Offset Trace-Based Video Quality Evaluation after Network Transport

Patrick Seeling and Martin Reisslein
Dept. of Electrical Engineering
Arizona State University
Tempe, AZ 85287-5706
Email: {patrick.seeling, reisslein}@asu.edu

Frank H.P. Fitzek
Dept. of Communication Technology
Aalborg University
Aalborg, Denmark, DK-9220
Email: ff@kom.aau.dk

***Abstract*— Video traces contain information about encoded video frames, such as frame sizes and qualities, and provide a convenient method to conduct multimedia networking research. Although widely used in networking research, these traces do not allow to determine the video quality in an accurate manner after networking transport that includes losses and delays.**

In this work, we provide (i) an overview of frame dependencies that have to be taken into consideration when working with video traces, (ii) an algorithmic approach to combine traditional video traces and offset distortion traces to determine the video quality or distortion after lossy network transport, (iii) offset distortion and quality characteristics and (iv) the offset distortion trace format and tools to create offset distortion traces.

***Index Terms*— Offset distortion, PSNR, RMSE, scalable video coding, video traces**

I. Introduction

Multimedia applications and services have attracted great popularity in usage and research activities. Large portions of the Internet traffic today are comprised of multimedia data, with video data accounting typically for the major fraction of the multimedia data transported over today's networks. With more application scenarios evolving, such as IPTV, the amount of multimedia traffic that is transported over networks in the future is bound to increase. Multimedia networking research has in turn received a great deal of attention.

For video networking research, the encoded video can be represented in several forms, such as

- The actual encoded bit stream, which typically is large in size, copyright protected and requires expertise in encoding/decoding, can not be easily exchanged among researchers.
- Video traces, which carry the information of the encoded video bit stream, but not the actual encoded information and are thus freely exchangeable among researchers.
- Video traffic models, which typically try to capture statistical properties of a certain genre of videos, are based on video traces. Additionally, models are typically limited in providing the networking researcher a model for a specific genre of video (e.g., sports videos, news videos).

Video traces thus present an appealing opportunity for networking researchers, as results can be conveniently reproduced and exchanged among researchers. At the same time, video traces are typically smaller in size than encoded video and can be used in simulation environments without much efforts. Video traces typically contain information about the encoded video frames, such as frame number and frame size, as well as the distortion or quality of the individual encoded video frames in comparison to the original and uncompressed video frames. The objective video quality is typically measured in terms of the root mean square error (RMSE) and the peak signal to noise ratio (PSNR), which is computed from the RMSE. We refer to the RMSE as *distortion* and to the PSNR as *quality*.

The information and frame loss probabilities, which are defined as the amount of data and the long run fraction of frames that miss their playout deadline at the receiver, can typically be determined in an easy fashion. These metrics, however, are not suitable to determine the video quality that is perceived at the receiving client. While video traces contain information about individual video frames, such as frame size and frame distortion or quality, this information about individual frames cannot be extended to capture the losses that occur due to lossy network transport mechanisms.

When video frames are not decodeable — either because they were not received in time or because they were damaged during network transport — the most basic and common approach is for the decoder to display the last successfully received and decoded frame until a new frame is correctly received and decoded. Video encoding mechanisms typically introduce a dependency among consecutive video frames, so that in most cases of individual frame losses, several video frames are lost for the decoding process due to inter-frame dependencies. The loss and subsequent re-display of the last successfully decoded video frame cannot be accommodated using the traditional video traces, as they do not contain

this information. *Offset distortion* video traces, on the other hand, complement the traditional video traces in providing the information needed to determine the video distortion or quality of non-decodeable video frames. The video quality as perceived at the receiving client can then be calculated by elementary statistics.

In this paper, we present offset distortion and quality video traces in combination with traditional video traces to determine the video distortion or quality after lossy network transport. In the remainder of this section, we review related works. In Section II, we review different video coding schemes, introduced video frame dependencies and decoding with re-display of the last successfully decoded video frame. We review these schemes for single layer encodings as well as temporal, spatial and SNR scalable encodings. In Section III, we present the distortion and quality metrics used for all different video traces and provide an algorithmic approach of how to use the quality and distortion values contained in traditional and offset traces to determine the video quality after network transport. In the following section, an overview of findings for the offset distortion and quality for rate-controlled and open-loop (i.e., quantization scale controlled) video encodings is given. We continue by presenting perceptual considerations and adjustments to the offset distortion or quality values in Section V. An overview of the suggested offset distortion trace format and publicly available tools is given in Section VI before we conclude in Section VII.

A. Related Works

Video traces have been used in networking research since the mid 1990s, see, e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9]. The information contained within video traces has been continuously updated as the research requirements increased. Currently, video traces typically carry information about the video frame sizes as well as video frame qualities [10].

To determine the video frame quality or video sequence quality, subjective tests or objective metrics can be used. The subjective video quality can be determined using experiments resulting in mean opinion scores (MOS) [11]. This approach, however, requires test subjects and is therefore not practical for utilization in mostly automated networking research projects. Objective metrics, on the other hand, are calculated using either the source video or the encoded video. A video quality metric that requires only the encoded the video bit stream is, e.g., the video quality metric (VQM) [12]. The RMSE and PSNR as metrics have been used continuously in video encoding and networking research to determine the video frame and video sequence quality. These two metrics compare the source video and the encoded video frame by frame to determine the distortion or quality for each video frame individually. The quality for a video stream can be determined from the individual video frame quality values using elementary statistics. Typically it is assumed that the video stream quality is maximized if the quality of individual frames is maximized and the variability of the quality among the frames of a video is minimized [13].

To quantitatively determine the impact of video frame losses on the video quality, either the video bit stream can be used, or a low quality value can be used to approximate the deteriorated quality after a frame loss [10]. Assuming a low quality value, however, is a static approach that does not take the differences in content into account. For a more thorough approach using the video bit stream, the impact of transport losses were studied for MPEG-2 in [14] and a tool was presented in [15]. In [16], the authors study the impact of differentiated QoS transport mechanisms on the video quality.

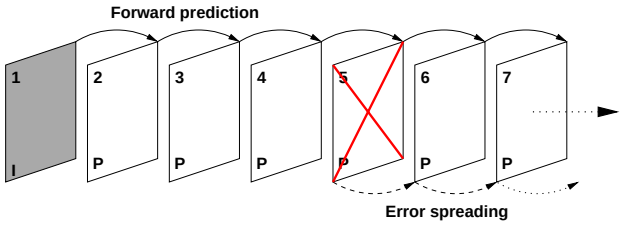

Fig. 1. Popular video coding scheme with inter-frame dependencies [17].

In [17], we introduced offset distortion traces for non-scalable video encodings. In [18], we extended the offset distortion traces for scalable video encodings. This work extends the previously conducted evaluation of the offset distortion by examining rate-controlled and non-rate-controlled offset distortions. We furthermore introduce an algorithmic view on the determination of the distortion or quality determination after potentially lossy network transport, which allows networking researchers to implement these traces into, e.g., existing interfaces for network simulators, available at [19].

II. Video Encoding and Decoding

In this section we briefly review for a video stream consisting of N frames (i) the commonly used video encoding schemes which were employed in our evaluation, (ii) the resulting inter-frame dependencies created by the encoding process and (iii) the result in terms of error spreading in case of individual frames not being available for the decoder.

A. Single Layer and Temporal Scalable Encoding

Video encoding utilizes in the most popular video coding standards the DCT transform of parts of a video frame. To increase the compression efficiency, the temporal correlation of subsequent video frames is exploited by motion estimation and motion compensation techniques. The result of applying motion estimation and compensation techniques are inter-frame dependencies. To illustrate the inter-frame dependencies created by the encoding mechanism, we consider without loss of generality a video sequence encoded with the *IPPP...* encoding pattern as illustrated in Figure 1. The *I* frames are intra-coded and rely on no other frame, whereas the forward predicted *P* frames rely on the previous I or P frames. We note that in addition to I and P frames, bi-directionally predicted *B* frames can be used as well. Frames of the B type rely on the previous **and** following I or P frames. This adds to the inter-frame dependencies and has to be taken into account when using a trace-based approach, as outlined below for temporal scalability. Without loss of generality, we assume that in case an individual frame is lost, all subsequent frames

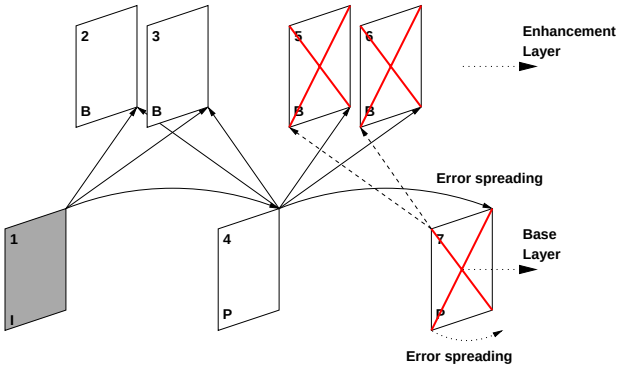

Fig. 2. Temporal scalable video with inter-frame dependencies and different error spreading possibilities [18].

that rely on the lost frames cannot be decoded. For each frame that is not available to the decoder, the decoder displays the last successfully received frame. In the example illustrated in Figure 1, we assume that the P frame 5 cannot be decoded. Subsequently, the frames 6 and 7 in the illustrated example cannot be decoded, as they rely on the availability of frame 5 at the decoder. Thus, the error from frame 5 spreads to the following frames in this example. Thus, the decoder would re-display frame 4 as replacements for frames 5, 6, and 7.

We assume without loss of generality that the error spreading from an unavailable frame at the decoder (e.g., due to transmission errors or transmission delays) spreads to subsequent frames until the decoder receives a new I frame serving as reference. This can be achieved either by following a fixed GoP structure and a limited GoP length, so that at fixed intervals a frame is encoded as an I frame, or by assuming a feedback from the decoder to the encoder to notify the encoder to encode a frame as an I frame.

Using a temporal scalability encoding mode, the B frames of the base layer (BL) of a single layer encoding constitute the enhancement layer (EL). For an example, we consider a temporal scalability scheme with an *IBBPBBPBB...* GoP pattern. With this GoP structure, the enhancement layer consists of all the B frames. As no other frames rely on the B frames in the enhancement layer, the anhancement layer can be easily added or dropped for the decoder. In the example illustrated in Figure 2, the base layer consists of I and P frames and the reception of the base layer gives a third of the original frame rate at the decoder, reception of the base and enhancement layer provides the original frame rate. The enhancement layer B frames are encoded with respect to the preceding I or P frame and the succeeding I or P frame in the base layer. As illustrated in Figure 2, the loss of a base layer (reference) frame results in the loss of the referencing frames in the enhancement layer. Simultaneously, the loss of a frame in the base layer spreads to the following frames in the base layer until a new I frame is received — either by a resynchronization request from the decoder to the encoder or by the correct reception of an I frame at the beginning at the next GoP — and the reference at the decoder has been updated. The example illustrated in Figure 2 shows how the P frame at position 7 is not available at the decoder. As the previous two B frames of the enhancement layer at positions 5 and 6 rely on the availability of the P frame at position 7, they cannot be decoded. In turn, the decoder re-displays frame 4 in place of frames 5, 6, and 7. In the same way the following frames of the base layer cannot be decoded until a new reference (I) frame of the base layer can be sent to the decoder. In turn, also the following frames of the enhancement layer cannot be decoded until the base layer has been updated with a new reference. In Algorithm 1, we provide an overview of the decoding algorithm for the single layer and temporal scalable encodings.

```
while n<N do
    if is_available(frame_{n+1}) then
        switch frame_type_{n+1} do
            case I_type
                decoded_frame_{n+1} = decode(frame_{n+1});
            case P_type
                if is_available(forward_reference(frame_{n+1})) then
                    decoded_frame_{n+1} = decode(frame_{n+1});
                else
                    decoded_frame_{n+1} = decoded_frame(frame_n);
            case B_type
                if is_available(forward_reference(frame_{n+1})) and
                is_available(backward_reference(frame_{n+1})) then
                    decoded_frame_{n+1} = decode(frame_{n+1});
                else
                    decoded_frame_{n+1} = decoded_frame(frame_n);
    else
        decoded_frame_{n+1} = decoded_frame(frame_n);
    display decoded_frame_{n+1};
    n = n + 1;
```

Algorithm 1: Decoding and display algorithm for single layer and temporal scalable video.

B. Spatial and SNR Scalable Video

Spatial scalable encoded video provides a low resolution base layer version of the encoded video for the decoder. With one or more available enhancement layers available to the decoder, the resolution of the decoded video is higher. To fix ideas here, we assume that the base layer provides a QCIF resolution and one enhancement layer in addition to the base layer provides (the original) CIF version of the encoded video. The enhancement layer frames can be encoded with respect to the corresponding base layer frames only, which we assume for our evaluations, or with respect to the corresponding base layer frame and the previous enhancement layer frame utilizing motion estimation and compensation techniques. For additional enhancement layers, the same mechanisms described here apply analogously for the additional layers. We illustrate the inter-frame dependencies for the considered spatial scalable video encoding scheme with P frames only in Figure 3. For the decoder at the receiver, three distinct cases for the decoding exist, which are

1) Neither base nor enhancement layer frames are available and the decoder displays the last frame or upsampled frame.
2) The base layer frame is available and the decoder displays an upsampled verison of the base layer frame.

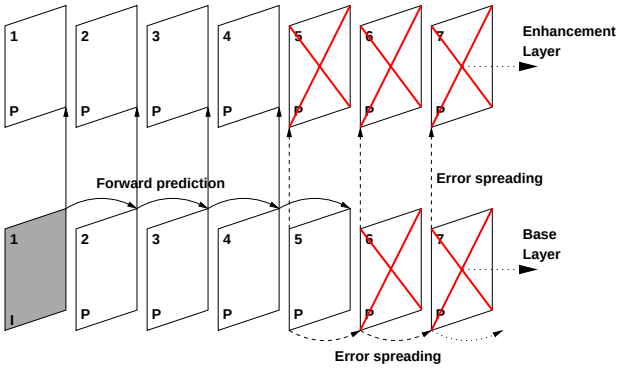

Fig. 3. Spatial scalable video with inter-frame dependencies and different error spreading possibilities [18].

3) Both, base and enhancement layer frames are available and the decoder displays the original resolution.

In the example outlined in Figure 3, the first four frames are received with their base and enhancement layer encodings. For frame 5, only the base layer is received and the decoder subsequently displays the upsampled version. For the remaining two frames 5 and 6, no data is available and the decoder re-displays the upsampled frame 5 for both frames.

SNR scalable coding provides a highly compressed and distorted version of the unencoded video frame in the base layer. Adding the enhancement layer information, additional information is available and the distortion of the decoded frame is reduced. For SNR scalable encodings, the mechanisms outlined above apply analogously. In particular, instead of using different resolutions, the decoder displays different qualities of the encoded frame. Thus, in the example provided in Figure 3, the decoder would display a high quality CIF-sized version of the video frame 1–4, a low quality version of the frame 5, and re-display frame 5 in place of frames 6 and 7.

We describe the decoder's behavior in Algorithm 2 for spatial and SNR scalable video.

```
while n<N do
    if is_available(bl_frame_{n+1}) then
        switch bl_frame_type_{n+1} do
            case I_type
                if is_available_with_references(el_frame_{n+1}) then
                    decoded_frame_{n+1} =
                    decode(bl_frame_{n+1},el_frame_{n+1});
                else
                    decoded_frame_{n+1} =
                    decode_and_upsample(bl_frame_{n+1});
            case P_type
                if is_available(forward_reference(bl_frame_{n+1})) then
                    if is_available_with_references(el_frame_{n+1})
                    then
                        decoded_frame_{n+1} =
                        decode(bl_frame_{n+1},el_frame_{n+1});
                    else
                        decoded_frame_{n+1} =
                        decode_and_upsample(bl_frame_{n+1});
                        /* In case of SNR
                           scalability, we do not
                           need to upsample        */
                else
                    decoded_frame_{n+1} = decoded_frame(frame_n);
            case B_type
                if is_available(forward_reference(bl_frame_{n+1})) and
                is_available(backward_reference(bl_frame_{n+1})) then
                    if is_available_with_references(el_frame_{n+1})
                    then
                        decoded_frame_{n+1} =
                        decode(bl_frame_{n+1},el_frame_{n+1});
                    else
                        decoded_frame_{n+1} =
                        decode_and_upsample(bl_frame_{n+1});
                        /* In case of SNR
                           scalability, we do not
                           need to upsample        */
                else
                    decoded_frame_{n+1} = decoded_frame(frame_n);
    else
        decoded_frame_{n+1} = decoded_frame(frame_n);
    display decoded_frame_{n+1};
    n = n + 1;
```

Algorithm 2: Decoding and display algorithm for spatial scalable and SNR scalable video.

III. Video Quality after Network Transport

For networking research, the loss of video data or video frames is typically determinable without much effort, either by experiments or simulation. For the determination of the video quality in an environment without losses or delays during network transport, the video quality can also be determined in a fairly easy manner utilizing video traces. The video quality is typically measured in terms of the distortion as the root mean squared error (RMSE) and quality in terms of the peak signal to noise ratio (PSNR) between original and encoded and subsequently decoded individual video frames. The unencoded video frames are typically represented in the YUV 4:2:0 format usable for MPEG-2 and MPEG-4 encodings, whereby an 8-bit value is assigned to each pixel's luminance value and an 8-bit value for a block of 4 pixels' hue and chrominance values. Typically, only the luminance component is taken into account for the video frame quality evaluations, as the human eye is most sensitive to this component [20]. Both, RMSE and PSNR, are referential video quality metrics; they both require the original video frames in addition to the decoded video frames to determine the video quality. At the same time, these metrics allow for a trace-based video quality evaluation without the actual bit stream and are easily automated.

Let us assume a video with X, Y as resolution in pixels (e.g., for QCIF $X = 144, Y = 176$ and for CIF $X = 288, Y = 352$) which consists of N video frames encoded with a quantization scale q. We denote an individual pixel's luminance value in the nth original video frame at position (x, y) as $F_n^q(x, y)$ and its encoded and subsequently decoded counterpart by $f_n^q(x, y)$. We calculate the video frame distortion as RMSE for all the luminance differences of an individual frame as

$$RMSE_n^q = \sqrt{\frac{1}{XY}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[F_n^q(x,y) - f_n^q(x,y)]^2}. \quad (1)$$

The video frame quality as PSNR can be calculated from the

RMSE as

$$Q_n^q = 20 \log_{10} \frac{255}{RMSE_n^q}. \quad (2)$$

We calculate the average video quality or video stream quality as

$$\overline{Q}^q = \frac{1}{N} \cdot \sum_{n=1}^{N} PSNR_n^q \quad (3)$$

and the variability of the video frame qualities measured as standard deviation as

$$\sigma^q = \sqrt{\frac{1}{(N-1)} \sum_{n=1}^{N} (Q_n^q - \overline{Q}^q)^2}. \quad (4)$$

To obtain a more useful variability metric taking the average video frame quality into account, we additionally calculate the coefficient of variation of the video frame qualities as

$$CoV^q = \frac{\sigma^q}{\overline{Q}^q}. \quad (5)$$

We calculate the corresponding distortion metrics in analogous manner. The video stream quality is generally maximized if the quality of individual frames is maximized and the variability of the quality among the frames of a video stream is minimized [13].

The video frame qualities of the encoded video frames are readily available for networking researchers in video traces and can be employed into research without much efforts. For lossy network transport mechanisms or by introducing delay into the delivery of the video stream, the decoder may not be able to receive video frames of the base and/or enhancement layer(s) in time or at all. Typically, no individual distortion or quality value can be assigned to these lost video frames and only a rough approximation, e.g., less than 20 dB [10], can be made. In order to facilitate networking research that includes the video frame quality after network transport, additional information is needed to determine the quality of the video frames that are not available to the decoder, either themselves or by broken references. In the following, we determine the video qualities for basic error handling schemes at the decoder using the *offset distortion* approaches outlined in [17],[18].

A. Single Layer and Temporal Scalable Video

For single and temporal scalable video, the decoder can either decode an individual video frame or not. In case the video decoder is unable to decoder a video frame, the last successfully decoded video frame is re-displayed at the client, as outlined in the examples in Section II-A. For the successfully decoded and displayed video frames, the video quality can be determined by Equation (2). For video frames that are not decoded, the offset distortion can be used to determine the video quality. The offset video distortion of the encoded and decoded re-displayed video frame with respect to the original unencoded video frame can be determined as follows. Let n denote the position of the last successfully decoded video frame and let d denote the offset of the video frame under consideration with respect to video frame n. The offset distortion is then calculated in analogy to Equation (1) as

$$RMSE_n^q(d) = \sqrt{\frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} [F_n^q(x,y) - f_{(n+d)}^q(x,y)]^2}. \quad (6)$$

The corresponding video frame quality can be calculated similar to Equation (2) as

$$Q_n^q(d) = 20 \log_{10} \frac{255}{RMSE_n^q(d)}. \quad (7)$$

We note that given this approach, the original Equations (1) and (2) for the video frame distortion and video frame quality are given as $RMSE_n^q(0)$ and $Q_n^q(0)$, respectively.

With this approach, the video quality can be determined on a per-frame basis as outlined in Algorithm 3.

```
while n<N do
    if is_available(frame_{n+d+1}) then
        switch frame_type_{n+d+1} do
            case I_type
                n = n + d + 1;
                d = 0;
            case P_type
                if is_available(forward_reference(frame_{n+d+1}))
                then
                    n = n + d + 1;
                    d = 0;
                else
                    d = d + 1;
            case B_type
                if is_available(forward_reference(frame_{n+d+1})) and
                is_available(backward_reference(frame_{n+d+1})) then
                    n = n + d + 1;
                    d = 0;
                else
                    d = d + 1;
    else
        d = d + 1;
    Q_n^q(d);
```

Algorithm 3: Calculation of the video frame quality values for single layer and temporal scalable video.

B. Spatial Scalable Video

Spatial scalable video adds upsampled video frames on top of re-displaying video frames for the decoding process as outlined in Section II-B. The upsampling introduces an additional distortion (i.e., loss of quality) to the video displayed by the decoder. Thus, additional metrics are needed in order to accommodate the upsampling of base layer video frames and the re-display of the upsampled base layer (BL) video frames.

Let $f_{UP}^{n,q}$ denote the encoded (at quantization scale q) and upsampled base layer frame n. Let furthermore $F_{EL}^{n+d,q}$ denote the original unencoded enhancement layer frame in full resolution at offset d. The distortion caused by upsampling the base layer frame n and re-displaying it instead of the full resolution combination of base and enhancement layer frames is calculated as

$$RMSE_{UP}^{n,q}(d) = \sqrt{\frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} [F_{EL}^{n+d,q}(x,y) - f_{UP}^{n,q}(x,y)]^2}. \quad (8)$$

Note that in case of $d = 0$, the distortion is calculated for the upsampling of the base layer frame only and used to determine the distortion caused by displaying the upsampled base layer frame instead of the full resolution base and enhancement layer frame combination. The corresponding video frame quality can be calculated similar to Equation (2) as

$$Q_{UP}^{n,q}(d) = 20 \log_{10} \frac{255}{RMSE_{UP}^{n,q}(d)}. \quad (9)$$

For spatial scalable encoded video, we thus need to differentiate between the base layer resolution (BL) distortion and quality, the base and enhancement layer resolution (EL) and the upsampled base layer (UP).

The combination of all the different video qualities for spatial scalable video are given in Algorithm 4.

C. SNR Scalable Video

The determination of the video frame qualities after (lossy) network transport for SNR scalable encoded video can be done in analogy to the presented metrics and algorithm for spatial scalable encoded video in Section III-B with only minor modifications. In particular, rather than determining the upsampling distortion and quality of the base layer video frame, the difference in video distortions and qualities is already given by the encoding process and/or its output.

The two metrics that are required to determine the video quality at the receiving client are the offset distortion and quality for re-displayed base layer frames at offsets d compared to the unencoded enhancement layer frames, in analogy to Equations (8) and 9.

IV. Video Offset Distortion

In this section we examine the offset distortion values in greater detail. For the encodings evaluated throughout this section, we utilize the MPEG-4 reference encoder. We note that the video offset distortion trace data is publicly available at [19].

A. Comparison of Rate Controlled and Non-Rate-Controlled Video Encoding for Single-Layer Video

The single layer video encodings we evaluate in this section are of a QCIF resolution and encoded using a *IBBPBBPBBPBBI...* GoP pattern. For encodings that are quantization scale controlled, the same quantization scale q is used for all different frame types. For encodings that are rate controlled, the TM5 rate control algorithm with the specified target bit rate is applied.

We illustrate the video frame offset distortion for frames $n = 867, 5586$, and 9089 from the video sequence *Jurassic Park I* in Figure 4 and the corresponding video frame offset quality values in Figure 5. We observe the typical inverse relationship between the distortion and quality values. Similar to the findings in [17], we observe that an approximation of the offset distortion or quality by a fixed value is not advisable, as the individual frames exhibit (i) generally different levels of distortion (and hence quality) and (ii) a different behavior of the offset distortion or quality with increasing offset d.

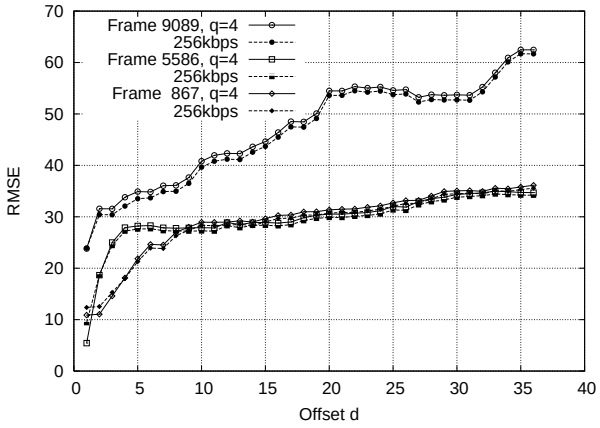

Fig. 4. Video frame offset distortion for frames $n = 867, 5586$, and 9089 from the *Jurassic Park I* video sequence encoded with a quantization scale of $q = 4$ and a target bit rate of 256kbps.

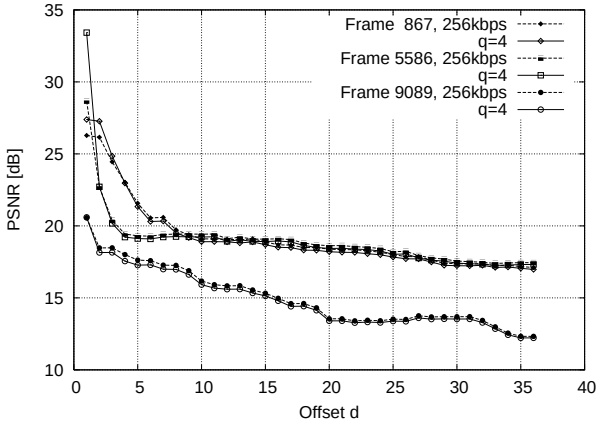

Fig. 5. Video frame offset quality for frames $n = 867, 5586$, and 9089 from the *Jurassic Park I* video sequence encoded with a quantization scale of $q = 4$ and a target bit rate of 256kbps.

Importantly, we observe that for the evaluated frames, the offset distortion values are nearly identical for the rate-controlled and non-rate-controlled versions of the encoded video. This is significant, as the non-rate-controlled version is a very high quality version, whereas the rate-controlled version is controlled by the TM5 algorithm, which dynamically adjusts the quantization scale q depending on the encoded video frame size and remaining target bandwidth for a GoP. The content of the video frames (and the content for the frames under evaluation due to the offset d) is important. Frame 867 (and the subsequent frames) shows people moving behind bushes in the dark, frame 5586 shows a reflection of a person in water and frame 9089 shows a group of people brushing off an artifact in a desert environment. We additionally observe that the relationship between the rate controlled and quantization scale controlled encodings changes as the offset d increases. For a small offset d, the quality of the quantization scale controlled encoding is higher for frames 867 and 5586, whereas with increasing offset, the quality of the rate controlled encoding for each frame is higher. Furthermore, we note that the sizes of the encoded macroblocks is different for both versions depicted in

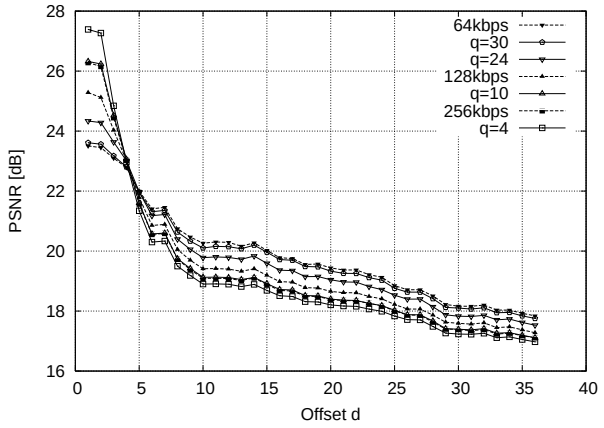

Fig. 6. Video frame offset quality for frame $n = 867$ from the *Jurassic Park I* video sequence encoded with different quantization scales q and target bit rates.

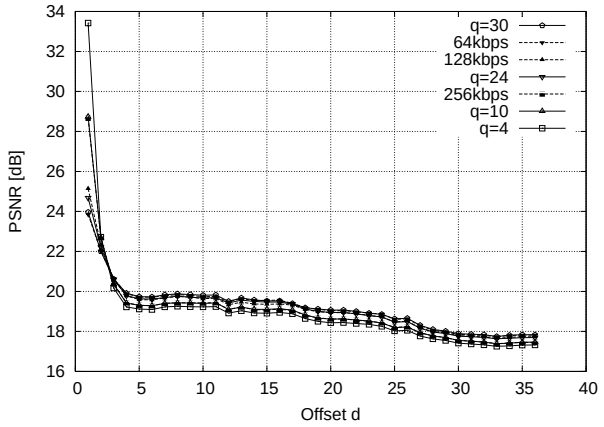

Fig. 7. Video frame offset quality for frame $n = 5586$ from the *Jurassic Park I* video sequence encoded with different quantization scales q and target bit rates.

Figures 4 and 5.

To further evaluate the different behaviors with increasing offsets and the closeness of the quantization scale controlled and rate controlled versions, we evaluate the video offset qualities for the two video frames 867 and 5586 now in greater detail. In Figures 6 and 7, we illustrate the offset quality values for both video frames encoded at different quantization scales and target bit rates. We observe that for both frames, the shape of the offset distortion curve does not change with the different levels of quantization or the different applied target bit rates. We additionally note that for both illustrated video frames, the order of the individual curves changes with increasing offset d. The frames have a higher quality for closer offsets and higher quality encodings. As the offset increases, this relationship is inversed. Two mechanisms are responsible for this behavior, namely (i) the encoding of the video frames and (ii) the change in content of subsequent video frames. For close offsets, the contents of the frames $n+d$ following the frame n under consideration are closely related and thus the differences in between the frames are small. With high quality encodings (by application of either a low quantization scale q or a high target bit rate), the small differences in content and the better quality from the encoding result in a low offset distortion (and thus high offset quality) for the higher quality encodings. For lower quality encodings, the differences are larger, as the distortions from the encoding process and not from the offset comparison are large. As the offset d increases, the differences in content of the video frames increase. With a higher quality encoding, the video frames have a higher level of detail. In turn, the differences between consecutive video frames are finer, whereas the differences in lower quality encodings are smaller due to the decreased level of detail. In turn, a higher quality encoding yields a lower offset quality (and higher offset distortion) at the larger offsets. Content dependency is also the source for the different encoding settings to be very close to another. We note that the curves for target bit rates 64kbps and quantization scale $q = 30$ are very close to another, as are the tuples of 128kbps and $q = 10$ and 256kbps and $q = 4$. This closeness is a result of the rate-distortion behavior of the encoder and the TM5 bandwidth matching. For low target bandwidths, the rate control has to assign a very high quantization scale, resulting in the first tuple of 64kbps, $q = 30$. As the target bandwidth increases, the rate control algorithm can assign lower quantization scales to the encoding, which results in the tuple 256kbps, $q = 4$. We also note that in previous findings, the TM5 algorithm was not found to be matching the lower target bit rates effectively [10]. This inability of rate control, due to a limited range of quantization scales available for the TM5 rate control algorithm, adds to the closeness of the offset qualities for the rate-controlled encodings, especially for the two lower target bit rates 64kbps and 128kbps.

To approximate unknown target bandwidths or quantization scales, it is beneficial to look at the combinations that are given by the offset quality figures and bandwidths of the encoded full video sequences. From [19], we derive the averaged bandwidths of the encoded videos as given in Table I. Comparing the different average bit rates of the encoded video, we observe that the offset qualities for encodings with similar average bit rates are very close to another in terms of their offset quality values depicted in Figures 6 and 7. For unknown quantization scales or target bit rates, the offset distortion of known quantization scales or bit rates can be used to derive a close approximation of unknown offset distortions by comparison of the average encoded bit rates.

B. Comparison of Rate Controlled and Non-Rate-Controlled Video Encoding for Scalable Video

For the evaluation of scalable video encodings, we consider spatial scalable video encodings with a QCIF base layer resolution and a CIF enhancement layer resolution. We utilize an *IBBPBBPBBPBBI...* GoP pattern in the base layer and a *PBB* structure the enhancement layer. For encodings that are quantization scale controlled, the same quantization scale q is used in the base and enhancement layers. For rate controlled encodings, the TM5 rate control algorithm is applied to the base layer with the given target bit rate while the enhancement layer is encoded using a quantization scale $q = 14$ for P and $q = 16$ for B frame types.

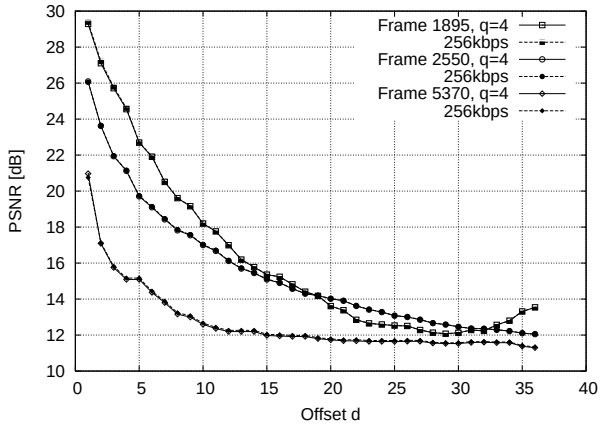

Fig. 8. Video frame offset qualities for frames $n = 1895, 2550$, and 5370 from the *Terminator I* video sequence encoded with a quantization scale of $q = 4$ and a target bit rate of 256kbps.

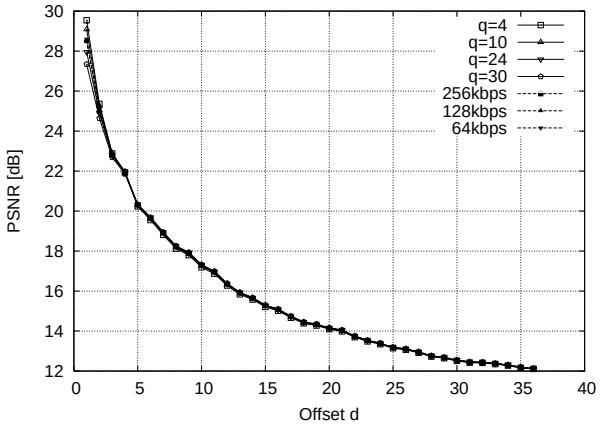

Fig. 10. Video frame offset quality for the *enhancement layer* frame $n = 2550$ from the *Terminator I* video sequence encoded with different quantization scales q and target bit rates.

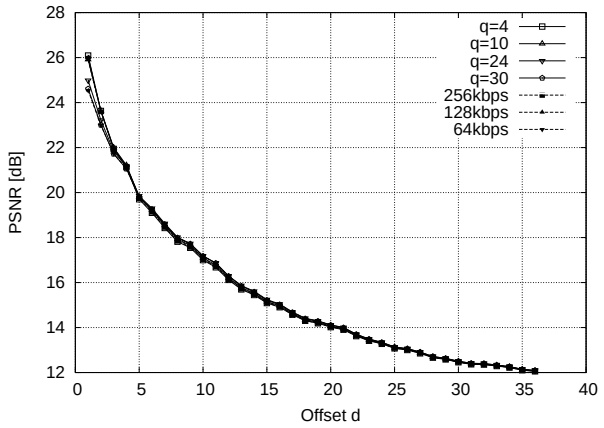

Fig. 9. Video frame offset quality for the *upsampled base layer* frame $n = 2550$ from the *Terminator I* video sequence encoded with different quantization scales q and target bit rates.

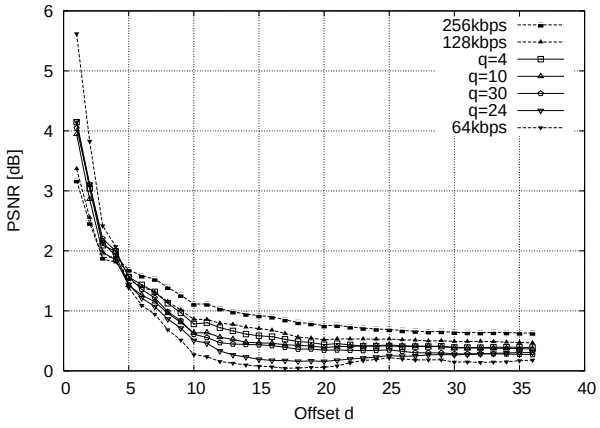

Fig. 11. Difference in the video frame offset qualities (upsampled base layer and enhancement layer) for frame $n = 2550$ from the *Terminator I* video sequence encoded with different quantization scales q and target bit rates.

We illustrate the upsampled base layer's offset qualities for frames $n = 1895, 2550$, and 5370 from the *Terminator I* video sequence, encoded with a quantization scale of $q = 4$ and a target bit rate of 256kbps, in Figure 8. We note that for the different video frames, the upsampled base layer offset qualities are very close for both quantization scale and bit rate controlled encodings and the corresponding curves are on top of each other. This can be explained as follows. Using the low resolution base layer, the rate control algorithm is able to assign a low quantization scale q to the encoded base layer, which is close to $q = 4$ illustrated in Figure 8 as well. We additionally observe that as in the single layer case, the offset quality is content dependent and thus different for each frame n and offset d.

Similar to the single layer case, we illustrate the upsampled base layer's offset quality for a variety of quantization scales q and target bit rates exemplarily for frame $n = 2550$ in Figure 9. We observe that for all different quantization scales and target bit rates, the different offset quality values are very close to another. This closeness of the upsampled base layer offset qualities can be used to approximate unknown offset qualities at other quantization scales or target bit rates.

We illustrate the offset quality for the enhancement layer frame $n = 2550$ in Figure 10. We observe that the enhancement layer offset quality closely follows the upsampled base layer offset quality values. In addition, the different enhancement layer offset quality values are close together, such that an approximation of unknown offset qualities can be done for unknown quantization scale or target bit rate settings.

We illustrate the difference in the offset qualities for the upsampled base layer and enhancement layer frame $n = 2550$ in Figure 11. For the difference between the two offset qualities, we observe a decrease with the offset d for all different encoding settings. We additionally note that for different encoding modes, the differences for the quantization scale controlled encodings are close for all offsets, whereas the differences for the rate controlled encoding modes are further apart and only seem to converge with large offsets d. Thus, for approximation of the upsampled base layer qualities by means of the qualities obtained for individual layers, the significant difference, especially for small offsets d and rate controlled encodings has to be taken into account.

V. Perceptual Considerations for Offset Distortions or Qualities

The RMSE and the RMSE-based PSNR metrics are based on the comparison between the individual video frames of the unencoded source video and the received and decoded video. These metrics therefore do not take the flow of the consecutive video frames as they are displayed at the receiver into account. This can lead to a slowing or even decrease in the offset distortion when consecutive frames have only little correlation as illustrated in Figure 7 for the offset quality of frame 5586 from the *Jurassic Park I* video sequence. This behavior, although consistent with the application of these metrics, is not consistent with the intuitive value of these video frames to the receiving client.

In order to derive a more suitable metric for comparing the source video and the received and decoded video with errors and offset distortions, it is necessary to take the impact of re-displaying the current frame multiple times on the *perceived* video quality into account. We propose an adjustment of the RMSE and PSNR metrics that takes the number of consecutive displays of a video frame into account yet does not require more information than stored in video traces. In the following, we therefore present an approximation of perceptual considerations based on the RMSE (and PSNR) values that are readily available in offset distortion video traces.

We consider a client that receives the encoded video and is presented with multiple re-displays of the same video frame. The client thus looks at a sum of distortions that originate from the last successfully decoded video frame n at $d = 0$ and the following re-displays of frame n with the offsets $d \geq 1$. We use this approach as basis to calculate the *perceptually adjusted RMSE* ($pRMSE$). In particular, we define

$$pRMSE_n^q(d) = \frac{\sum_{d=0}^{d} RMSE_n^q(d)}{d+1}, \qquad (10)$$

where the sum of distortions seen by the client is averaged over the number of re-displayed frames. The average offset distortion thus presents the upper bound for the perceptually adjusted RMSE values as the offset d increases. We determine the *perceptually adjusted quality* (pQ) for each frame and offset as

$$pQ_n^q(d) = 20 \cdot \log_{10} \frac{255}{pRMSE_n^q(d)}. \qquad (11)$$

We illustrate the traditional versus the perceptually adjusted video frame offset distortion and quality values in Figures 12 and 13 for the frames 867, 5586 and 9089 from the *Jurassic Park I* video sequence encoded with quantization scale $q = 4$ for all frame types. We observe that the perceptual adjustment results in a smooth rise of the video frame offset distortion (and a smooth decline of the video frame offset quality) for the evaluated frames and offsets.

Comparing the originally obtained video frame qualities $Q_n^4(d)$ and the perceptually adjusted video frame qualities $pQ_n^4(d)$, we observe that the perceptual quality values obtained at small offsets are generally higher than their traditionally calculated counterparts. This reflects that for a small number of re-displayed video frames, e.g., one or two frames, the perceived quality would not experience a large degradation. For frames that are further away, the perceptually adjusted quality values can be smaller than the traditionally calculated quality values. This reflects that the more video frames were not correctly received, the lower the perceived quality for the client.

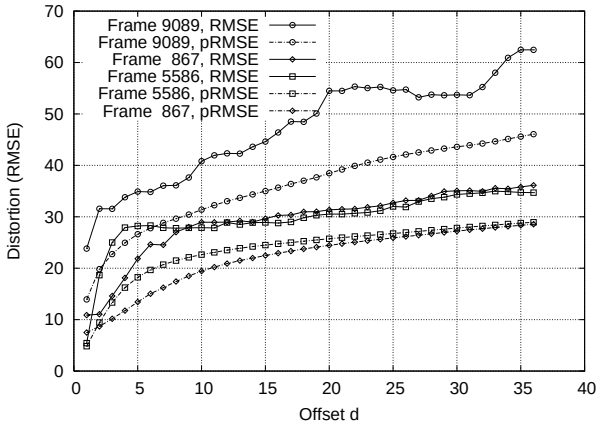

Fig. 12. Video frame offset distortion (original and perceptually adjusted) for the frames 867, 5586 and 9089 from the *Jurassic Park I* video sequence encoded with quantization scale $q = 4$.

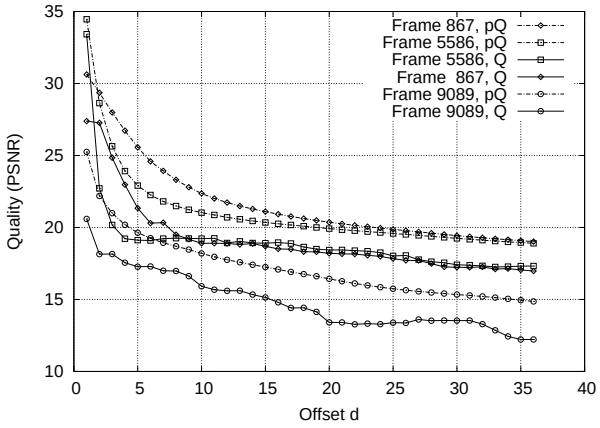

Fig. 13. Video frame offset quality (original and perceptually adjusted) for the frames 867, 5586 and 9089 from the *Jurassic Park I* video sequence encoded with quantization scale $q = 4$.

VI. Using Video Offset Distortion Traces

Traditional video traces display the frame time, frame number, frame type, video quality per frame, and additional information [10]. Offset distortion traces provide the distortion (RMSE) information per frame as a function of the offset d given in Equation (6). In order to facilitate managing the additional amount of data per frame, the offset distortion trace format has to be easily accessible. We thus propose a table format in which each frame n constitutes a line and each offset d constitutes a column in the row indexed by n. Note that for each individual encoding setup (e.g., choosing a different quantization scale q or target bit rate), a new offset distortion trace has to be generated. We illustrate an exemplary offset distortion trace as can be found at [19] in Table II for the first

TABLE II
EXEMPLARY OFFSET DISTORTION TRACE FOR STAR WARS IV, ENCODED WITH $q = 4$.

Frame #	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	...
0	79.50700667	79.62517641	80.16716058	162.5373498	122.5335215	122.3341171	...
1	79.50866246	80.05223989	162.4254677	122.4373796	122.2356938	82.56895003	...
2	80.21261549	162.5813237	122.5847470	122.3799119	82.71486860	82.83361557	...
			...				

```
while n<N do
    if is_available(bl_frame_{n+d+1}) then
        switch bl_frame_type_{n+d+1} do
            case I_type
                n = n + d + 1;
                d = 0;
                if is_available_with_references(el_frame_{n+d+1}) then
                    Q_EL^{n,q}(d);
                else
                    Q_UP^{n,q}(d);
            case P_type
                if is_available(forward_reference(bl_frame_{n+d+1}))
                then
                    n = n + d + 1;
                    d = 0;
                    if is_available_with_references(el_frame_{n+d+1})
                    then
                        Q_EL^{n,q}(d);
                    else
                        Q_UP^{n,q}(d);
                else
                    d = d + 1;
                    if last_good_frame_was_el then
                        Q_EL^{n,q}(d);
                    else
                        Q_UP^{n,q}(d);
            case B_type
                if is_available(forward_reference(bl_frame_{n+d+1}))
                and
                is_available(backward_reference(bl_frame_{n+d+1}))
                then
                    n = n + d + 1;
                    d = 0;
                    if is_available_with_references(el_frame_{n+d+1})
                    then
                        Q_EL^{n,q}(d);
                    else
                        Q_UP^{n,q}(d);
                else
                    d = d + 1;
                    if last_good_frame_was_el then
                        Q_EL^{n,q}(d);
                    else
                        Q_UP^{n,q}(d);
    else
        d = d + 1;
        if last_good_frame_was_el then
            Q_EL^{n,q}(d);
        else
            Q_UP^{n,q}(d);
```

Algorithm 4: Calculation of the video frame quality values for spatial scalable video.

TABLE I
ENCODED BANDWIDTHS FOR DIFFERENT QUANTIZATION SCALES q AND TARGET BIT RATES FOR THE VIDEO SEQUENCE *Jurassic Park I* FROM [19].

Quantization scale q / Target bit rate	Average Bitrate [Mbps]
$q = 4$	0.78435
$q = 10$	0.22787
$q = 24$	0.07845
$q = 30$	0.06645
64kbps	0.06632
128kbps	0.12813
256kbps	0.25613

3 video frames and values of d up to 6. We note that the trace format is the same for all different offset distortions calculated and presented in this contribution and that given the relation between the offset distortion and offset quality as given in Equation (7), only the offset distortion values are included in the traces.

A. Assessing the Video Quality After Network Transport Using Video Traces

To determine the video quality after potentially lossy network transport, both the conventional traces and the offset distortion traces are required. In Algorithm 5, we provide the function to determine the video frame quality values for the algorithms presented in Section III and the traces available at [19]. In particular, we consider the terse video traces from [19], as they contain the information for frame n in row n (they are in frame order), whereas the verbose traces are using an index for each frame (they are in encoding order). Both trace formats can be used, but the difference in order and the requirement for additional indexing when using the verbose video traces have to be kept in mind.

B. Available Tools

Video offset distortion traces have become available to the networking research community for several different video encodings, see [19]. The availability of offset distortion traces alone may not be sufficient for researchers that have access to the unencoded video bit streams or for video researchers that are interested in working with the commonly used video test sequences. For those researchers, we made our software tools used in the generation of the offset distortion traces available for download at [19]. The tools generate the suggested output in a comma separated (CSV) format that allows for easy access

```
switch d do
    case d = 0
        /* Traditional traces already provide the
           video quality values.                    */
        if single_layer then
            Q_n^q(0) = terse_trace(row n);
        else
            if base_and_enhancement_layer then
                Q_EL^{n,q}(0) = terse_aggregated_trace(row n);
            else
                Q_UP^{n,q}(0) = terse_trace_for_BL(row n);
    case d ≥ 1
        /* Offset distortion traces provide the
           video distortion values.                 */
        switch video_encoding_mode do
            case single_layer
                Q_n^q(d) = 20 · log10 255 / offset_trace(row n, col. d);
            case temporal_scalable
                Q_n^q(d) = 20 · log10 255 / offset_trace(row n, col. d);
                /* For temporal scalable encodings,
                   the single layer offset traces
                   can be used.                     */
            case spatial_scalable
                if base_and_enhancement_layer then
                    Q_n^q(d) =
                    20 · log10 255 / offset_trace_EL(row n, col. d);
                else
                    Q_UP^{n,q}(0) = 20 ·
                    log10 255 / upsampled_BL_offset_trace(row n, col. d);
```

Algorithm 5: Function to derive the video frame quality values for single and scalable video encodings after network transport from video traces at [19].

to the values in spreadsheet programs and further processing. The tools require the input video to be in the YUV 4:2:0 format.

1) Offset Distortion Calculator: The offset distortion calculator can be used to calculate the offset distortion values for the single layer, temporal scalable, and SNR scalable encodings. The program supports QCIF and CIF video frame sizes. We illustrate the program window for the offset distoriton calculator in Figure 14. The program requires the unencoded original video (Org.) and the encoded and subsequently decoded (Enc.) video sequences. The largest offset d can be changed to suit different needs.

2) Spatial Scalable Offset Distortion Calculator: The spatial scalable offset distortion calculator can be used to calculate the offset distortion values for the upsampled and re-displayed base layer frames in spatial scalable encodings. Currently, the program supports a QCIF frame size for the base layer and a CIF frame size of the enhancement layer. The base layer frames are upsampled without further processing, e.g., no filters are used to smooth out imperfections. The program requires the unencoded original enhancement layer resolution video (Org. EL) and the encoded and subsequently decoded base layer resolution (Enc. BL) video sequences. The largest offset d can be changed to suit different needs as for the offset distortion calculator.

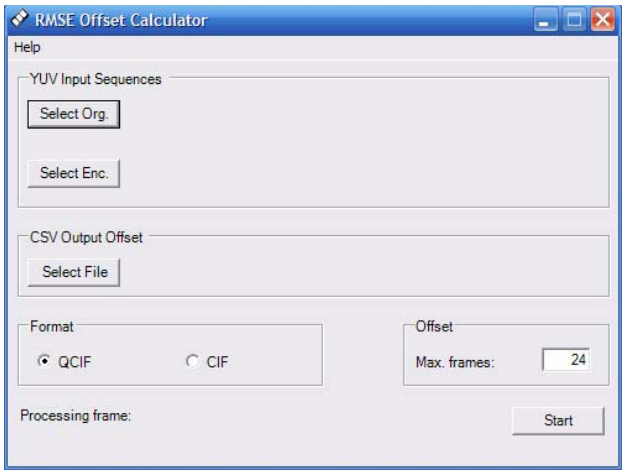

Fig. 14. Screenshot of the video offset distortion calculator for individual layers, currently QCIF and CIF video formats are supported.

VII. Conclusion

In this contribution, we reviewed the offset distortion concept and how offset distortion traces can be used in conjunction with traditional video traces to facilitate networking research with evaluation of the video quality after network transport. We examined the offset distortion and quality behaviors of single layer and scalable encoded video and compared quantization scale controlled and rate-controlled offset qualities.

We gave algorithmic approaches for networking researchers on how the decoder behavior can be approximated for simulation purposes and how the different video frame and offset qualities are calculated. We showed how using the combination of both video traces, frame losses and delays during the network transport can be accommodated.

We introduced our suggested format for the offset distortion traces and introduced available tools to create offset distortion traces for single layer, temporal scalable, spatial scalable, and SNR scalable video encodings.

Acknowledgment

Supported in part by the National Science Foundation under Grant No. ANI-0136774.

REFERENCES

[1] M. Dai and D. Loguinov, "Analysis and modeling of MPEG-4 and H.264 multi-layer video traffic," in *Proceedings of IEEE INFOCOM*, Miami, FL, Mar. 2005.

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

[3] P. Koutsakis and M. Paterakis, "Call-admission-control and traffic-policing mechanisms for the transmission of videoconference traffic from MPEG-4 and H.263 video coders in wireless ATM networks," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1525–1530, Sept. 2004.

[4] M. Krunz and S. Tripathi, "Exploiting the temporal structure of MPEG video for the reduction of bandwidth requirements," in *Proc. of IEEE Infocom*, vol. 1, no. 1, Kobe, Japan, Apr. 1997, pp. 67–74.

[5] M. Krunz, R. Sass, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," in *Proceedings of IEEE INFOCOM*, Boston, MA, Apr. 1995, pp. 455–462.

[6] J. Liebeherr and D. Wrege, "Traffic characterization algorithms for VBR video in multimedia networks," *Multimedia Systems*, vol. 6, no. 4, pp. 271–283, July 1998.

[7] J. W. Roberts, "Internet traffic, QoS, and pricing," *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1389–1399, Sept. 2004.

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

[9] U. Sarkar, S. Ramakrishnan, and D. Sarkar, "Study of long-duration MPEG-trace segmentation methods for developing frame-size-based traffic models," *Computer Networks*, vol. 44, no. 2, pp. 177–188, Feb. 2004.

[10] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation with frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 3, pp. 58–78, Third Quarter 2004.

[11] ITU-T Recommendation P.800.1, "Mean opinion score (MOS) terminology," Mar. 2003.

[12] M. Pinson and S. Wolf, "Low bandwidth reduced reference video quality monitoring system," in *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, AZ, Jan. 2005.

[13] T. Lakshman, A. Ortega, and A. Reibman, "VBR video: Tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[14] W. Luo and M. ElZarki, "Analysis of error concealment schemes for MPEG–2 video transmission over ATM based networks," in *Proceedings of SPIE Visual Communications and Image Processing 1995*, Taiwan, May 1995, pp. 102–108.

[15] ——, "MPEG2Tool: A toolkit for the study of MPEG–2 video transmission over ATM based networks," Department of Electrical Engineering, University of Pennsylvania, Tech. Rep., 1996.

[16] J. Shin, J. W. Kim, and C.-C. J. Kuo, "Quality–of–service mapping mechanism for packet video in differentiated services network," *IEEE Transactions on Multimedia*, vol. 3, no. 2, pp. 219–231, June 2001.

[17] P. Seeling, M. Reisslein, and F. Fitzek, "Offset distortion traces for trace-based evaluation of video quality after network transport," in *Proc. of IEEE Int. Conference on Computer Communications and Networks (ICCCN)*, San Diego, CA, Oct. 2005, pp. 375–380.

[18] ——, "Layered video coding offset distortion traces for trace-based evaluation of video quality after network transport," in *Proc. of IEEE Consumer Communications and Networking Conference CCNC*, Las Vegas, NV, Jan. 2006, pp. 292–296.

[19] "Video traces for network performance evaluation," Traces available at: http://trace.eas.asu.edu.

[20] S. Winkler, "Vision models and quality metrics for image processing applications," Ph.D. dissertation, EPFL, Switzerland, 2000.

Patrick Seeling is a Faculty Research Associate in the Department of Electrical Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. degree in industrial engineering and management (specializing in electrical engineering) from the Technical University Berlin (TUB), Germany, in 2002. He received his Ph.D. in electrical engineering from Arizona State University (ASU), Tempe, in 2005. He has served on the Technical Program Committee of the *IEEE Consumer Communications and Networking Conference (CCNC)* and in the Program Committee of the *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE)*. He is co–recipient of the Best Paper Award at the 2006 *IEEE Consumer Communications and Networking Conference (CCNC)*. His main research interests are in the areas of video transmission over wired and wireless networks, multimedia characteristics, wireless networking and engineering education.

Martin Reisslein is an Associate Professor in the Department of Electrical Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994–1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin and lecturer at the Technical University Berlin. From October 2000 through August 2005 he was an Assistant Professor at ASU. He is editor–in–chief of the *IEEE Communications Surveys and Tutorials* and has served on the Technical Program Committees of *IEEE Infocom*, *IEEE Globecom*, and the *IEEE International Symposium on Computer and Communications*. He has organized sessions at the *IEEE Computer Communications Workshop (CCW)*. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG–4 and H.263 encoded video, at `http://trace.eas.asu.edu`. He is co–recipient of Best Paper Awards at the *SPIE Photonics East 2000 — Terabit Optical Networking Confrence* and the 2006 *IEEE Consumer Communications and Networking Conference (CCNC)*. His research interests are in the areas of Internet Quality of Service, video traffic characterization, wireless networking, optical networking, and engineering education.

Frank H.P. Fitzek is an Associate Professor in the Department of Communication Technology, University of Aalborg, Denmark heading the Future Vision group. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology - Rheinisch-Westfaelische Technische Hochschule (RWTH) - Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 for quality of service support in wireless CDMA networks. As a visiting student at the Arizona State University he conducted research in the field of video services over wireless networks. He co-founded the start-up company acticom GmbH in Berlin in 1999. In 2002 he was Adjunct Professor at the University of Ferrara, Italy giving lectures on wireless communications and conducting research on multi-hop networks. In 2005 he won the YRP award for the work on MIMO MDC and in 2005 he received the Young Elite Researcher Award of Denmark. His current research interests are in the areas of 4G wireless communication networks, cross layer protocol design and cooperative networking.