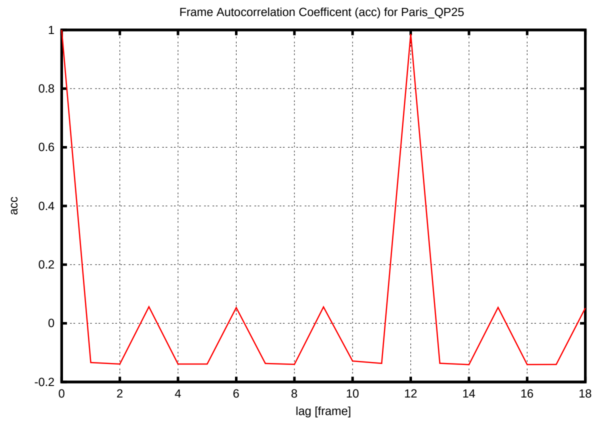


Figure 25: Autocorrelation coefficients for *Paris (quantization 25)*

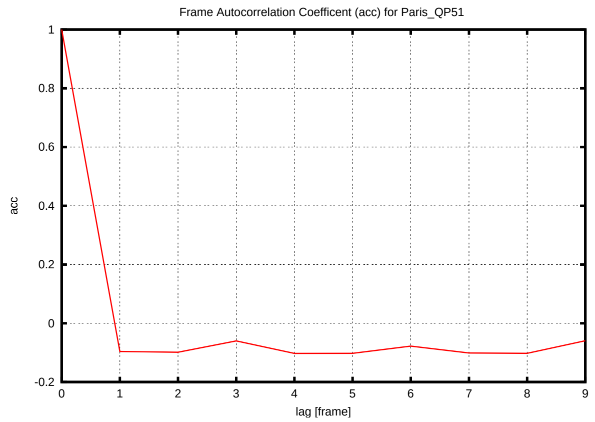


Figure 26: Autocorrelation coefficients for *Paris (quantization 51)*

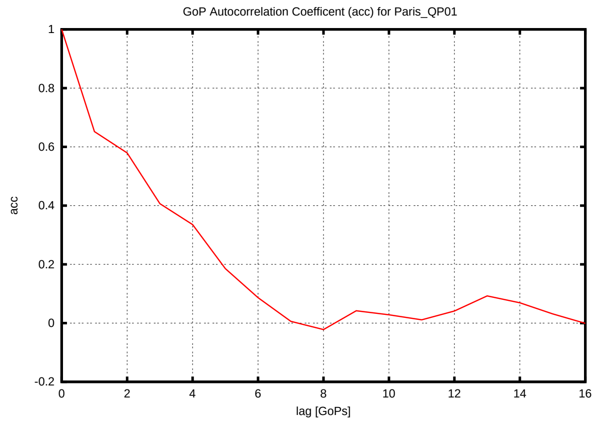


Figure 27: GoP autocorrelation coefficients for *Paris (quantization 01)*

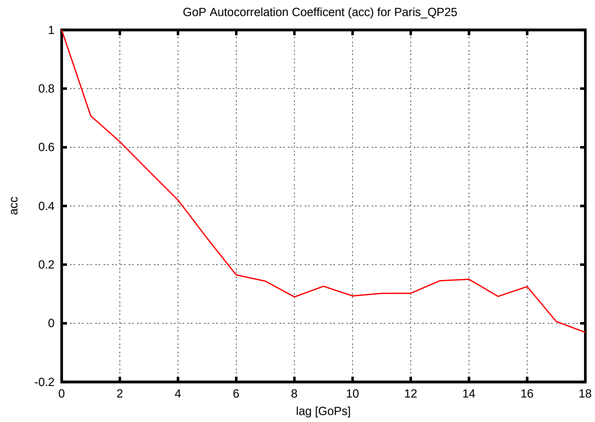


Figure 28: GoP autocorrelation coefficients for *Paris (quantization 25)*

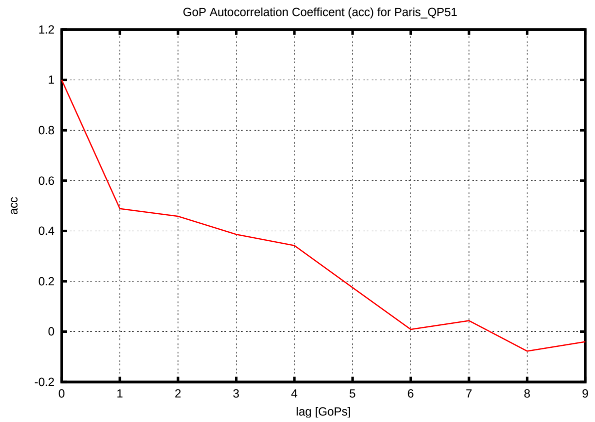


Figure 29: GoP autocorrelation coefficients for *Paris (quantization 51)*

and 33, 34, 35 on a GoP–basis, the hurst parameters stay well above 0.5, reflecting the presence of long–term dependence.

We applied the $4\sigma$–test [11] to eliminate all outlying residuals for a better estimation of the hurst parameter.

### 5.7 Variance Time Plot

The variance time plot is applied to a time series to show the development of the variance as in Equation 8 over different aggregation levels. This provides another test for long–range dependency [10, 5, 8, 2]. It is furthermore used to derive an estimation of the Hurst parameter. In order to obtain the plot, the normalized variance as given in Equation 14 of the trace is plotted as a function of different aggregation levels $k$ of the single frame sizes in a log–log plot. For each agggregation level $k$ the total amount of frames $N$ is divided into blocks and the variance calculated as shown before in Equations 11 and 12.

$$S_{norm} = \frac{S_X^2(k)}{S_X^2} \tag{14}$$

If no long range dependency is present, the slope of the function would be $-1$. For slopes larger than $-1$, a dependency is present. For simple reference we plot a reference line with a slope of $-1$ in the figures. We did not apply any regression–fits up to now but plan to do so in the future.

Our plots in Figures 36, 37, and 38 indicate a certain degree of long term dependency since the estimated slope is less than $-1$. We estimate that this is due to the occurence of the I–frames every 12 frames.

### 5.8 Periodogram Plot

A periodogram is a graphical data analysis technique for examining frequency–domain models of an equi–spaced time series. The periodogram is the Fourier transform of the autocovariance function. This calculation is currently employed in measuring the spectral density, following the idea that this spectrum is actually the variance at a given frequency. Therefore additional information about the magnitude of the variance of a given time series can be obtained by identifying the frequency component. This is done by correlating the series against the sine/cosine functions, leading to the Fourier frequencies [12]. For the calculation of the periodogram plot, the frame sizes $x_i$ of $N$ frames are aggregated into equidistant blocks $k$. For each block, the moving averages and their according logarithms are calculated as in Equations 15 and 16 with $n = 1, \ldots, N/k$.

$$Y_n^{(k)} = \frac{1}{k} \cdot \sum_{i=1}^{\frac{N}{k}} x_i \tag{15}$$

$$Z_n^{(k)} = log_{10} Y_n^{(k)} \tag{16}$$

In order to determine the frequency part of the periodogram, we calculate $\lambda_k$ as in Equation 17. The periodogram itself is then derived as given in Equation 18. For each different aggregation level, we plot the resulting $I(\lambda_k)$ and $\lambda_k$ in a *log/log*–plot.

$$\lambda_k = \frac{2\pi i}{\frac{N}{k}},\ i = 1, \ldots, \frac{M-1}{2} \tag{17}$$

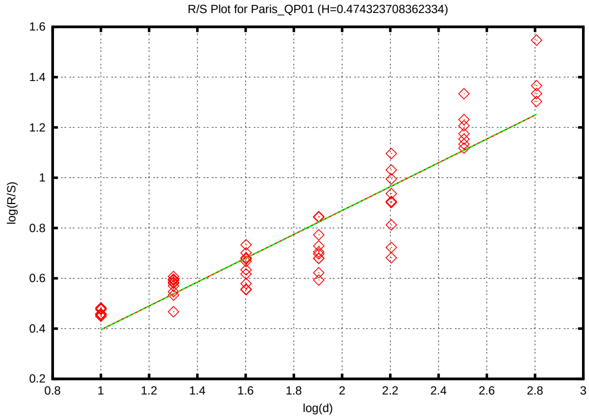


Figure 30: Single Frame R/S plot and $H$ parameter for *Paris (quantization 01)*

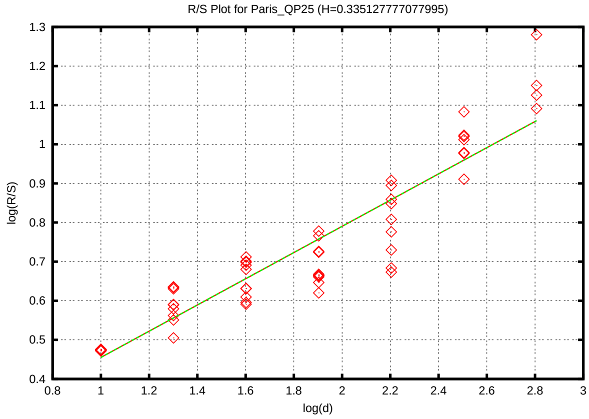


Figure 31: Single Frame R/S plot and $H$ parameter for *Paris (quantization 25)*

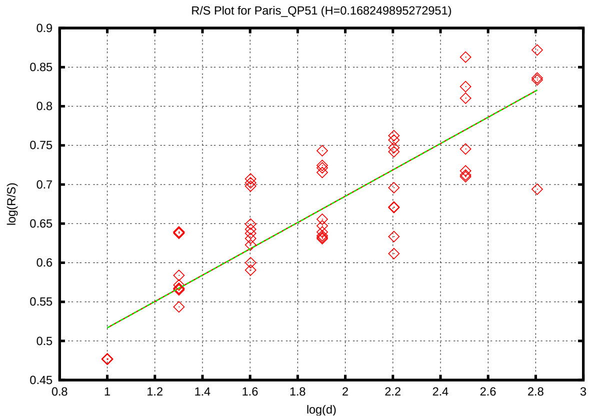


Figure 32: Single Frame R/S plot and $H$ parameter for *Paris (quantization 51)*

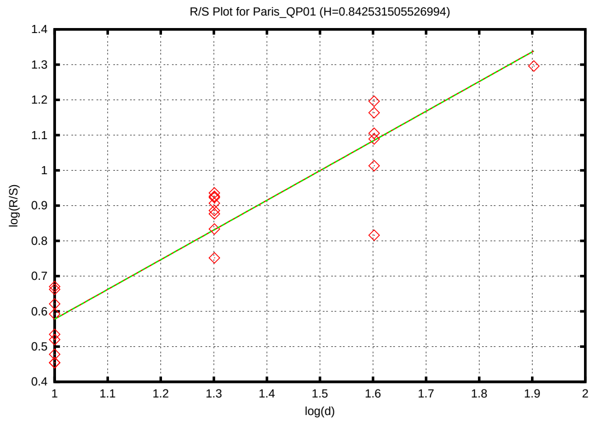


Figure 33: GoP R/S plot and $H$ parameter for *Paris (quantization 01)*

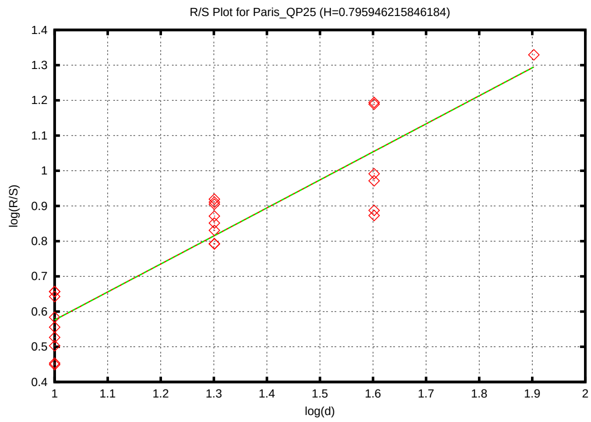


Figure 34: GoP R/S plot and $H$ parameter for *Paris (quantization 25)*

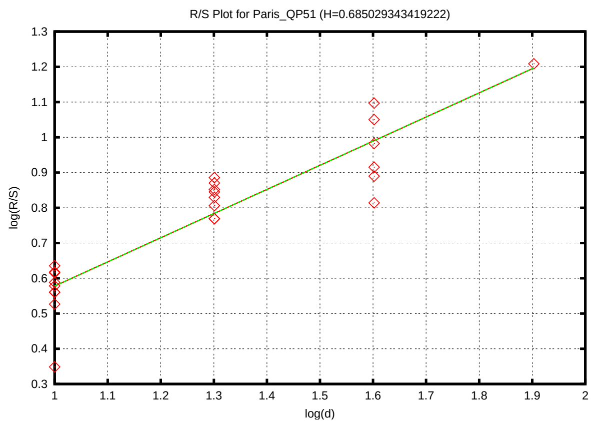


Figure 35: GoP R/S plot and $H$ parameter for *Paris (quantization 51)*

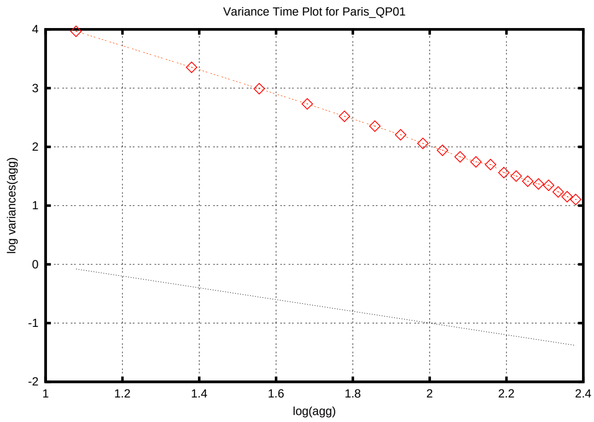


Figure 36: Variance time plot for *Paris (quantization 01)*

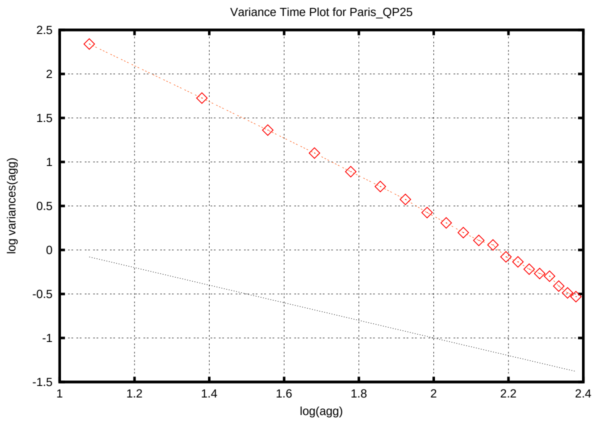


Figure 37: Variance time plot for *Paris (quantization 25)*

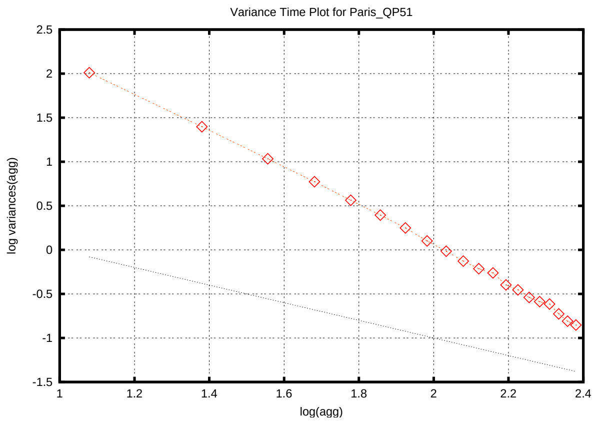


Figure 38: Variance time plot for *Paris (quantization 51)*

$$I(\lambda_k) = \frac{1}{2\pi\frac{N}{k}} \cdot \left| \sum_{l=0}^{\frac{N}{k}-1} Z_l^{(a)} \cdot e^{-jl\lambda_k} \right|^2 \tag{18}$$

The resulting plots are shown in Figures 39, 40, and 41 for a single frame aggregation and Figures 42, 43, and 44 for an aggregation level of a single GoP.

The Hurst paramter is estimated as $H = (1 - \beta_1)/2$, using a least squares regression on the samples. The Equations 20 and 21 are applied to determine the slope of the fitted line $y = \beta_0 + \beta_1 x$ and $H$.

$$K = 0.7 \cdot \frac{\frac{N}{k} - 2}{2} \tag{19}$$

$$\beta_1 = \frac{K \cdot \sum_{i=1}^{K} x_i y_i - \left(\sum_{i=1}^{K} x_i\right) \cdot \left(\sum_{i=1}^{K} y_i\right)}{K \cdot \left(\sum_{i=1}^{K} x_i^2\right) - \left(\sum_{i=1}^{K} y_i\right)^2}, \tag{20}$$

$$\beta_0 = \frac{\sum_{i=1}^{K} y_i - \beta_1 \cdot \sum_{i=1}^{K} x_i}{K} \tag{21}$$

## 6 Conclusion

In this report we have reported on our pre-standard evaluation of the H.26L video compression standard. Although the standard is not finalized as of the writing of this report and some changes in the algorithms employed as well as the output format generated are possible, it is expected that the basic routines utilized in our study will not change. As a consequence, our traffic characterizations give very close approximations of the final H.26L standard.

In the ongoing research of H.26L video compression we want to make comparisons with the currently utilized standard video encoding formats according to compression and statistical behavior. Once the final H.26L standard has been approved by the ITU–T, we will encode full–length movies to make more appropriate and consistent evaluations. The currently evaluated sequences do lack length and differ in content from what is expected to be a *typical* movie that would be transmitted over wireless networks in the future. Therefore the presented results will likely differ from what could be expected by a more lengthy evaluation. Since dynamic behavior and content are correlated, we will examine different categories such as action, comedy, animated, and news. Additionally, we want to expand the statistical evaluation further with additional long term dependency evaluations and outlier elimination.

## References

[1] acticom GmbH. Publicly available research results. http://www.acticom.info. 10, 16

[2] J. Beran. *Statistics for Long–Memory Processes*. Chapman and Hall, London, 1994. 21, 24

[3] Box, G. E. P., and Jenkins, G. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976. 16

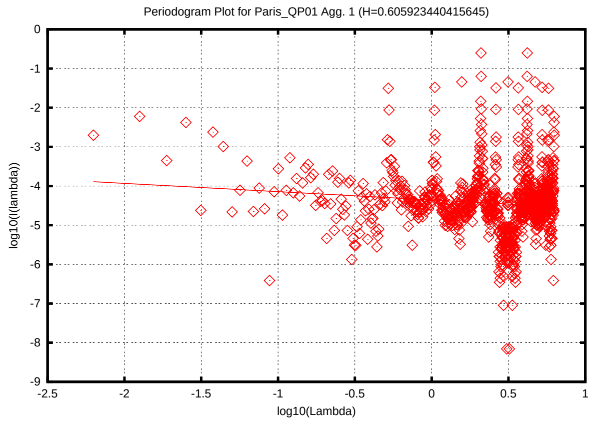


Figure 39: Single frame periodogram plot for *Paris (quantization 01)*

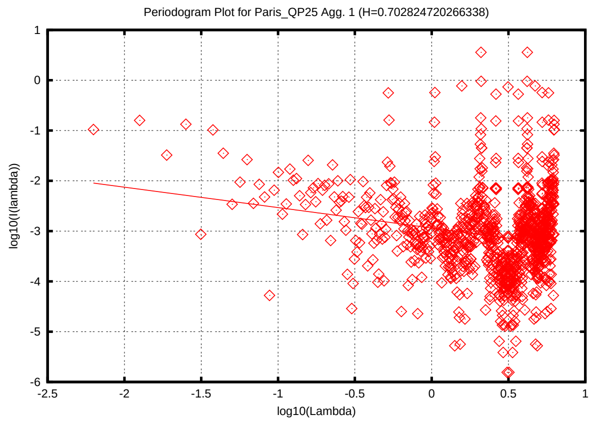


Figure 40: Single frame periodogram plot for *Paris (quantization 25)*

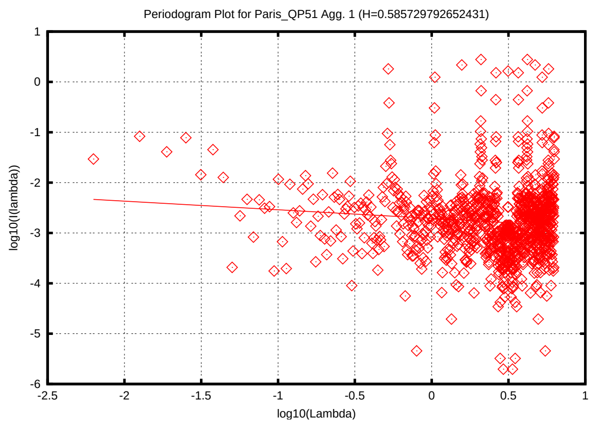


Figure 41: Single frame periodogram plot for *Paris (quantization 51)*

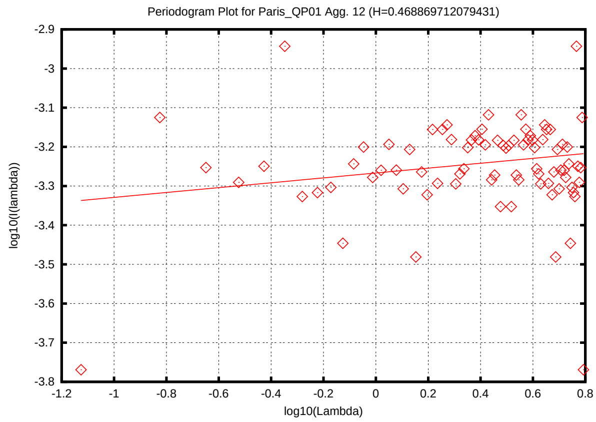


Figure 42: Single GoP periodogram plot for *Paris (quantization 01)*

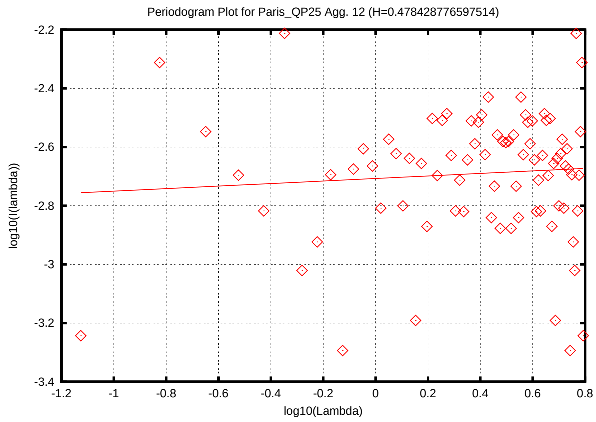


Figure 43: Single GoP periodogram plot for *Paris (quantization 25)*

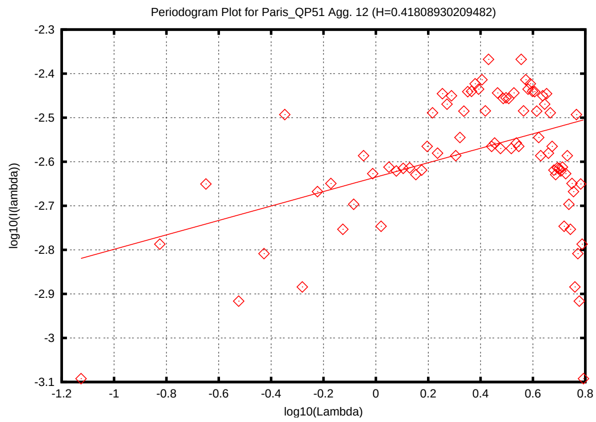


Figure 44: Single GoP periodogram plot for *Paris (quantization 51)*

[4] Kristofer Dovstam. *Video Coding in H.26L Video Coding in H.26L*. PhD thesis, 2000. 9

[5] Frank H.P. Fitzek and Martin Reisslein. MPEG–4 and H.263 Video Traces for Network Performance Evaluation. Technical report, Technical University of Berlin, 2000. TKN–00–06. 7, 10, 24

[6] Bernd Girod and Niko Färber. *Compressed Video Over Networks*, chapter Wireless Video. November 1999. 6

[7] H.E. Hurst. Long–Term Storage Capacity of Reservoirs. *Proc. American Society of Civil Engineering*, 76(11), 1950. 21

[8] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993. 21, 24

[9] A.W. Lo. Long–term Memory in Stock Market Prices. *Economatria*, (59):1276–1313, 1991. 21

[10] P. Morin. The impact of self-similarity on network performance analysis, 1995. 24

[11] Sachs, Lothar. *Angewandte Statistik*. Springer–Verlag, 2002. 24

[12] Stoffer David S. Shumway, Robert H. *Time Series Analysis and Its Applications*. Springer, New York, 2000. 24

[13] Suehring, Carsten. H.26l software coordination. http://bs.hhi.de/ suehring/tml/. 10

[14] Wiegand Thomas Sullivan, Garry J. and Thomas Stockhammer. Using the Draft H.26L Video Coding Standard for Mobile applications. 6

[15] D. S. Turaga and T. Chen. *Fundamentals of Video Coding: H.263 as an example Compressed Video over Networks*, chapter Fundamentals of Video Coding: H.263 as an example. The Signal Processing Series. MarcelDekker, 2000. 6

[16] Wenger, Stephan. The tml project web-page and archive. http://kbs.cs.tu-berlin.de/ stewe/vceg/. 10

[17] Th. Wiegand. H.26L Test Model Long–Term Number 9 (TML-9) draft0. ITU-T Study Group 16, Dec. 2001. 7, 9

# Statistical Results for Other Reference Videos

## A Carphone

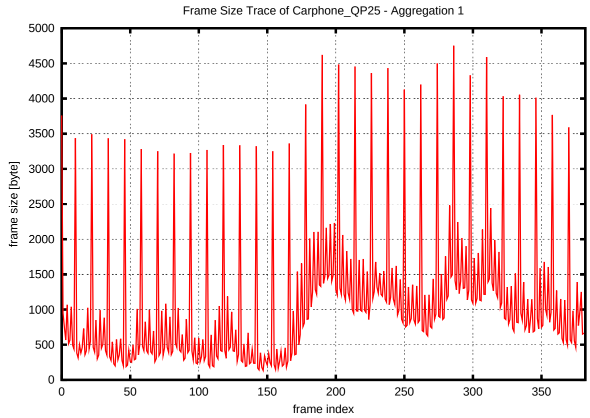


Figure 45: Frame size trace for *Carphone (quantization 25)*

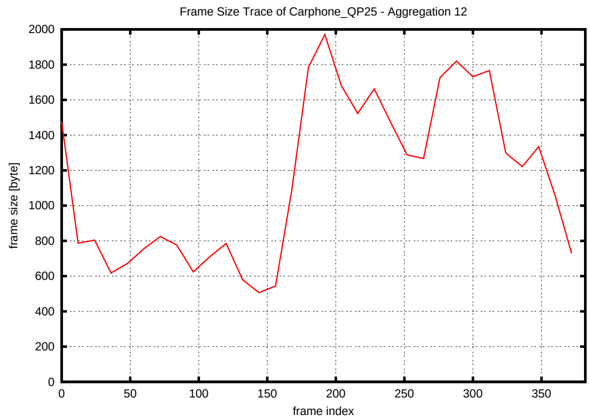


Figure 46: Frame size trace for one GoP *Carphone (quantization 25)*

Table 4: Single frame statistics for *Carphone*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 11263 | 7497 | 2986 | 1026 | 318 | 134 | 43 | 20 | 18 | 13 | 13 |
| $X_{max}$ [byte] | 22367 | 18807 | 14005 | 10059 | 6986 | 4751 | 2977 | 1730 | 991 | 541 | 243 |
| $\overline{X}$ [byte] | 15030.94 | 11131.97 | 6398.54 | 3525.17 | 1949.04 | 1094.14 | 563.97 | 288.25 | 148.55 | 79.89 | 40.81 |
| $\overline{X}_{I-frame}$ [byte] | 20617.12 | 17027.69 | 12166.59 | 8438.16 | 5760.34 | 3831.91 | 2350.25 | 1392.09 | 807.25 | 460.12 | 215.44 |
| $\overline{X}_{P-frame}$ [byte] | 15991.07 | 11868.86 | 6993.22 | 3934.02 | 2211.04 | 1243.75 | 630.73 | 315.68 | 158.80 | 81.05 | 30.76 |
| $\overline{X}_{B-frame}$ [byte] | 13964.29 | 10110.69 | 5447.09 | 2751.69 | 1369.86 | 692.68 | 313.69 | 138.81 | 61.69 | 31.55 | 22.61 |
| $S_X^2$ | 5764401.44 | 5835062.73 | 5447232.48 | 3609468.01 | 1967678.99 | 940211.71 | 370694.35 | 131964.33 | 44803.54 | 14335.54 | 2890.45 |
| $CoV$ | 0.16 | 0.22 | 0.36 | 0.54 | 0.72 | 0.89 | 1.08 | 1.26 | 1.42 | 1.50 | 1.32 |
| **Mean bitrate** [bit/s] | 3607426.18 | 2671672.46 | 1535648.80 | 846040.84 | 467770.68 | 262593.93 | 135352.46 | 69179.69 | 35651.31 | 19174.24 | 9794.76 |
| **Peak bitrate** [bit/s] | 5964533.33 | 5015200.00 | 3734666.67 | 2682400.00 | 1862933.33 | 1266933.33 | 793866.67 | 461333.33 | 264266.67 | 144266.67 | 64800.00 |
| **Peak to mean** | 1.65 | 1.88 | 2.43 | 3.17 | 3.98 | 4.82 | 5.87 | 6.67 | 7.41 | 7.52 | 6.62 |

Table 5: GoP statistics for *Carphone*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 20435 | 16906 | 12000 | 8264 | 5363 | 3464 | 2054 | 1243 | 724 | 428 | 209 |
| $X_{max,GoP}$ [byte] | 210971 | 163857 | 106085 | 64375 | 38104 | 22389 | 12165 | 6356 | 3206 | 1535 | 674 |
| $\overline{X}_{GoP}$ [byte] | 179504.00 | 132977.06 | 76476.29 | 42142.68 | 23306.52 | 13083.16 | 6741.77 | 3447.55 | 1778.00 | 955.39 | 488.55 |
| $S^2_{X,GoP}$ | 322008210.13 | 303239404.13 | 276673349.48 | 159149249.49 | 70175931.99 | 26848586.21 | 7878839.78 | 1791887.79 | 362037.53 | 56501.65 | 4601.52 |
| $CoV_{GoP}$ | 0.10 | 0.13 | 0.22 | 0.30 | 0.36 | 0.40 | 0.42 | 0.39 | 0.34 | 0.25 | 0.14 |
| **Mean GoP rate** [bit/s] | 3988.98 | 2955.05 | 1699.47 | 936.50 | 517.92 | 290.74 | 149.82 | 76.61 | 39.51 | 21.23 | 10.86 |
| **Peak GoP rate** [bit/s] | 4688.24 | 3641.27 | 2357.44 | 1430.56 | 846.76 | 497.53 | 270.33 | 141.24 | 71.24 | 34.11 | 14.98 |
| **Peak to mean** | 1.18 | 1.23 | 1.39 | 1.53 | 1.63 | 1.71 | 1.80 | 1.84 | 1.80 | 1.61 | 1.38 |

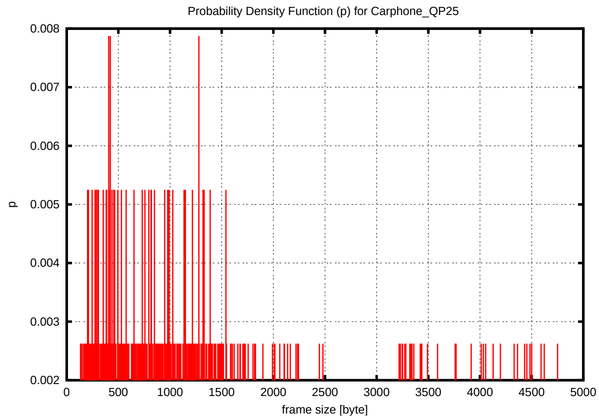


Figure 47: Frame size distribution for *Carphone (quantization 25)*

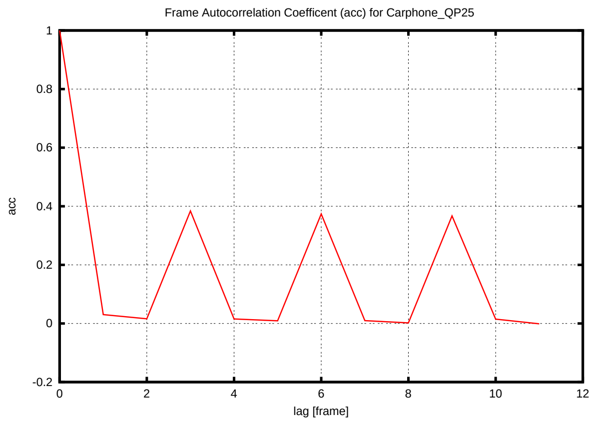


Figure 48: Autocorrelation coefficients for *Carphone (quantization 25)*

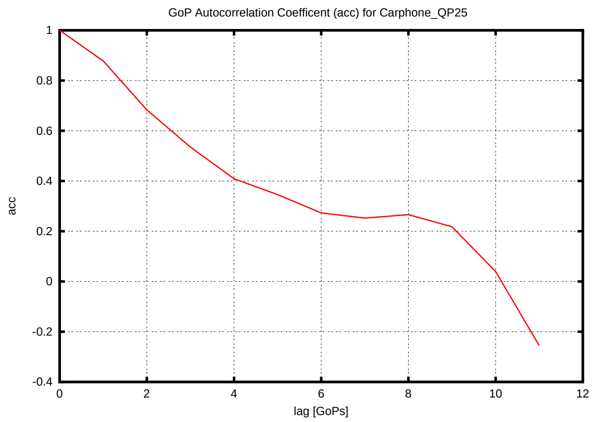


Figure 49: GoP autocorrelation coefficients for *Carphone (quantization 25)*

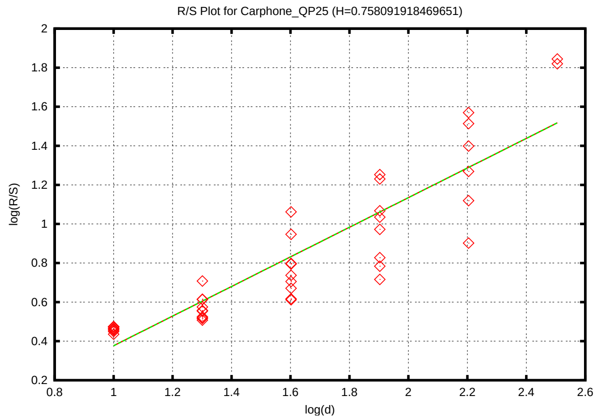


Figure 50: Single Frame R/S plot and for *Carphone (quantization 25)*

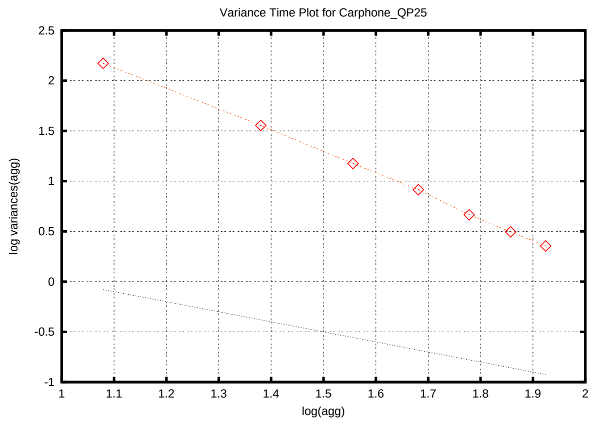


Figure 51: Variance time plot for *Carphone (quantization 25)*

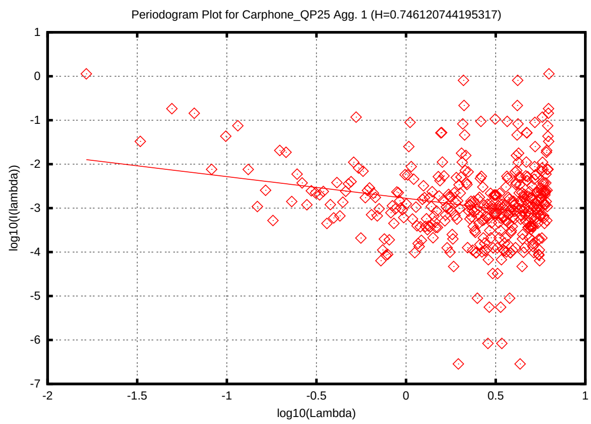


Figure 52: Single frame periodogram plot for *Carphone (quantization 25)*

# B Claire

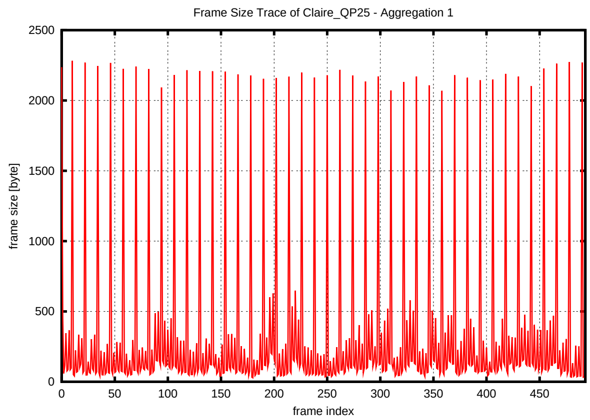


Figure 53: Frame size trace for *Claire (quantization 25)*

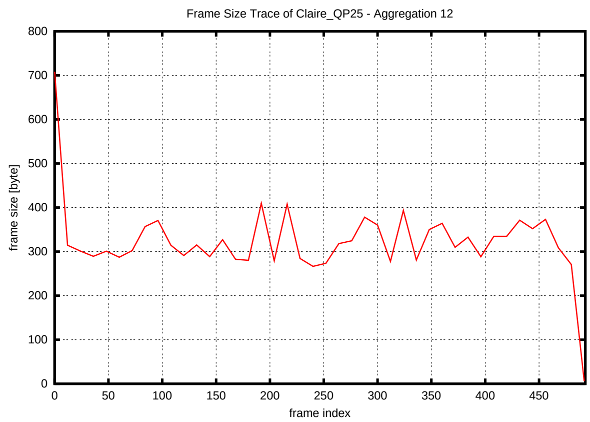


Figure 54: Frame size trace for one GoP *Claire (quantization 25)*

Table 6: Single frame statistics for *Claire*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 7374 | 4175 | 910 | 237 | 62 | 28 | 19 | 18 | 16 | 13 | 12 |
| $X_{max}$ [byte] | 14857 | 11505 | 7172 | 4885 | 3376 | 2282 | 1465 | 873 | 522 | 356 | 218 |
| $\overline{X}$ [byte] | 9185.88 | 5809.70 | 2204.47 | 1063.37 | 558.76 | 319.80 | 184.55 | 107.13 | 67.89 | 49.01 | 35.16 |
| $\overline{X}_{I-frame}$ [byte] | 14670.48 | 11334.17 | 7010.14 | 4729.36 | 3250.19 | 2187.36 | 1404.93 | 803.69 | 486.12 | 321.50 | 199.40 |
| $\overline{X}_{P-frame}$ [byte] | 10026.28 | 6424.75 | 2469.33 | 1199.46 | 601.60 | 312.39 | 157.21 | 85.59 | 50.03 | 30.35 | 16.89 |
| $\overline{X}_{B-frame}$ [byte] | 8168.44 | 4871.65 | 1489.79 | 542.91 | 198.06 | 83.43 | 38.54 | 26.02 | 21.04 | 21.12 | 20.98 |
| $S_X^2$ | 3534886.27 | 3368796.37 | 2393434.54 | 1363059.72 | 717295.04 | 339236.00 | 142775.82 | 46314.36 | 16607.09 | 6986.96 | 2533.58 |
| $CoV$ | 0.20 | 0.32 | 0.70 | 1.10 | 1.52 | 1.82 | 2.05 | 2.01 | 1.90 | 1.71 | 1.43 |
| **Mean bitrate** [bit/s] | 2204612.25 | 1394327.46 | 529072.45 | 255208.11 | 134102.07 | 76750.83 | 44292.90 | 25712.13 | 16294.69 | 11762.92 | 8437.97 |
| **Peak bitrate** [bit/s] | 3961866.67 | 3068000.00 | 1912533.33 | 1302666.67 | 900266.67 | 608533.33 | 390666.67 | 232800.00 | 139200.00 | 94933.33 | 58133.33 |
| **Peak to mean** | 1.80 | 2.20 | 3.61 | 5.10 | 6.71 | 7.93 | 8.82 | 9.05 | 8.54 | 8.07 | 6.89 |

Table 7: GoP statistics for *Claire*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 14838 | 11442 | 7113 | 4811 | 3277 | 2137 | 1339 | 763 | 452 | 321 | 202 |
| $X_{max,GoP}$ [byte] | 116473 | 74897 | 30900 | 15862 | 8468 | 4868 | 2704 | 1506 | 926 | 648 | 450 |
| $\overline{X}_{GoP}$ [byte] | 109731.95 | 69374.22 | 26288.00 | 12656.05 | 6633.24 | 3788.37 | 2182.61 | 1266.85 | 802.85 | 580.29 | 416.85 |
| $S^2_{X,GoP}$ | 12825367.00 | 7667306.08 | 4689651.30 | 2051947.55 | 707354.14 | 214328.09 | 44381.44 | 11194.38 | 2814.48 | 952.86 | 229.18 |
| $CoV_{GoP}$ | 0.03 | 0.04 | 0.08 | 0.11 | 0.13 | 0.12 | 0.10 | 0.08 | 0.07 | 0.05 | 0.04 |
| **Mean GoP rate** [bit/s] | 2438.49 | 1541.65 | 584.18 | 281.25 | 147.41 | 84.19 | 48.50 | 28.15 | 17.84 | 12.90 | 9.26 |
| **Peak GoP rate** [bit/s] | 2588.29 | 1664.38 | 686.67 | 352.49 | 188.18 | 108.18 | 60.09 | 33.47 | 20.58 | 14.40 | 10.00 |
| **Peak to mean** | 1.06 | 1.08 | 1.18 | 1.25 | 1.28 | 1.28 | 1.24 | 1.19 | 1.15 | 1.12 | 1.08 |

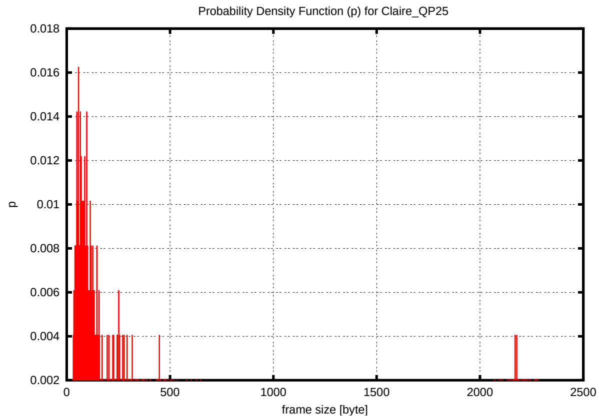


Figure 55: Frame size distribution for *Claire (quantization 25)*

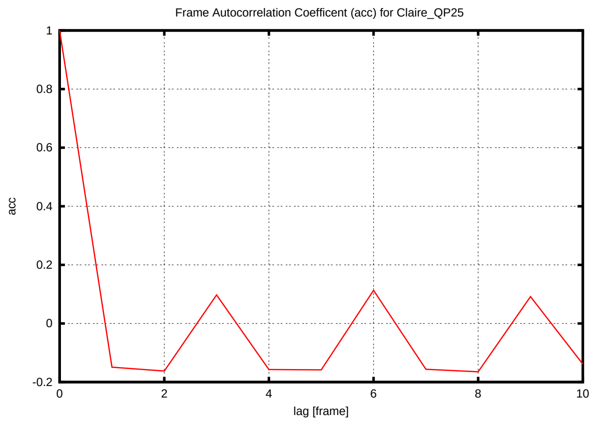


Figure 56: Autocorrelation coefficients for *Claire (quantization 25)*

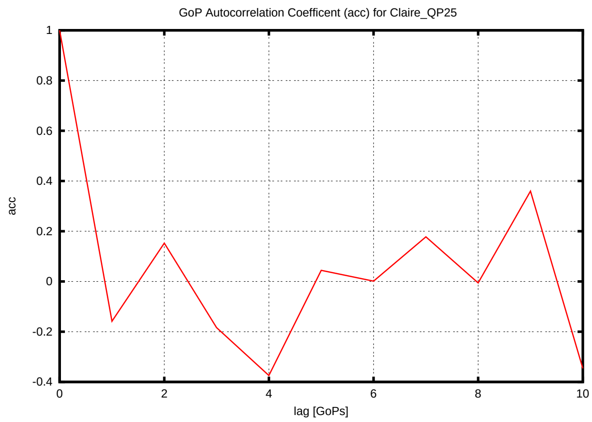


Figure 57: GoP autocorrelation coefficients for *Claire (quantization 25)*

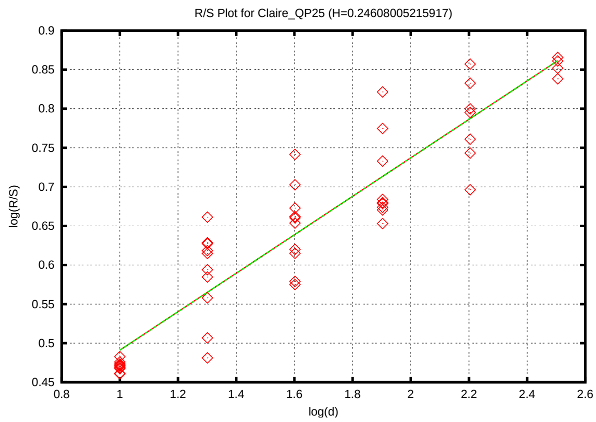


Figure 58: Single Frame R/S plot and for *Claire (quantization 25)*

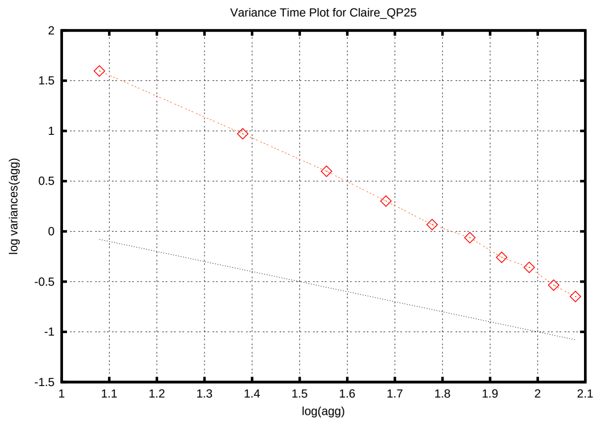


Figure 59: Variance time plot for *Claire (quantization 25)*

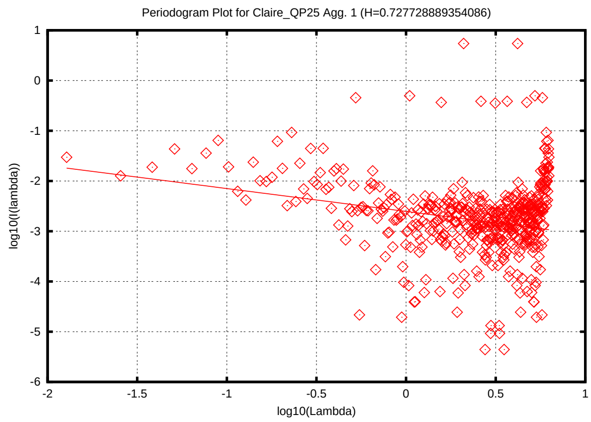


Figure 60: Single frame periodogram plot for *Claire (quantization 25)*

# C Container

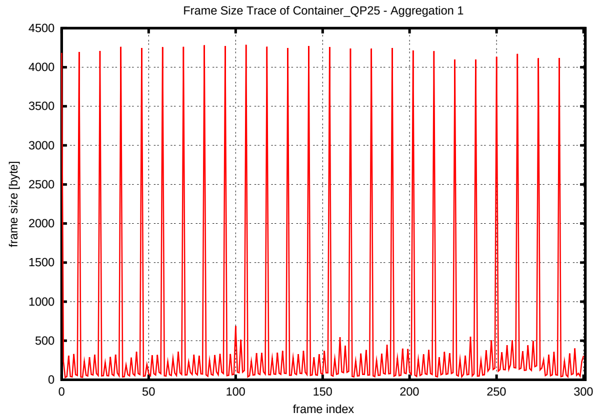


Figure 61: Frame size trace for *Container (quantization 25)*

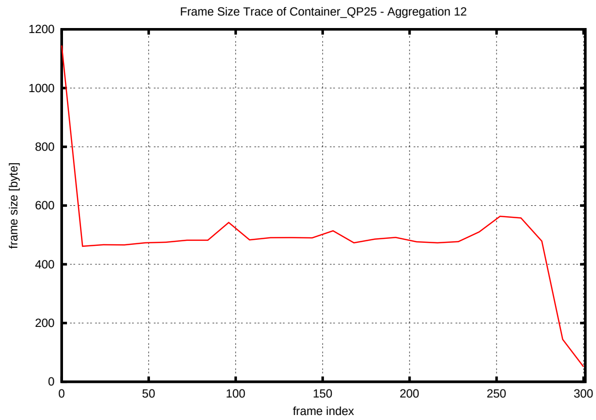


Figure 62: Frame size trace for one GoP *Container (quantization 25)*

Table 8: Single frame statistics for *Container*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 66 | 53 | 49 | 47 | 46 | 29 | 19 | 18 | 17 | 18 | 12 |
| $X_{max}$ [byte] | 20643 | 17174 | 12824 | 9287 | 6444 | 4287 | 2653 | 1590 | 951 | 512 | 252 |
| $\overline{X}$ [byte] | 10713.66 | 7197.36 | 3562.57 | 1718.55 | 887.30 | 484.36 | 274.47 | 164.70 | 100.78 | 61.99 | 38.25 |
| $\overline{X}_{I-frame}$ [byte] | 19746.08 | 16383.50 | 12187.46 | 8819.27 | 6101.58 | 4054.42 | 2509.42 | 1500.04 | 880.50 | 473.92 | 233.96 |
| $\overline{X}_{P-frame}$ [byte] | 12150.69 | 8341.67 | 4467.27 | 2241.43 | 897.59 | 336.95 | 133.71 | 71.96 | 43.07 | 28.57 | 17.77 |
| $\overline{X}_{B-frame}$ [byte] | 9000.56 | 5574.05 | 2102.08 | 599.38 | 205.58 | 75.53 | 36.70 | 25.88 | 21.05 | 20.97 | 20.48 |
| $S_X^2$ | 10925540.02 | 10368803.61 | 8629068.13 | 5565628.40 | 2806810.02 | 1280270.89 | 497151.75 | 177050.87 | 60277.83 | 16780.41 | 3776.45 |
| $CoV$ | 0.31 | 0.45 | 0.82 | 1.37 | 1.89 | 2.34 | 2.57 | 2.55 | 2.44 | 2.09 | 1.61 |
| **Mean bitrate** [bit/s] | 2571278.67 | 1727366.11 | 855017.94 | 412451.56 | 212950.96 | 116245.32 | 65871.63 | 39527.44 | 24186.58 | 14877.61 | 9179.00 |
| **Peak bitrate** [bit/s] | 5504800.00 | 4579733.33 | 3419733.33 | 2476533.33 | 1718400.00 | 1143200.00 | 707466.67 | 424000.00 | 253600.00 | 136533.33 | 67200.00 |
| **Peak to mean** | 2.14 | 2.65 | 4.00 | 6.00 | 8.07 | 9.83 | 10.74 | 10.73 | 10.49 | 9.18 | 7.32 |

Table 9: GoP statistics for *Container*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 9871 | 6609 | 3098 | 1359 | 524 | 281 | 125 | 91 | 80 | 82 | 71 |
| $X_{max,GoP}$ [byte] | 131676 | 89129 | 45717 | 22736 | 11806 | 6611 | 3825 | 2290 | 1369 | 843 | 479 |
| $\overline{X}_{GoP}$ [byte] | 128166.12 | 86094.28 | 42618.04 | 20562.96 | 10632.64 | 5808.16 | 3294.96 | 1976.56 | 1208.44 | 741.60 | 456.40 |
| $S^2_{X,GoP}$ | 13760651.78 | 5555771.96 | 1843470.62 | 590539.29 | 225383.91 | 104331.89 | 29999.54 | 9025.67 | 2394.01 | 844.92 | 254.00 |
| $CoV_{GoP}$ | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 | 0.06 | 0.05 | 0.05 | 0.04 | 0.04 | 0.03 |
| **Mean GoP rate** [bit/s] | 2848.14 | 1913.21 | 947.07 | 456.95 | 236.28 | 129.07 | 73.22 | 43.92 | 26.85 | 16.48 | 10.14 |
| **Peak GoP rate** [bit/s] | 2926.13 | 1980.64 | 1015.93 | 505.24 | 262.36 | 146.91 | 85.00 | 50.89 | 30.42 | 18.73 | 10.64 |
| **Peak to mean** | 1.03 | 1.04 | 1.07 | 1.11 | 1.11 | 1.14 | 1.16 | 1.16 | 1.13 | 1.14 | 1.05 |

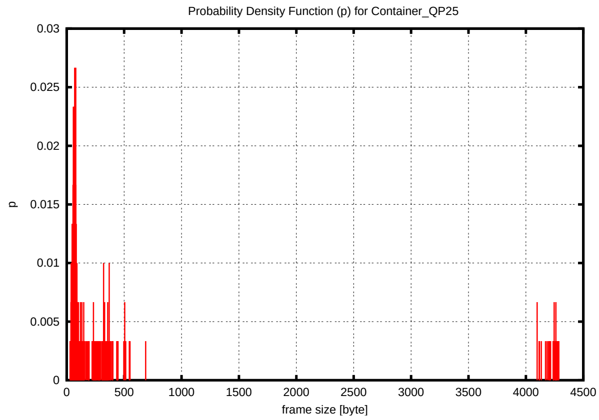


Figure 63: Frame size distribution for *Container (quantization 25)*

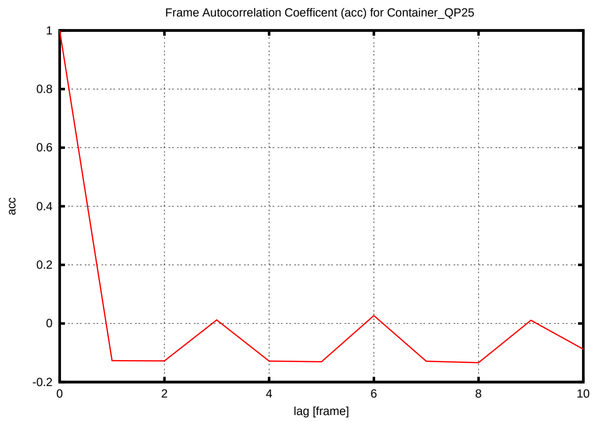


Figure 64: Autocorrelation coefficients for *Container (quantization 25)*

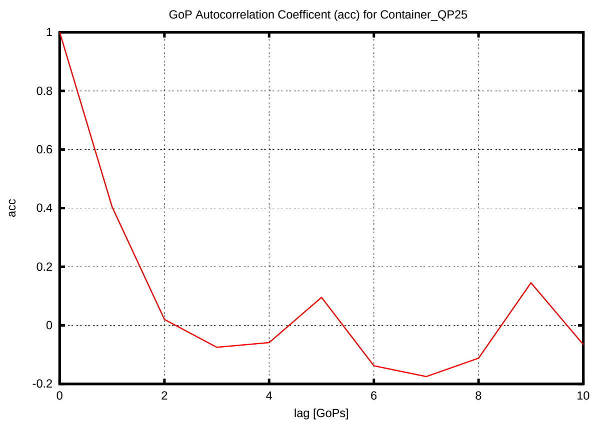


Figure 65: GoP autocorrelation coefficients for *Container (quantization 25)*

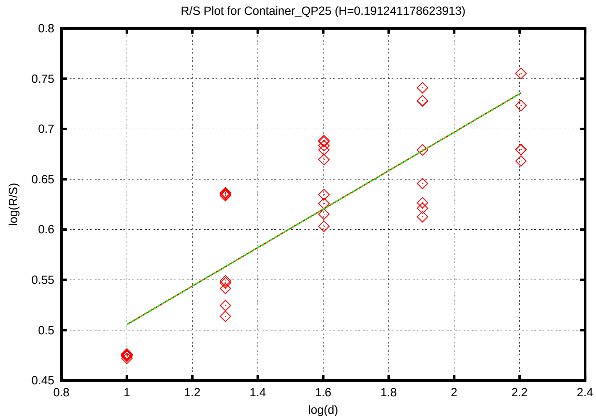


Figure 66: Single Frame R/S plot and for *Container (quantization 25)*

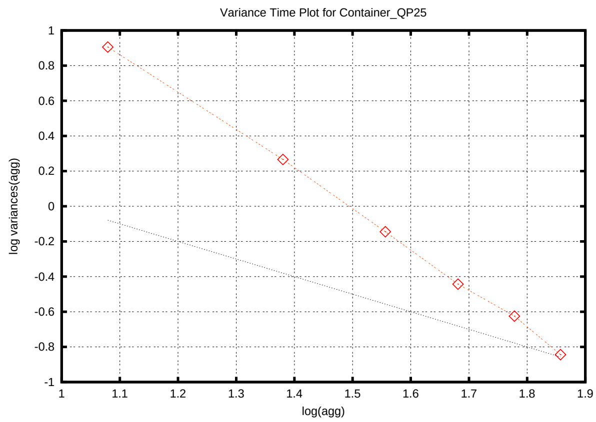


Figure 67: Variance time plot for *Container (quantization 25)*

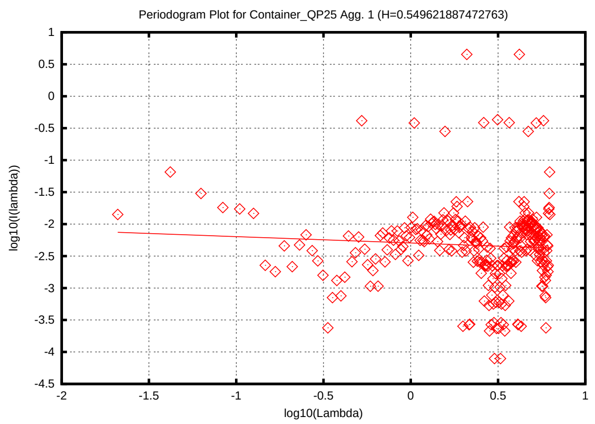


Figure 68: Single frame periodogram plot for *Container (quantization 25)*

# D Foreman

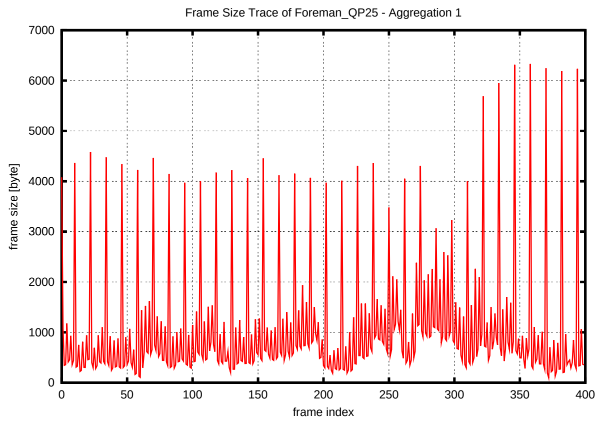


Figure 69: Frame size trace for *Foreman (quantization 25)*

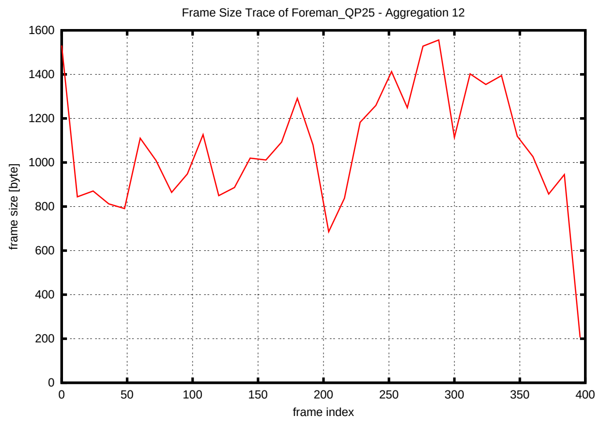


Figure 70: Frame size trace for one GoP *Foreman (quantization 25)*

Table 10: Single frame statistics for *Foreman*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 11934 | 8191 | 3896 | 1437 | 366 | 98 | 25 | 18 | 18 | 18 | 13 |
| $X_{max}$ [byte] | 26002 | 22598 | 18056 | 13621 | 9624 | 6330 | 3711 | 1942 | 998 | 563 | 229 |
| $\overline{X}$ [byte] | 15859.57 | 12027.09 | 7430.07 | 4143.16 | 2083.85 | 1038.28 | 508.08 | 268.82 | 149.76 | 88.48 | 46.30 |
| $\overline{X}_{I-frame}$ [byte] | 22705.41 | 19203.47 | 14466.41 | 10417.53 | 7064.97 | 4517.91 | 2676.38 | 1518.47 | 855.18 | 479.15 | 202.24 |
| $\overline{X}_{P-frame}$ [byte] | 17019.51 | 12977.53 | 8255.13 | 4791.12 | 2489.03 | 1263.00 | 597.06 | 305.16 | 173.11 | 103.98 | 47.07 |
| $\overline{X}_{B-frame}$ [byte] | 14548.47 | 10752.51 | 6220.52 | 3097.59 | 1294.85 | 509.03 | 197.47 | 95.42 | 50.82 | 32.72 | 26.09 |
| $S_X^2$ | 6843517.97 | 7032503.69 | 6655068.24 | 4976908.85 | 2950084.04 | 1384889.19 | 517421.78 | 166606.66 | 52443.34 | 16148.91 | 2621.91 |
| $CoV$ | 0.16 | 0.22 | 0.35 | 0.54 | 0.82 | 1.13 | 1.42 | 1.52 | 1.53 | 1.44 | 1.11 |
| **Mean bitrate** [bit/s] | 3806296.80 | 2886502.80 | 1783216.80 | 994359.60 | 500124.60 | 249187.20 | 121938.60 | 64516.20 | 35942.40 | 21235.20 | 11113.20 |
| **Peak bitrate** [bit/s] | 6933866.67 | 6026133.33 | 4814933.33 | 3632266.67 | 2566400.00 | 1688000.00 | 989600.00 | 517866.67 | 266133.33 | 150133.33 | 61066.67 |
| **Peak to mean** | 1.82 | 2.09 | 2.70 | 3.65 | 5.13 | 6.77 | 8.12 | 8.03 | 7.40 | 7.07 | 5.49 |

Table 11: GoP statistics for *Foreman*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 22042 | 18466 | 13665 | 9672 | 6462 | 4014 | 2240 | 1238 | 665 | 356 | 161 |
| $X_{max,GoP}$ [byte] | 215273 | 169960 | 115495 | 67853 | 34703 | 18636 | 9719 | 5142 | 2746 | 1592 | 925 |
| $\overline{X}_{GoP}$ [byte] | 189043.21 | 143259.39 | 88366.79 | 49212.30 | 24725.67 | 12321.82 | 6030.45 | 3193.30 | 1781.09 | 1053.45 | 552.06 |
| $S^2_{X,GoP}$ | 137575944.55 | 124844010.00 | 115464961.17 | 59348651.09 | 22080629.98 | 6719176.22 | 1854297.07 | 530968.09 | 143635.77 | 45380.44 | 12853.43 |
| $CoV_{GoP}$ | 0.06 | 0.08 | 0.12 | 0.16 | 0.19 | 0.21 | 0.23 | 0.23 | 0.21 | 0.20 | 0.21 |
| **Mean GoP rate** [bit/s] | 4200.96 | 3183.54 | 1963.71 | 1093.61 | 549.46 | 273.82 | 134.01 | 70.96 | 39.58 | 23.41 | 12.27 |
| **Peak GoP rate** [bit/s] | 4783.84 | 3776.89 | 2566.56 | 1507.84 | 771.18 | 414.13 | 215.98 | 114.27 | 61.02 | 35.38 | 20.56 |
| **Peak to mean** | 1.14 | 1.19 | 1.31 | 1.38 | 1.40 | 1.51 | 1.61 | 1.61 | 1.54 | 1.51 | 1.68 |

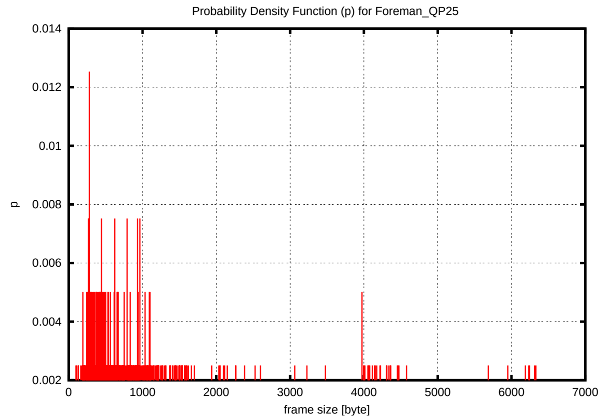


Figure 71: Frame size distribution for *Foreman (quantization 25)*

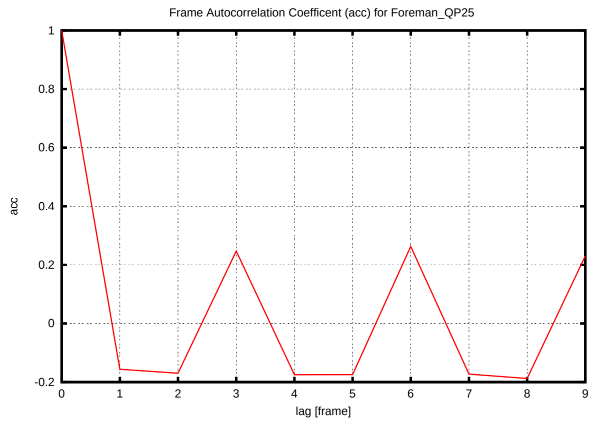


Figure 72: Autocorrelation coefficients for *Foreman (quantization 25)*

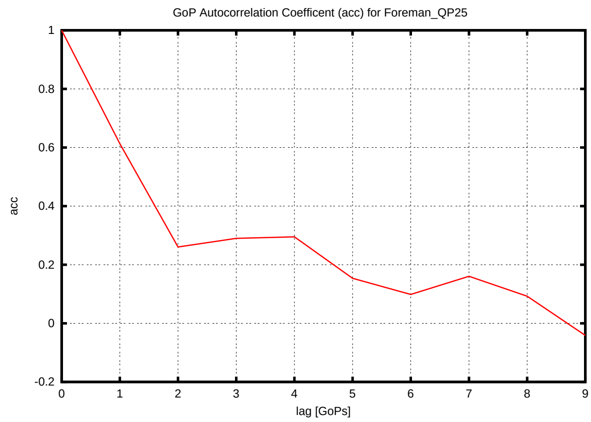


Figure 73: GoP autocorrelation coefficients for *Foreman (quantization 25)*

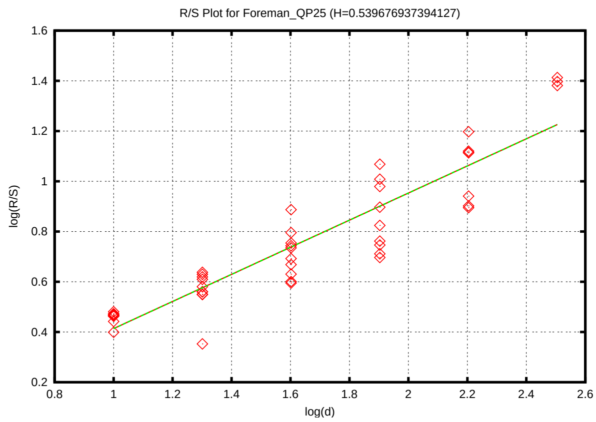


Figure 74: Single Frame R/S plot and for *Foreman (quantization 25)*

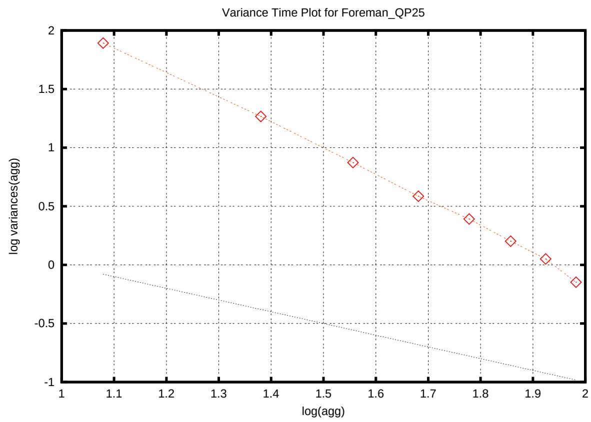


Figure 75: Variance time plot for *Foreman (quantization 25)*

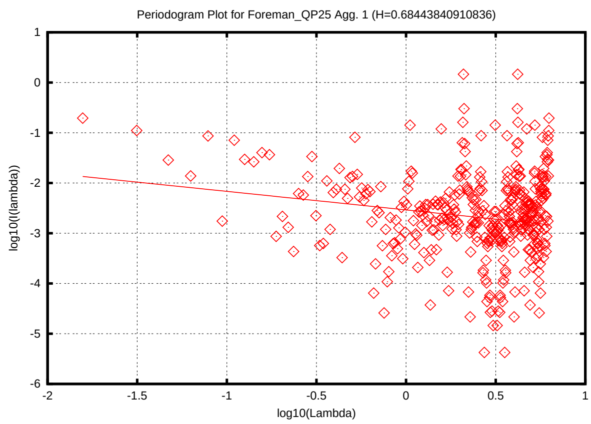


Figure 76: Single frame periodogram plot for *Foreman (quantization 25)*

# E Grandma

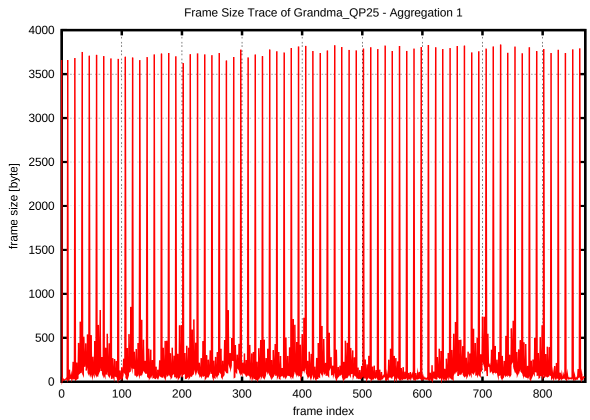


Figure 77: Frame size trace for *Grandma (quantization 25)*

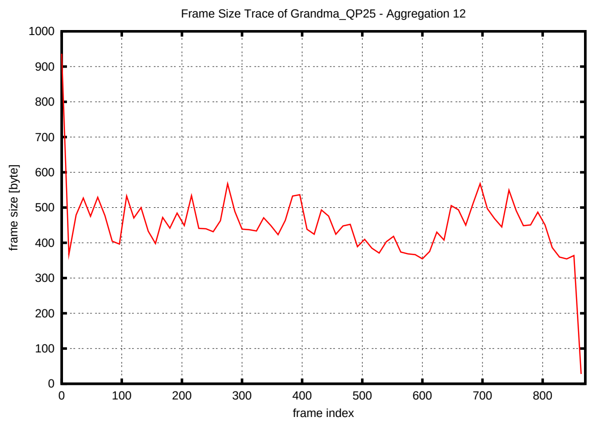


Figure 78: Frame size trace for one GoP *Grandma (quantization 25)*

Table 12: Single frame statistics for *Grandma*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 79 | 66 | 63 | 60 | 25 | 12 | 13 | 14 | 12 | 12 | 12 |
| $X_{max}$ [byte] | 21225 | 17726 | 12842 | 8941 | 6072 | 3835 | 2196 | 1164 | 620 | 351 | 193 |
| $\overline{X}$ [byte] | 12517.72 | 8698.79 | 3934.27 | 1641.42 | 819.85 | 443.58 | 235.39 | 127.04 | 71.62 | 47.46 | 32.74 |
| $\overline{X}_{I-frame}$ [byte] | 21046.27 | 17563.93 | 12663.37 | 8813.00 | 5959.07 | 3752.60 | 2134.27 | 1135.71 | 594.55 | 336.64 | 180.34 |
| $\overline{X}_{P-frame}$ [byte] | 13039.84 | 9114.05 | 4245.24 | 1668.12 | 735.70 | 329.54 | 138.95 | 64.66 | 33.10 | 19.71 | 14.14 |
| $\overline{X}_{B-frame}$ [byte] | 11248.05 | 7426.93 | 2718.73 | 728.75 | 204.64 | 69.96 | 32.64 | 23.53 | 20.29 | 21.50 | 21.16 |
| $S_X^2$ | 7749794.93 | 8077830.08 | 7661600.88 | 5000822.72 | 2506370.30 | 1024230.69 | 333989.41 | 93752.49 | 25132.62 | 7677.73 | 2008.40 |
| $CoV$ | 0.22 | 0.33 | 0.70 | 1.36 | 1.93 | 2.28 | 2.46 | 2.41 | 2.21 | 1.85 | 1.37 |
| **Mean bitrate** [bit/s] | 3004252.49 | 2087710.13 | 944225.58 | 393940.57 | 196763.35 | 106458.60 | 56493.96 | 30489.37 | 17189.62 | 11391.04 | 7858.55 |
| **Peak bitrate** [bit/s] | 5660000.00 | 4726933.33 | 3424533.33 | 2384266.67 | 1619200.00 | 1022666.67 | 585600.00 | 310400.00 | 165333.33 | 93600.00 | 51466.67 |
| **Peak to mean** | 1.88 | 2.26 | 3.63 | 6.05 | 8.23 | 9.61 | 10.37 | 10.18 | 9.62 | 8.22 | 6.55 |

Table 13: GoP statistics for *Grandma*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 20793 | 17392 | 12431 | 8657 | 5841 | 3661 | 2078 | 1112 | 584 | 337 | 185 |
| $X_{max,GoP}$ [byte] | 159866 | 113306 | 55601 | 25427 | 12850 | 6719 | 3382 | 1712 | 947 | 620 | 412 |
| $\overline{X}_{GoP}$ [byte] | 150107.83 | 104326.10 | 47213.83 | 19703.72 | 9826.38 | 5308.50 | 2814.12 | 1517.71 | 855.25 | 566.50 | 390.71 |
| $S^2_{X,GoP}$ | 44043766.96 | 33422801.69 | 24505395.41 | 11775843.98 | 2540153.93 | 419184.90 | 56350.73 | 7809.56 | 1474.08 | 435.38 | 96.77 |
| $CoV_{GoP}$ | 0.04 | 0.06 | 0.10 | 0.17 | 0.16 | 0.12 | 0.08 | 0.06 | 0.04 | 0.04 | 0.03 |
| **Mean GoP rate** [bit/s] | 3335.73 | 2318.36 | 1049.20 | 437.86 | 218.36 | 117.97 | 62.54 | 33.73 | 19.01 | 12.59 | 8.68 |
| **Peak GoP rate** [bit/s] | 3552.58 | 2517.91 | 1235.58 | 565.04 | 285.56 | 149.31 | 75.16 | 38.04 | 21.04 | 13.78 | 9.16 |
| **Peak to mean** | 1.07 | 1.09 | 1.18 | 1.29 | 1.31 | 1.27 | 1.20 | 1.13 | 1.11 | 1.09 | 1.05 |

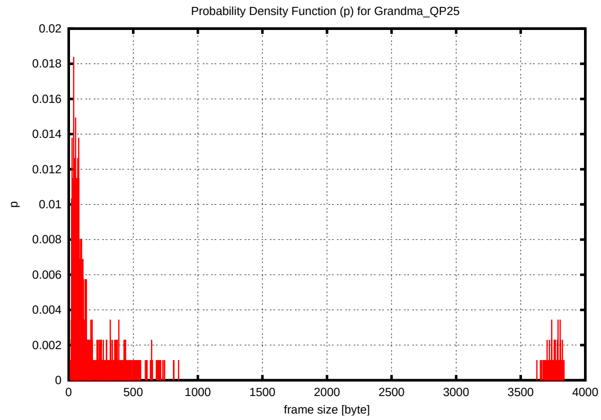


Figure 79: Frame size distribution for *Grandma (quantization 25)*

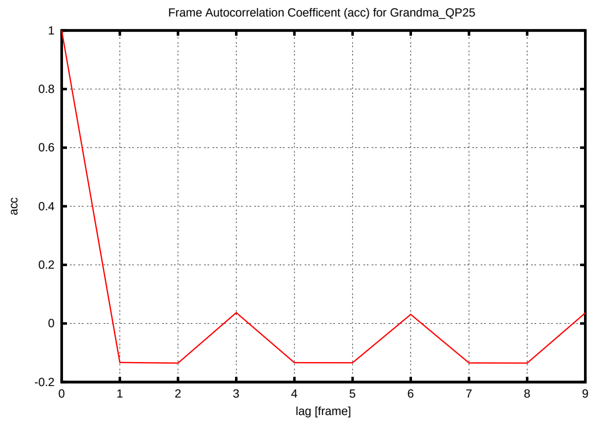


Figure 80: Autocorrelation coefficients for *Grandma (quantization 25)*

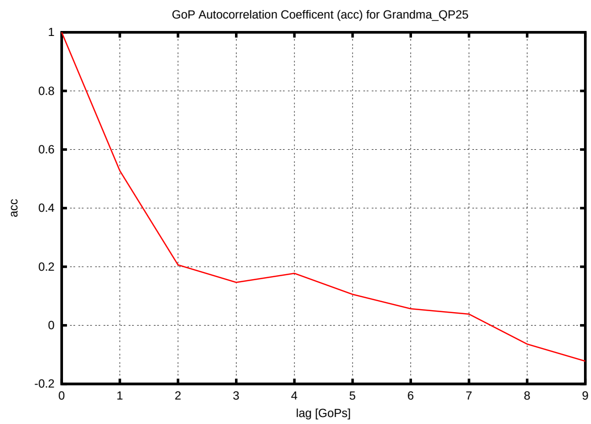


Figure 81: GoP autocorrelation coefficients for *Grandma (quantization 25)*

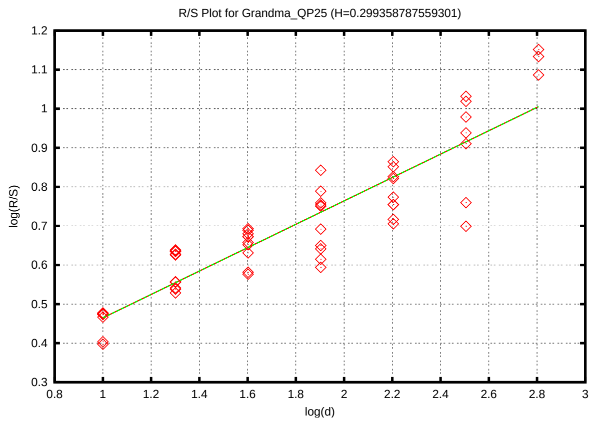


Figure 82: Single Frame R/S plot and for *Grandma (quantization 25)*

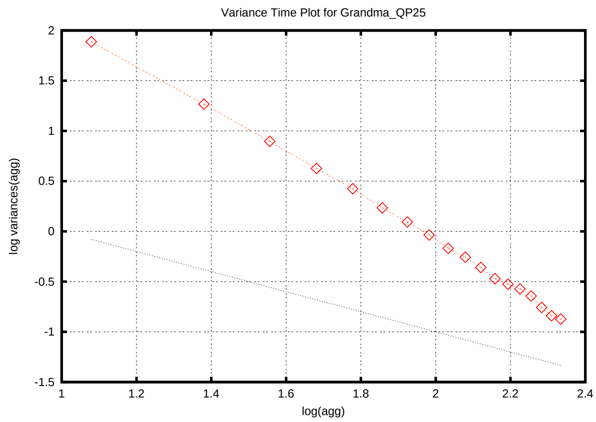


Figure 83: Variance time plot for *Grandma (quantization 25)*

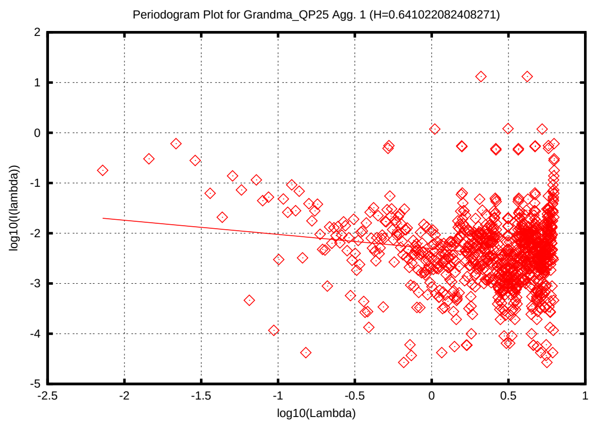


Figure 84: Single frame periodogram plot for *Grandma (quantization 25)*

# F Mobile

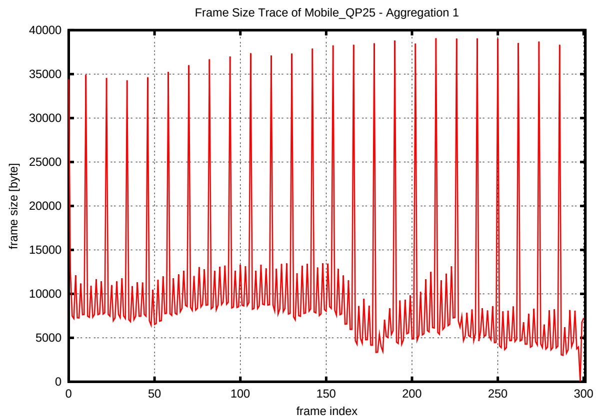


Figure 85: Frame size trace for *Mobile (quantization 25)*

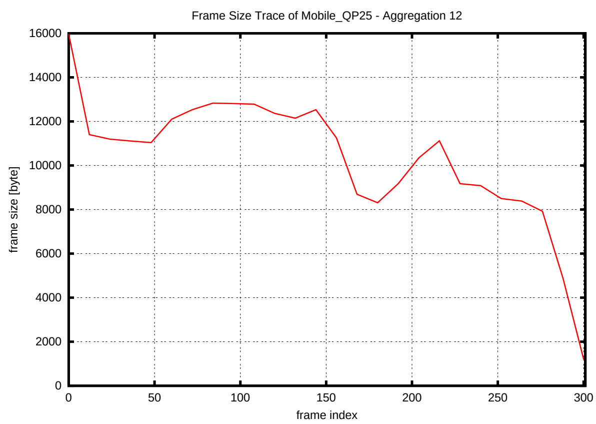


Figure 86: Frame size trace for one GoP *Mobile (quantization 25)*

Table 14: Single frame statistics for *Mobile*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 79 | 75 | 74 | 74 | 73 | 72 | 74 | 67 | 65 | 64 | 46 |
| $X_{max}$ [byte] | 118449 | 104711 | 86616 | 69247 | 53186 | 39074 | 26588 | 16708 | 9866 | 5577 | 2736 |
| $\overline{X}$ [byte] | 78447.45 | 63461.26 | 45667.85 | 30220.46 | 18019.59 | 10082.94 | 4901.11 | 2375.40 | 1233.03 | 664.55 | 332.98 |
| $\overline{X}_{I-frame}$ [byte] | 110923.58 | 97590.00 | 80210.50 | 63816.38 | 48859.65 | 35838.65 | 24329.08 | 15357.54 | 9079.12 | 5041.85 | 2286.15 |
| $\overline{X}_{P-frame}$ [byte] | 81525.07 | 65983.00 | 47931.80 | 32101.59 | 19313.73 | 10757.08 | 4997.15 | 2123.81 | 992.97 | 508.31 | 236.59 |
| $\overline{X}_{B-frame}$ [byte] | 73071.45 | 58078.87 | 40328.32 | 25147.56 | 13525.08 | 6481.90 | 2339.45 | 782.07 | 303.06 | 154.10 | 115.22 |
| $S_X^2$ | 173892464.32 | 172401258.95 | 162907058.36 | 143381795.82 | 112798949.30 | 74132068.26 | 39752235.72 | 17204631.18 | 6228006.61 | 1934733.66 | 386327.71 |
| $CoV$ | 0.17 | 0.21 | 0.28 | 0.40 | 0.59 | 0.85 | 1.29 | 1.75 | 2.02 | 2.09 | 1.87 |
| **Mean bitrate** [bit/s] | 18827388.44 | 15230702.19 | 10960283.32 | 7252910.03 | 4324701.93 | 2419905.65 | 1176265.51 | 570096.48 | 295926.38 | 159493.16 | 79916.01 |
| **Peak bitrate** [bit/s] | 31586400.00 | 27922933.33 | 23097600.00 | 18465866.67 | 14182933.33 | 10419733.33 | 7090133.33 | 4455466.67 | 2630933.33 | 1487200.00 | 729600.00 |
| **Peak to mean** | 1.68 | 1.83 | 2.11 | 2.55 | 3.28 | 4.31 | 6.03 | 7.82 | 8.89 | 9.32 | 9.13 |

Table 15: GoP statistics for *Mobile*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 71069 | 56372 | 38929 | 24530 | 13338 | 6845 | 2715 | 1175 | 637 | 425 | 343 |
| $X_{max,GoP}$ [byte] | 1003069 | 822632 | 606466 | 413229 | 254439 | 145046 | 68723 | 30744 | 15726 | 8452 | 4410 |
| $\overline{X}_{GoP}$ [byte] | 938793.00 | 759538.24 | 546694.64 | 361875.12 | 215866.12 | 120834.60 | 58777.88 | 28496.20 | 14792.16 | 7968.00 | 3983.48 |
| $S^2_{X,GoP}$ | 2784281504.08 | 2413707187.77 | 2067802530.91 | 1533374464.78 | 904409033.19 | 357694740.42 | 49164767.61 | 2893049.33 | 539169.31 | 131375.17 | 47563.01 |
| $CoV_{GoP}$ | 0.06 | 0.06 | 0.08 | 0.11 | 0.14 | 0.16 | 0.12 | 0.06 | 0.05 | 0.05 | 0.05 |
| **Mean GoP rate** [bit/s] | 20862.07 | 16878.63 | 12148.77 | 8041.67 | 4797.02 | 2685.21 | 1306.18 | 633.25 | 328.71 | 177.07 | 88.52 |
| **Peak GoP rate** [bit/s] | 22290.42 | 18280.71 | 13477.02 | 9182.87 | 5654.20 | 3223.24 | 1527.18 | 683.20 | 349.47 | 187.82 | 98.00 |
| **Peak to mean** | 1.07 | 1.08 | 1.11 | 1.14 | 1.18 | 1.20 | 1.17 | 1.08 | 1.06 | 1.06 | 1.11 |

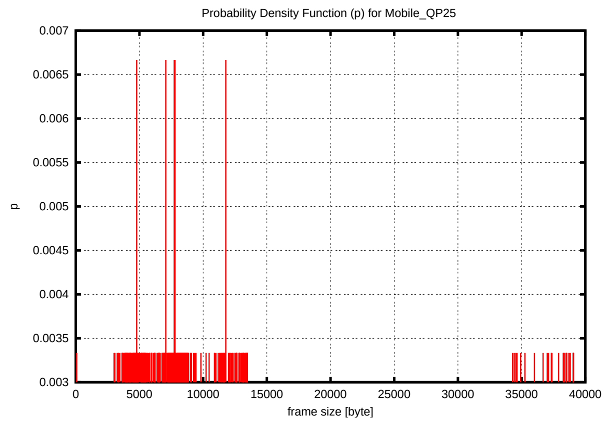


Figure 87: Frame size distribution for *Mobile (quantization 25)*

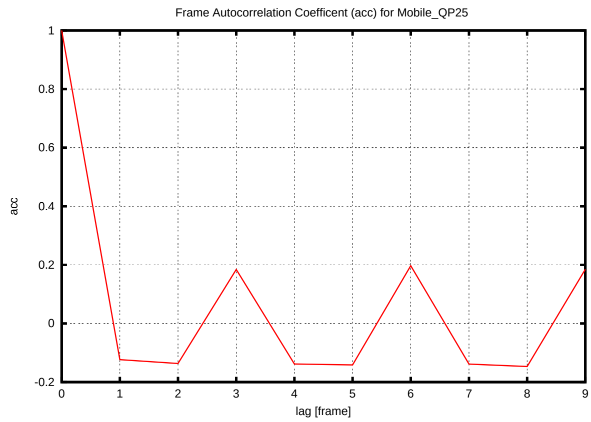


Figure 88: Autocorrelation coefficients for *Mobile (quantization 25)*

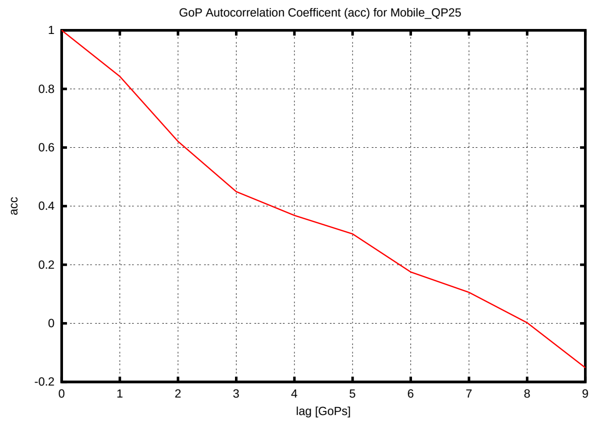


Figure 89: GoP autocorrelation coefficients for *Mobile (quantization 25)*

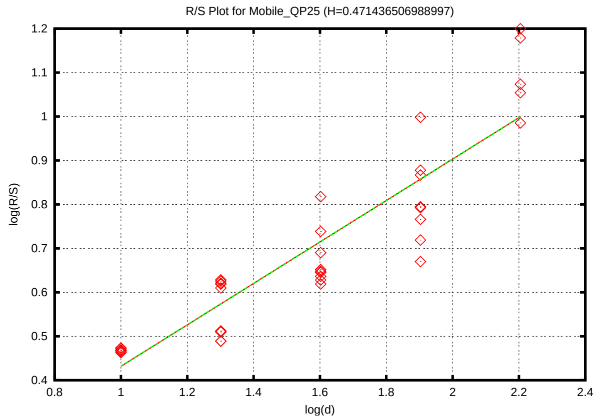


Figure 90: Single Frame R/S plot and for *Mobile (quantization 25)*

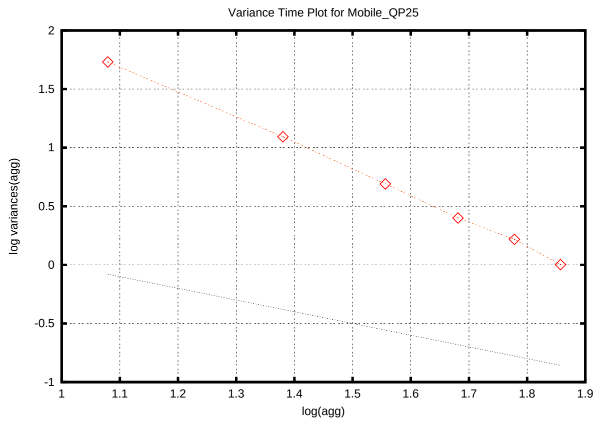


Figure 91: Variance time plot for *Mobile (quantization 25)*

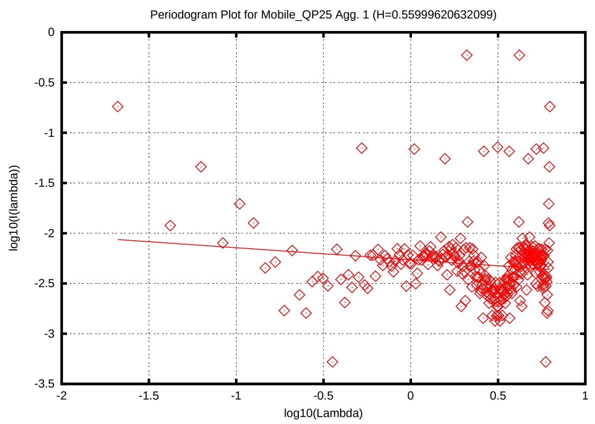


Figure 92: Single frame periodogram plot for *Mobile (quantization 25)*

# G Mother and Daughter

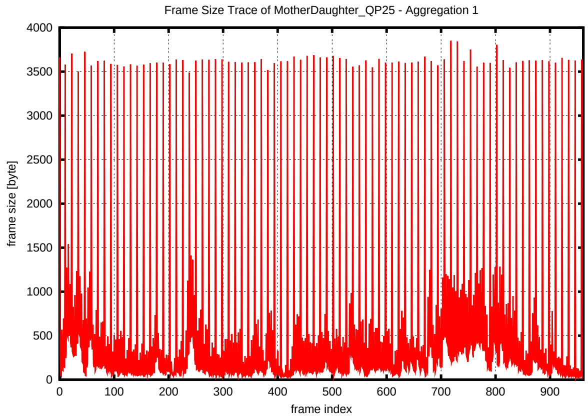


Figure 93: Frame size trace for *MotherDaughter (quantization 25)*

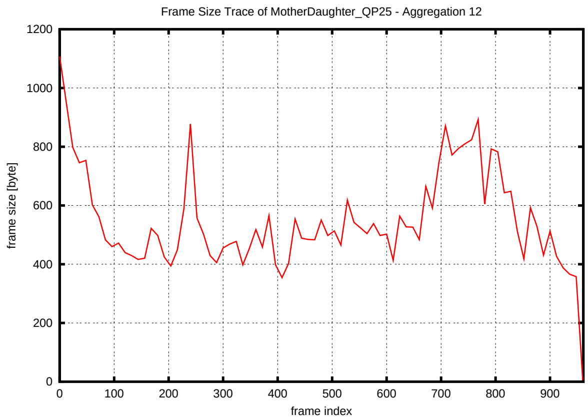


Figure 94: Frame size trace for one GoP *MotherDaughter (quantization 25)*

Table 16: Single frame statistics for *MotherDaughter*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 10214 | 6465 | 1825 | 155 | 31 | 20 | 18 | 17 | 13 | 12 | 12 |
| $X_{max}$ [byte] | 21320 | 17861 | 12948 | 9061 | 6086 | 3850 | 2203 | 1242 | 678 | 388 | 186 |
| $\overline{X}$ [byte] | 13283.92 | 9445.40 | 4715.64 | 2155.96 | 1033.42 | 540.14 | 276.77 | 148.07 | 83.37 | 52.38 | 32.81 |
| $\overline{X}_{I-frame}$ [byte] | 20627.19 | 17094.10 | 12171.16 | 8452.26 | 5714.84 | 3623.94 | 2086.59 | 1153.49 | 639.54 | 359.04 | 172.83 |
| $\overline{X}_{P-frame}$ [byte] | 14342.71 | 10355.73 | 5460.37 | 2632.17 | 1222.88 | 567.80 | 252.72 | 121.18 | 60.65 | 31.65 | 17.12 |
| $\overline{X}_{B-frame}$ [byte] | 11957.50 | 8135.98 | 3492.78 | 1180.51 | 369.88 | 139.48 | 56.73 | 30.91 | 21.50 | 21.34 | 20.98 |
| $S_X^2$ | 6762779.27 | 6964248.17 | 6448300.72 | 4378181.79 | 2260550.78 | 942619.15 | 316267.52 | 96161.95 | 29104.72 | 8773.79 | 1823.35 |
| $CoV$ | 0.20 | 0.28 | 0.54 | 0.97 | 1.45 | 1.80 | 2.03 | 2.09 | 2.05 | 1.79 | 1.30 |
| **Mean bitrate** [bit/s] | 3188141.27 | 2266894.90 | 1131754.34 | 517431.26 | 248021.39 | 129634.46 | 66424.81 | 35537.73 | 20009.66 | 12571.16 | 7874.80 |
| **Peak bitrate** [bit/s] | 5685333.33 | 4762933.33 | 3452800.00 | 2416266.67 | 1622933.33 | 1026666.67 | 587466.67 | 331200.00 | 180800.00 | 103466.67 | 49600.00 |
| **Peak to mean** | 1.78 | 2.10 | 3.05 | 4.67 | 6.54 | 7.92 | 8.84 | 9.32 | 9.04 | 8.23 | 6.30 |

Table 17: GoP statistics for *MotherDaughter*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 20705 | 17150 | 12261 | 8521 | 5702 | 3609 | 2069 | 1136 | 620 | 354 | 166 |
| $X_{max,GoP}$ [byte] | 179334 | 132017 | 74349 | 39366 | 20465 | 10895 | 5412 | 2627 | 1368 | 759 | 441 |
| $\overline{X}_{GoP}$ [byte] | 159059.77 | 113088.16 | 56449.56 | 25789.88 | 12342.04 | 6442.34 | 3297.82 | 1763.88 | 993.16 | 624.24 | 391.44 |
| $S^2_{X,GoP}$ | 92664051.77 | 80182193.07 | 71449789.49 | 37333825.10 | 10344312.75 | 2623134.20 | 509903.89 | 88790.95 | 13823.78 | 2401.83 | 295.46 |
| $CoV_{GoP}$ | 0.06 | 0.08 | 0.15 | 0.24 | 0.26 | 0.25 | 0.22 | 0.17 | 0.12 | 0.08 | 0.04 |
| **Mean GoP rate** [bit/s] | 3534.66 | 2513.07 | 1254.43 | 573.11 | 274.27 | 143.16 | 73.28 | 39.20 | 22.07 | 13.87 | 8.70 |
| **Peak GoP rate** [bit/s] | 3985.20 | 2933.71 | 1652.20 | 874.80 | 454.78 | 242.11 | 120.27 | 58.38 | 30.40 | 16.87 | 9.80 |
| **Peak to mean** | 1.13 | 1.17 | 1.32 | 1.53 | 1.66 | 1.69 | 1.64 | 1.49 | 1.38 | 1.22 | 1.13 |

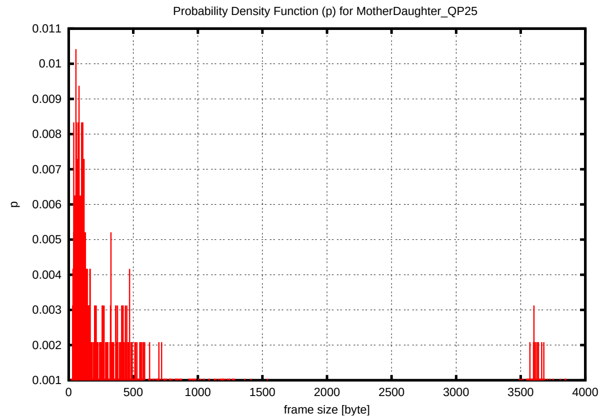


Figure 95: Frame size distribution for *MotherDaughter (quantization 25)*

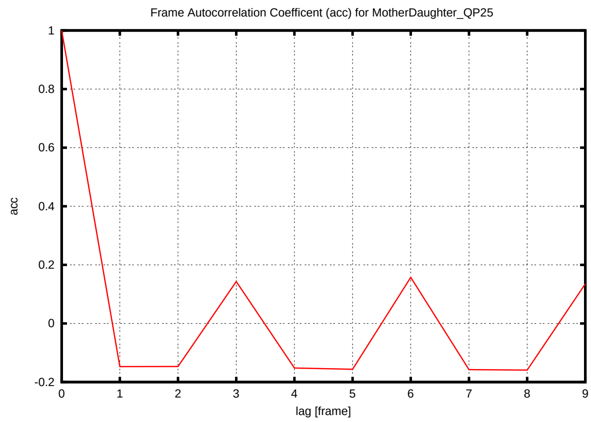


Figure 96: Autocorrelation coefficients for *MotherDaughter (quantization 25)*

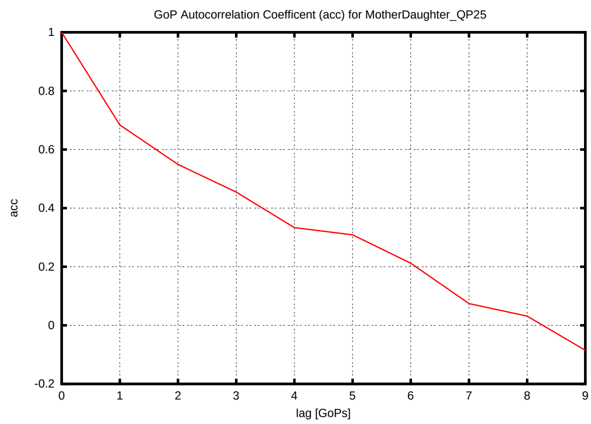


Figure 97: GoP autocorrelation coefficients for *MotherDaughter (quantization 25)*

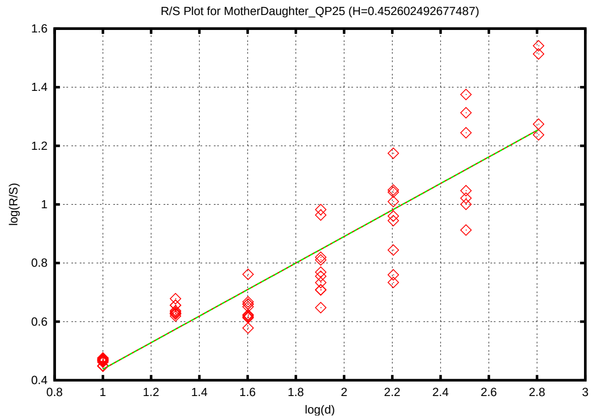


Figure 98: Single Frame R/S plot and for *MotherDaughter (quantization 25)*

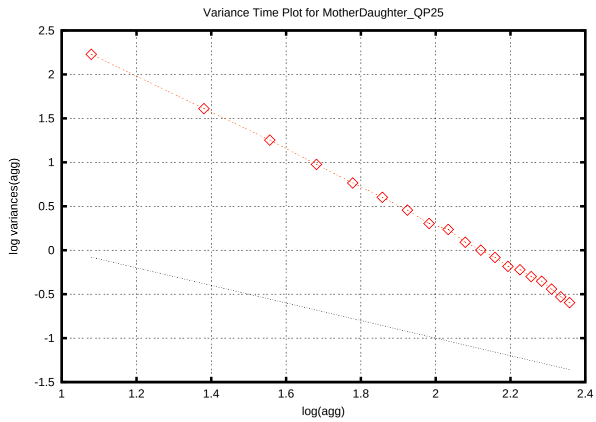


Figure 99: Variance time plot for *MotherDaughter (quantization 25)*

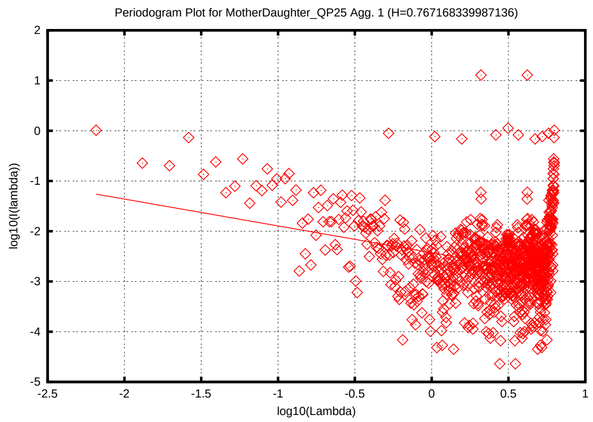


Figure 100: Single frame periodogram plot for *MotherDaughter (quantization 25)*

# H News

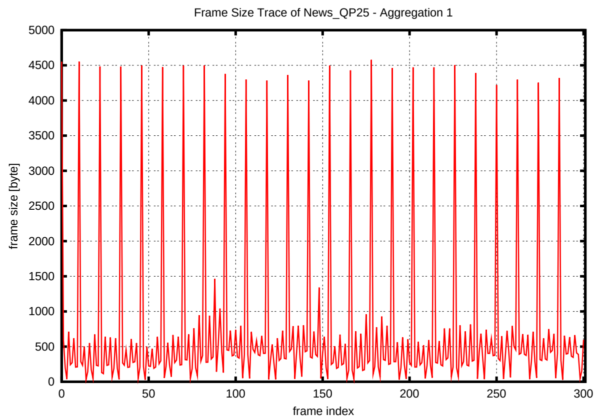


Figure 101: Frame size trace for *News (quantization 25)*

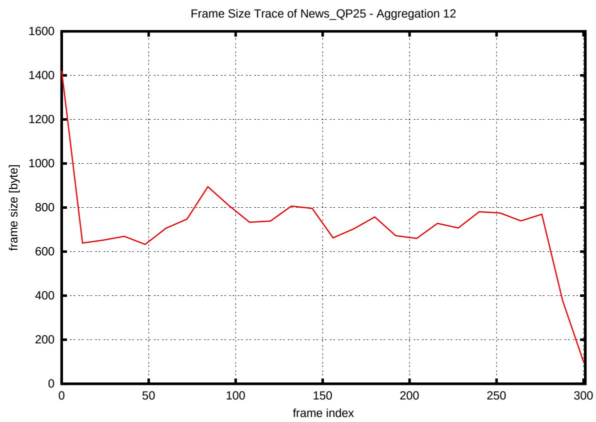


Figure 102: Frame size trace for one GoP *News (quantization 25)*

Table 18: Single frame statistics for *News*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 66 | 53 | 49 | 47 | 35 | 29 | 24 | 25 | 18 | 19 | 12 |
| $X_{max}$ [byte] | 20607 | 16954 | 12400 | 9038 | 6532 | 4578 | 2986 | 1884 | 1136 | 659 | 322 |
| $\overline{X}$ [byte] | 8178.24 | 5858.84 | 3241.40 | 1878.40 | 1143.00 | 708.14 | 420.70 | 249.22 | 145.55 | 85.65 | 47.18 |
| $\overline{X}_{I-frame}$ [byte] | 19658.62 | 16123.77 | 11767.04 | 8538.08 | 6129.62 | 4253.15 | 2774.92 | 1748.50 | 1062.04 | 613.58 | 300.88 |
| $\overline{X}_{P-frame}$ [byte] | 8678.81 | 6299.28 | 3612.41 | 2106.36 | 1223.21 | 698.45 | 375.32 | 203.47 | 113.11 | 59.67 | 25.57 |
| $\overline{X}_{B-frame}$ [byte] | 6498.08 | 4359.23 | 1993.93 | 927.16 | 464.66 | 250.91 | 131.66 | 71.47 | 38.57 | 26.77 | 22.30 |
| $S_X^2$ | 15257550.62 | 11957828.87 | 8091292.97 | 4830720.22 | 2639248.19 | 1307971.68 | 568005.92 | 228165.58 | 84816.08 | 27956.96 | 6394.70 |
| $CoV$ | 0.48 | 0.59 | 0.88 | 1.17 | 1.42 | 1.62 | 1.79 | 1.92 | 2.00 | 1.95 | 1.69 |
| **Mean bitrate** [bit/s] | 1962778.21 | 1406121.73 | 777935.68 | 450816.48 | 274320.00 | 169952.69 | 100967.44 | 59813.42 | 34931.56 | 20556.28 | 11323.06 |
| **Peak bitrate** [bit/s] | 5495200.00 | 4521066.67 | 3306666.67 | 2410133.33 | 1741866.67 | 1220800.00 | 796266.67 | 502400.00 | 302933.33 | 175733.33 | 85866.67 |
| **Peak to mean** | 2.80 | 3.22 | 4.25 | 5.35 | 6.35 | 7.18 | 7.89 | 8.40 | 8.67 | 8.55 | 7.58 |

Table 19: GoP statistics for *News*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 5516 | 3673 | 1596 | 679 | 291 | 203 | 108 | 97 | 91 | 92 | 82 |
| $X_{max,GoP}$ [byte] | 110009 | 81418 | 48524 | 29314 | 17599 | 10398 | 5936 | 3502 | 2022 | 1185 | 699 |
| $\overline{X}_{GoP}$ [byte] | 97934.20 | 70170.20 | 38838.04 | 22518.52 | 13710.60 | 8494.00 | 5049.48 | 2989.96 | 1744.56 | 1024.60 | 562.92 |
| $S^2_{X,GoP}$ | 18214575.17 | 13682735.25 | 9666299.62 | 4703273.51 | 1590853.83 | 503967.33 | 164397.01 | 72089.54 | 23891.09 | 8519.00 | 1724.74 |
| $CoV_{GoP}$ | 0.04 | 0.05 | 0.08 | 0.10 | 0.09 | 0.08 | 0.08 | 0.09 | 0.09 | 0.09 | 0.07 |
| **Mean GoP rate** [bit/s] | 2176.32 | 1559.34 | 863.07 | 500.41 | 304.68 | 188.76 | 112.21 | 66.44 | 38.77 | 22.77 | 12.51 |
| **Peak GoP rate** [bit/s] | 2444.64 | 1809.29 | 1078.31 | 651.42 | 391.09 | 231.07 | 131.91 | 77.82 | 44.93 | 26.33 | 15.53 |
| **Peak to mean** | 1.12 | 1.16 | 1.25 | 1.30 | 1.28 | 1.22 | 1.18 | 1.17 | 1.16 | 1.16 | 1.24 |

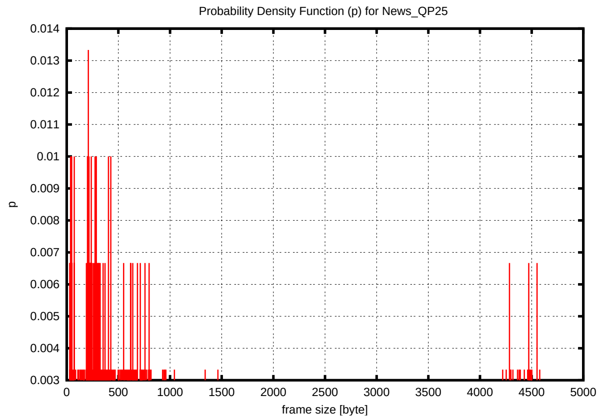


Figure 103: Frame size distribution for *News (quantization 25)*

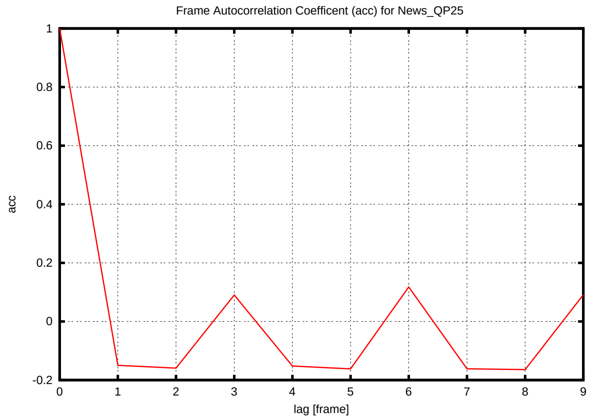


Figure 104: Autocorrelation coefficients for *News (quantization 25)*

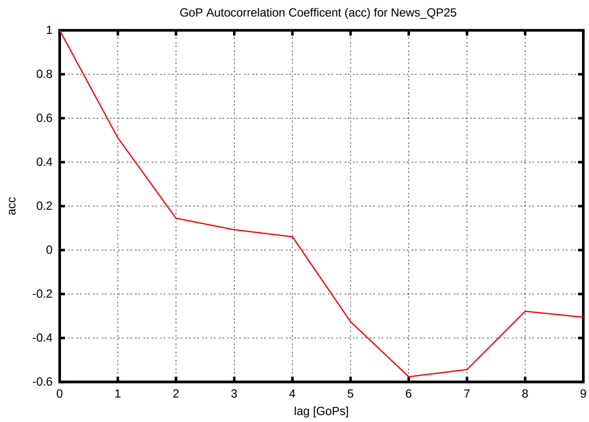


Figure 105: GoP autocorrelation coefficients for *News (quantization 25)*

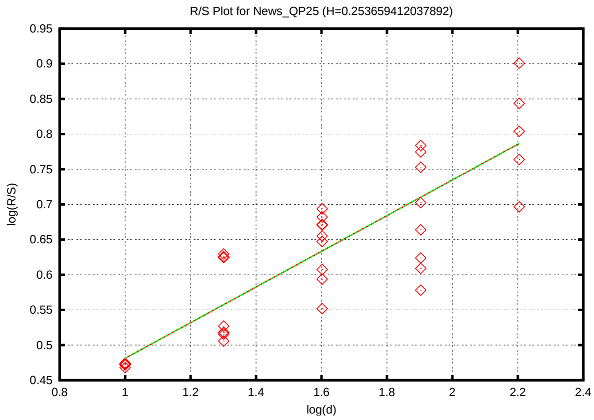


Figure 106: Single Frame R/S plot and for *News (quantization 25)*

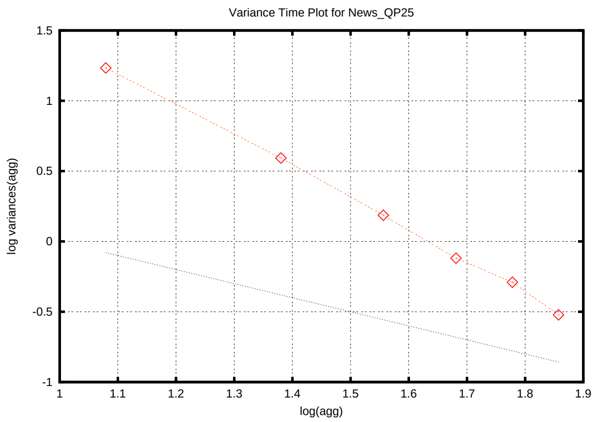


Figure 107: Variance time plot for *News (quantization 25)*

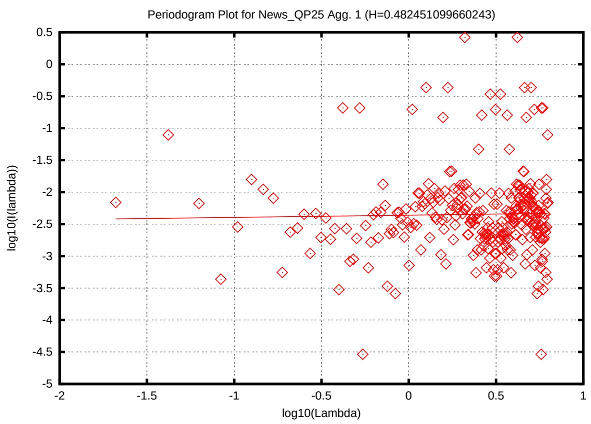


Figure 108: Single frame periodogram plot for *News (quantization 25)*

# I Salesman

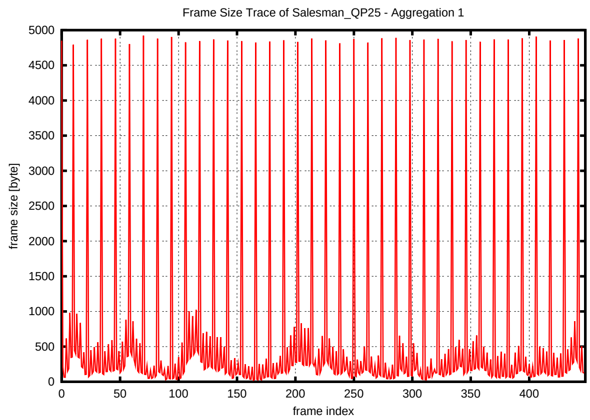


Figure 109: Frame size trace for *Salesman (quantization 25)*

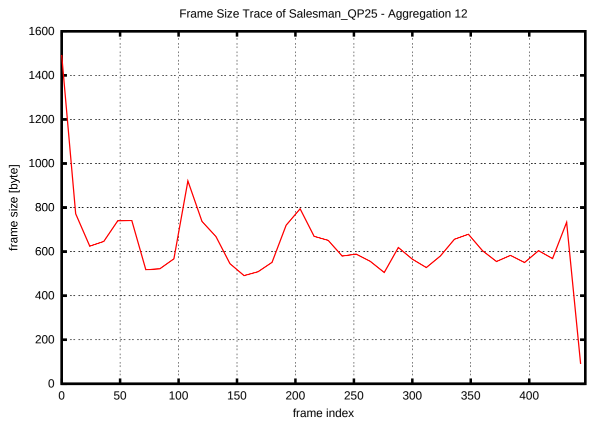


Figure 110: Frame size trace for one GoP *Salesman (quantization 25)*

Table 20: Single frame statistics for *Salesman*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 8785 | 5424 | 1409 | 279 | 63 | 21 | 19 | 17 | 17 | 13 | 12 |
| $X_{max}$ [byte] | 23117 | 19514 | 14851 | 10909 | 7551 | 4923 | 2886 | 1602 | 827 | 414 | 180 |
| $\overline{X}$ [byte] | 11398.27 | 7892.75 | 3651.02 | 1883.50 | 1063.37 | 620.98 | 339.81 | 186.37 | 96.95 | 53.21 | 31.84 |
| $\overline{X}_{I-frame}$ [byte] | 23019.95 | 19409.42 | 14755.39 | 10818.45 | 7489.53 | 4858.89 | 2829.50 | 1556.71 | 804.13 | 390.18 | 167.79 |
| $\overline{X}_{P-frame}$ [byte] | 11584.35 | 8008.38 | 3634.46 | 1730.08 | 880.11 | 464.29 | 232.79 | 120.03 | 56.66 | 25.89 | 15.20 |
| $\overline{X}_{B-frame}$ [byte] | 9846.37 | 6380.71 | 2241.25 | 801.81 | 312.81 | 139.47 | 62.56 | 33.85 | 21.92 | 20.50 | 20.76 |
| $S_X^2$ | 13435514.38 | 13087284.30 | 12064243.99 | 7697474.61 | 3940524.26 | 1703347.46 | 585238.80 | 179803.14 | 46879.78 | 10598.07 | 1726.92 |
| $CoV$ | 0.32 | 0.46 | 0.95 | 1.47 | 1.87 | 2.10 | 2.25 | 2.28 | 2.23 | 1.93 | 1.31 |
| **Mean bitrate** [bit/s] | 2735583.75 | 1894258.93 | 876244.82 | 452040.00 | 255209.46 | 149035.71 | 81555.00 | 44729.80 | 23268.75 | 12769.29 | 7641.43 |
| **Peak bitrate** [bit/s] | 6164533.33 | 5203733.33 | 3960266.67 | 2909066.67 | 2013600.00 | 1312800.00 | 769600.00 | 427200.00 | 220533.33 | 110400.00 | 48000.00 |
| **Peak to mean** | 2.25 | 2.75 | 4.52 | 6.44 | 7.89 | 8.81 | 9.44 | 9.55 | 9.48 | 8.65 | 6.28 |

Table 21: GoP statistics for *Salesman*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 23069 | 19511 | 14835 | 10909 | 7536 | 4851 | 2846 | 1548 | 797 | 395 | 165 |
| $X_{max,GoP}$ [byte] | 152586 | 109126 | 57158 | 32090 | 18538 | 10714 | 5586 | 2809 | 1387 | 733 | 421 |
| $\overline{X}_{GoP}$ [byte] | 135974.46 | 94106.35 | 43445.57 | 22362.19 | 12606.95 | 7356.57 | 4023.43 | 2171.30 | 1148.05 | 630.57 | 378.19 |
| $S^2_{X,GoP}$ | 37528558.81 | 28171359.46 | 24440230.14 | 11699782.60 | 3926095.77 | 1190301.36 | 266349.86 | 57525.99 | 8130.94 | 1162.14 | 175.10 |
| $CoV_{GoP}$ | 0.05 | 0.06 | 0.11 | 0.15 | 0.16 | 0.15 | 0.13 | 0.11 | 0.08 | 0.05 | 0.03 |
| **Mean GoP rate** [bit/s] | 3021.65 | 2091.25 | 965.46 | 496.94 | 280.15 | 163.48 | 89.41 | 48.25 | 25.51 | 14.01 | 8.40 |
| **Peak GoP rate** [bit/s] | 3390.80 | 2425.02 | 1270.18 | 713.11 | 411.96 | 238.09 | 124.13 | 62.42 | 30.82 | 16.29 | 9.36 |
| **Peak to mean** | 1.12 | 1.16 | 1.32 | 1.44 | 1.47 | 1.46 | 1.39 | 1.29 | 1.21 | 1.16 | 1.11 |

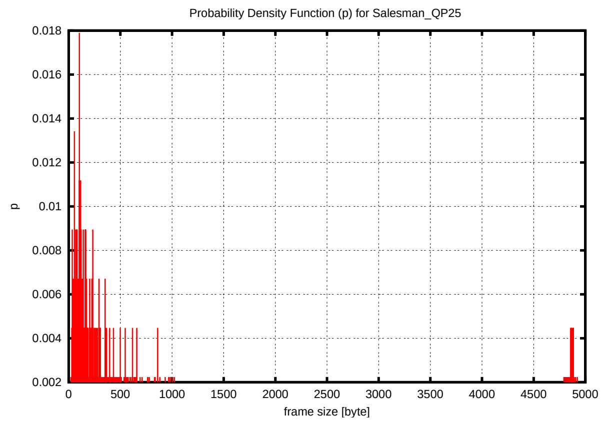


Figure 111: Frame size distribution for *Salesman (quantization 25)*

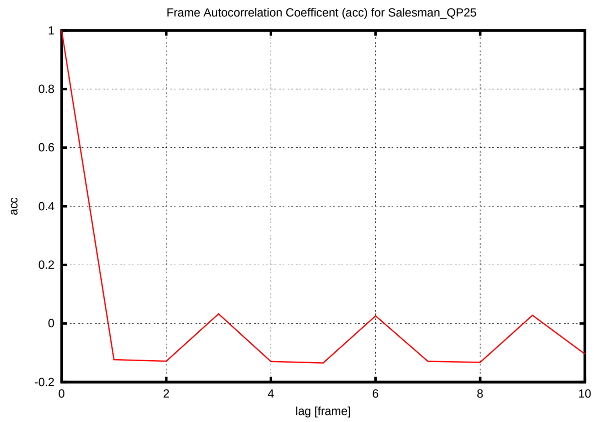


Figure 112: Autocorrelation coefficients for *Salesman (quantization 25)*

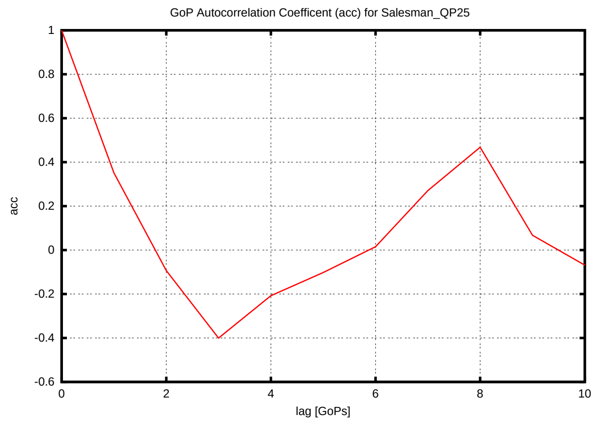


Figure 113: GoP autocorrelation coefficients for *Salesman (quantization 25)*

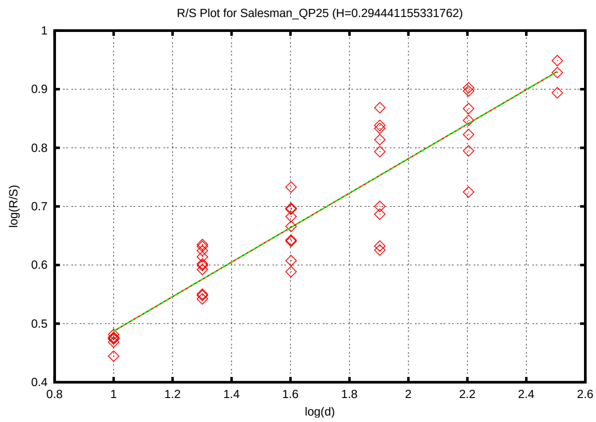


Figure 114: Single Frame R/S plot and for *Salesman (quantization 25)*

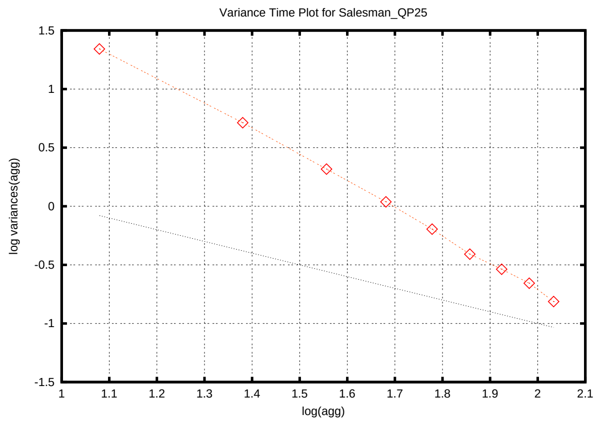


Figure 115: Variance time plot for *Salesman (quantization 25)*

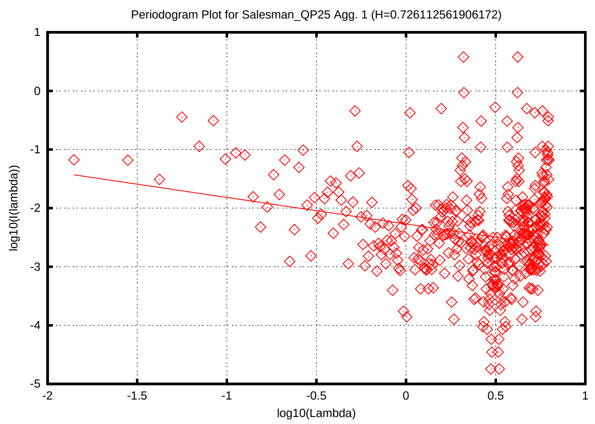


Figure 116: Single frame periodogram plot for *Salesman (quantization 25)*

# J Silent


Figure 117: Frame size trace for *Silent (quantization 25)*

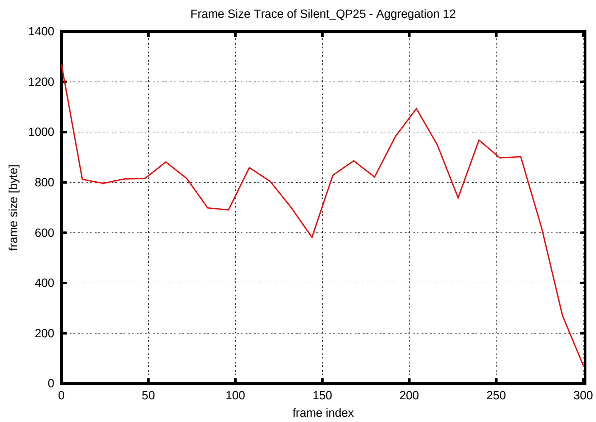


Figure 118: Frame size trace for one GoP *Silent (quantization 25)*

Table 22: Single frame statistics for *Silent*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 66 | 53 | 49 | 47 | 39 | 32 | 23 | 24 | 17 | 12 | 12 |
| $X_{max}$ [byte] | 23229 | 19695 | 15219 | 11179 | 7697 | 4889 | 2811 | 1575 | 828 | 412 | 183 |
| $\overline{X}$ [byte] | 10871.76 | 7506.24 | 3928.34 | 2266.29 | 1317.90 | 779.24 | 423.68 | 230.84 | 120.53 | 62.81 | 34.81 |
| $\overline{X}_{I-frame}$ [byte] | 21828.46 | 18427.04 | 14120.00 | 10307.19 | 7018.85 | 4476.15 | 2557.19 | 1437.92 | 772.58 | 369.04 | 165.85 |
| $\overline{X}_{P-frame}$ [byte] | 11272.43 | 7819.68 | 3899.27 | 2240.11 | 1269.60 | 728.57 | 389.32 | 208.72 | 107.13 | 52.77 | 24.33 |
| $\overline{X}_{B-frame}$ [byte] | 9297.14 | 5968.99 | 2614.33 | 1230.79 | 594.89 | 317.64 | 159.21 | 82.22 | 40.80 | 26.77 | 21.71 |
| $S_X^2$ | 14380827.52 | 13635582.97 | 11236120.97 | 6880304.36 | 3410205.98 | 1422954.66 | 471319.30 | 150086.99 | 43514.76 | 9530.54 | 1708.85 |
| $CoV$ | 0.35 | 0.49 | 0.85 | 1.16 | 1.40 | 1.53 | 1.62 | 1.68 | 1.73 | 1.55 | 1.19 |
| **Mean bitrate** [bit/s] | 2609221.79 | 1801497.41 | 942802.13 | 543908.57 | 316296.08 | 187017.41 | 101683.46 | 55401.73 | 28928.37 | 15075.35 | 8355.35 |
| **Peak bitrate** [bit/s] | 6194400.00 | 5252000.00 | 4058400.00 | 2981066.67 | 2052533.33 | 1303733.33 | 749600.00 | 420000.00 | 220800.00 | 109866.67 | 48800.00 |
| **Peak to mean** | 2.37 | 2.92 | 4.30 | 5.48 | 6.49 | 6.97 | 7.37 | 7.58 | 7.63 | 7.29 | 5.84 |

Table 23: GoP statistics for *Silent*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 9208 | 5976 | 2638 | 1196 | 600 | 347 | 192 | 118 | 89 | 97 | 73 |
| $X_{max,GoP}$ [byte] | 145961 | 104077 | 59174 | 36098 | 21447 | 12712 | 6874 | 3670 | 1869 | 944 | 500 |
| $\overline{X}_{GoP}$ [byte] | 130135.48 | 89875.92 | 47072.84 | 27177.00 | 15812.80 | 9350.92 | 5085.32 | 2770.12 | 1445.12 | 750.28 | 414.88 |
| $S^2_{X,GoP}$ | 98910894.01 | 69524304.83 | 43012443.39 | 20664140.67 | 7362071.08 | 2498743.49 | 678530.64 | 174606.69 | 37332.69 | 8437.63 | 1200.78 |
| $CoV_{GoP}$ | 0.08 | 0.09 | 0.14 | 0.17 | 0.17 | 0.17 | 0.16 | 0.15 | 0.13 | 0.12 | 0.08 |
| **Mean GoP rate** [bit/s] | 2891.90 | 1997.24 | 1046.06 | 603.93 | 351.40 | 207.80 | 113.01 | 61.56 | 32.11 | 16.67 | 9.22 |
| **Peak GoP rate** [bit/s] | 3243.58 | 2312.82 | 1314.98 | 802.18 | 476.60 | 282.49 | 152.76 | 81.56 | 41.53 | 20.98 | 11.11 |
| **Peak to mean** | 1.12 | 1.16 | 1.26 | 1.33 | 1.36 | 1.36 | 1.35 | 1.32 | 1.29 | 1.26 | 1.21 |

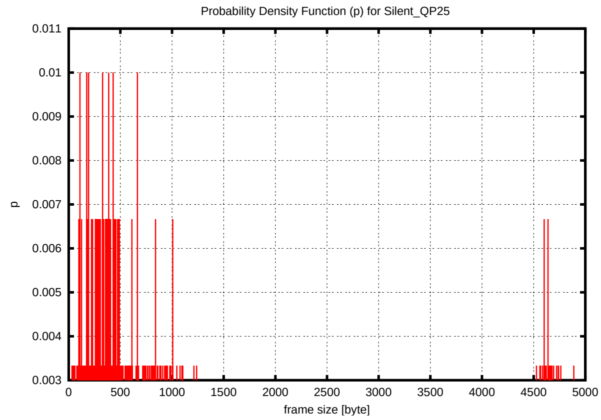


Figure 119: Frame size distribution for *Silent (quantization 25)*

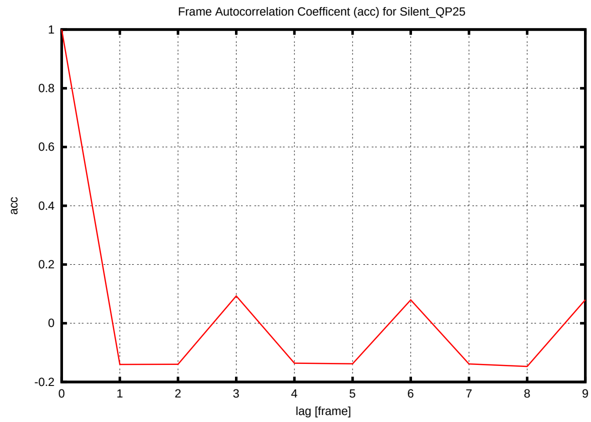


Figure 120: Autocorrelation coefficients for *Silent (quantization 25)*

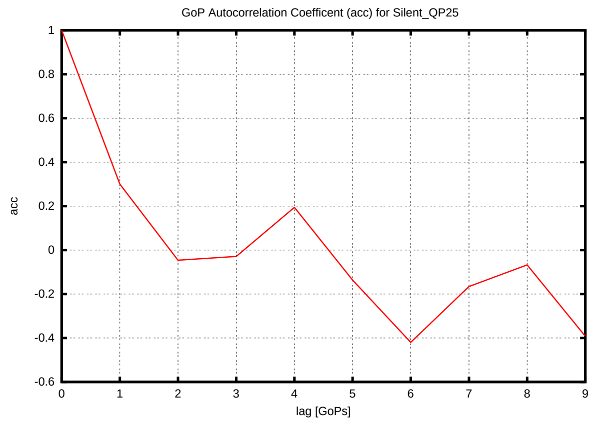


Figure 121: GoP autocorrelation coefficients for *Silent (quantization 25)*

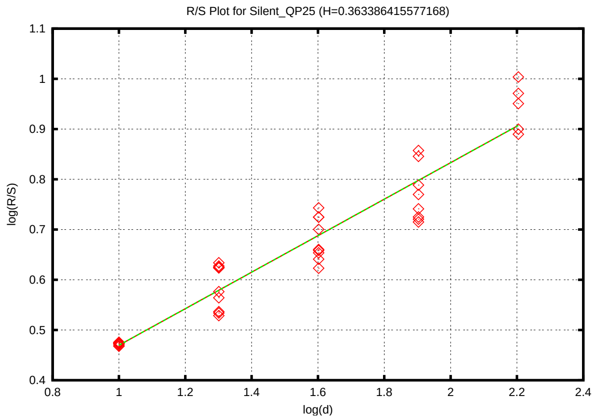


Figure 122: Single Frame R/S plot and for *Silent (quantization 25)*

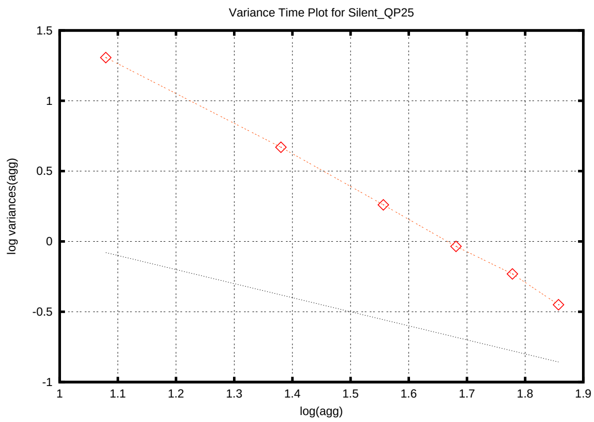


Figure 123: Variance time plot for *Silent (quantization 25)*

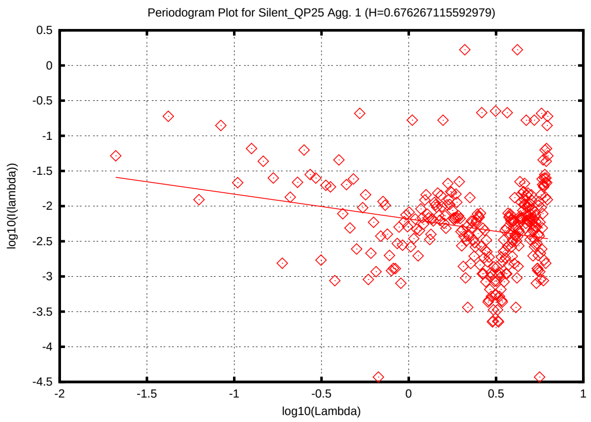


Figure 124: Single frame periodogram plot for *Silent (quantization 25)*

# K Tempete

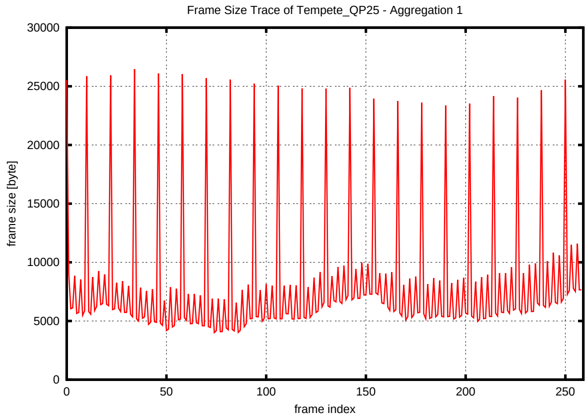


Figure 125: Frame size trace for *Tempete (quantization 25)*

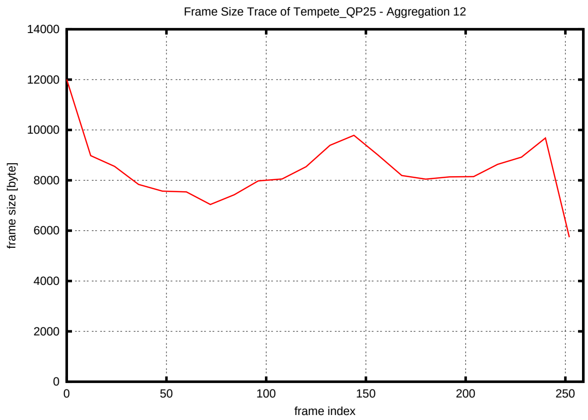


Figure 126: Frame size trace for one GoP *Tempete (quantization 25)*

Table 24: Single frame statistics for *Tempete*

| **QP** | **01** | **05** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **51** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min}$ [byte] | 66025 | 50772 | 32641 | 18576 | 9292 | 4014 | 983 | 213 | 70 | 37 | 34 |
| $X_{max}$ [byte] | 101682 | 87719 | 69159 | 52312 | 37934 | 26473 | 16875 | 10046 | 5494 | 2703 | 1128 |
| $\overline{X}$ [byte] | 74489.63 | 59063.48 | 40669.10 | 25675.72 | 14780.36 | 8029.84 | 3656.76 | 1661.11 | 806.52 | 394.49 | 177.94 |
| $\overline{X}_{I-frame}$ [byte] | 98993.18 | 85019.50 | 66432.68 | 49902.23 | 36070.18 | 24939.05 | 15815.27 | 9381.41 | 5175.09 | 2569.68 | 1069.00 |
| $\overline{X}_{P-frame}$ [byte] | 77589.52 | 61453.42 | 42704.35 | 27183.74 | 15794.35 | 8616.88 | 3961.38 | 1692.06 | 778.80 | 394.25 | 129.15 |
| $\overline{X}_{B-frame}$ [byte] | 70183.98 | 54840.35 | 36604.62 | 22007.09 | 11674.05 | 5645.19 | 1986.48 | 661.93 | 258.22 | 116.37 | 82.41 |
| $S_X^2$ | 70930887.73 | 75458278.31 | 73218070.62 | 62876688.85 | 47089339.57 | 29075289.06 | 14794354.98 | 5830631.97 | 1845986.66 | 458469.00 | 75280.63 |
| $CoV$ | 0.11 | 0.15 | 0.21 | 0.31 | 0.46 | 0.67 | 1.05 | 1.45 | 1.68 | 1.72 | 1.54 |
| **Mean bitrate** [bit/s] | 17877511.04 | 14175234.90 | 9760584.09 | 6162172.36 | 3547287.10 | 1927162.01 | 877621.62 | 398665.95 | 193564.17 | 94678.61 | 42706.10 |
| **Peak bitrate** [bit/s] | 27115200.00 | 23391733.33 | 18442400.00 | 13949866.67 | 10115733.33 | 7059466.67 | 4500000.00 | 2678933.33 | 1465066.67 | 720800.00 | 300800.00 |
| **Peak to mean** | 1.52 | 1.65 | 1.89 | 2.26 | 2.85 | 3.66 | 5.13 | 6.72 | 7.57 | 7.61 | 7.04 |

Table 25: GoP statistics for *Tempete*

| QP | 01 | 05 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_{min,GoP}$ [byte] | 100827 | 86869 | 68278 | 51257 | 36889 | 25535 | 16202 | 9555 | 5191 | 2539 | 1010 |
| $X_{max,GoP}$ [byte] | 938493 | 752182 | 530890 | 342361 | 200628 | 110792 | 51909 | 23811 | 11396 | 5549 | 2474 |
| $\overline{X}_{GoP}$ [byte] | 884489.86 | 700847.71 | 481893.24 | 303686.33 | 174449.81 | 94552.38 | 42876.48 | 19428.10 | 9447.19 | 4641.00 | 2109.71 |
| $S^2_{X,GoP}$ | 1155746930.63 | 816403714.01 | 552139035.69 | 341044172.63 | 161522634.26 | 66571933.35 | 22432126.46 | 5279694.19 | 1096240.76 | 214245.80 | 45613.91 |
| $CoV_{GoP}$ | 0.04 | 0.04 | 0.05 | 0.06 | 0.07 | 0.09 | 0.11 | 0.12 | 0.11 | 0.10 | 0.10 |
| **Mean GoP rate** [bit/s] | 19655.33 | 15574.39 | 10708.74 | 6748.59 | 3876.66 | 2101.16 | 952.81 | 431.74 | 209.94 | 103.13 | 46.88 |
| **Peak GoP rate** [bit/s] | 20855.40 | 16715.16 | 11797.56 | 7608.02 | 4458.40 | 2462.04 | 1153.53 | 529.13 | 253.24 | 123.31 | 54.98 |
| **Peak to mean** | 1.06 | 1.07 | 1.10 | 1.13 | 1.15 | 1.17 | 1.21 | 1.23 | 1.21 | 1.20 | 1.17 |

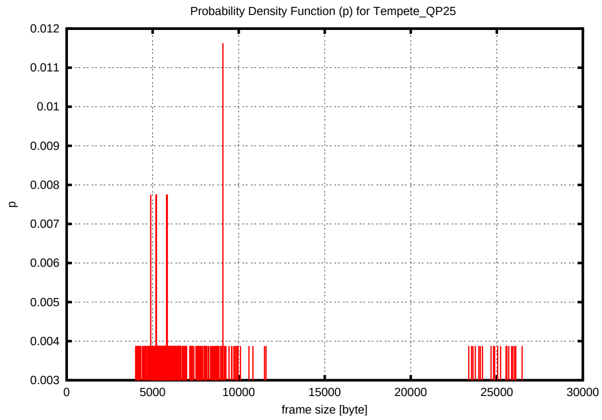


Figure 127: Frame size distribution for *Tempete (quantization 25)*

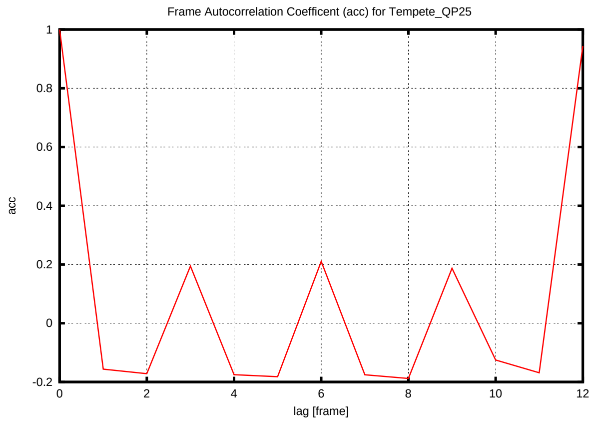


Figure 128: Autocorrelation coefficients for *Tempete (quantization 25)*

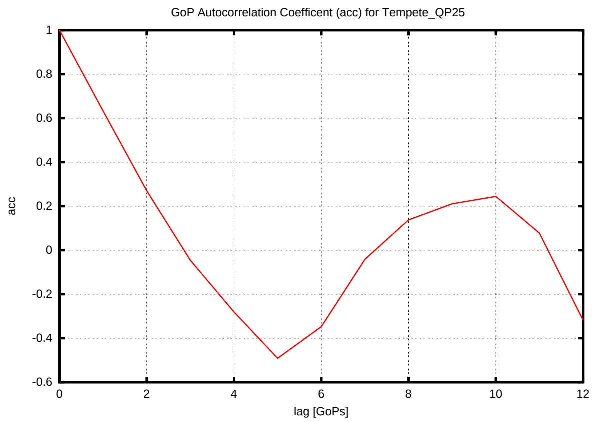


Figure 129: GoP autocorrelation coefficients for *Tempete (quantization 25)*

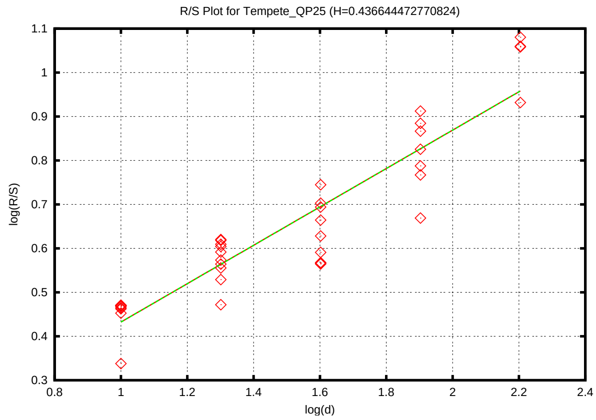


Figure 130: Single Frame R/S plot and for *Tempete (quantization 25)*

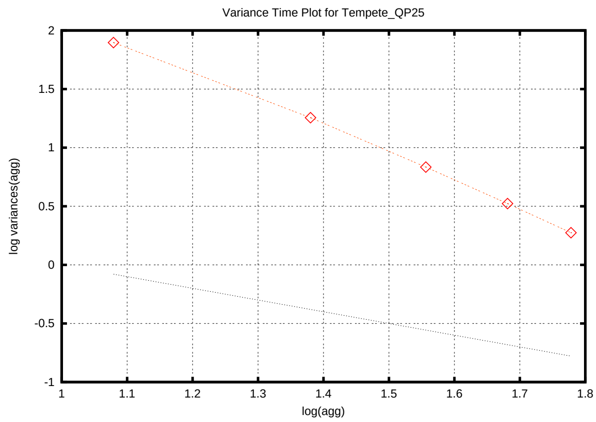


Figure 131: Variance time plot for *Tempete (quantization 25)*

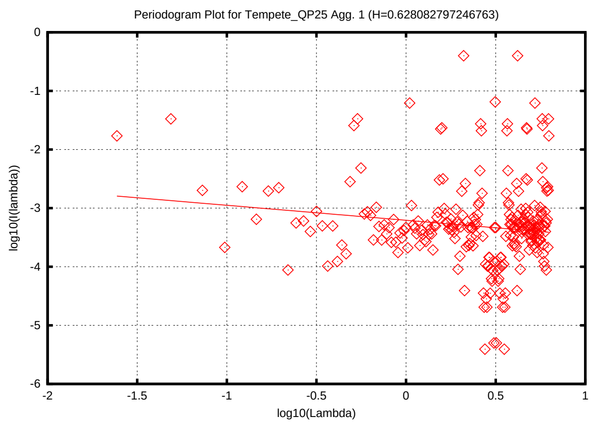


Figure 132: Single frame periodogram plot for *Tempete (quantization 25)*