Error concealment analysis for H.264/advanced video coding encoded video sequences

L. Superiori, O. Nemethova Member IEEE, M. Rupp Senior Member IEEE, Member EURASIP

Error concealment methods have become very important in particular when transmitting video streams over error prone wireless links. Often a retransmission of corrupted sequences is not possible and thus the receiver has to make the best out of the received stream. The contributions of this article are the following: firstly, a performance comparison of various error concealment strategies (straight decoding, slice level concealment and macroblock level concealment) is presented based on the detection of errors, the exact location of which is unknown. Secondly, an analytical treatment of the slice level concealment, resulting in a precise mathematical model is provided. Finally, further improvements are proposed by subjective methods based on visual inspection and comparison of their performance by means of simulations.

Keywords: error concealment; error detection; syntax check; subjective quality improvement

Fehlerverschleierungsanalyse in H264/Advanved Video Coding-codierten Videosequenzen.

Fehlerverschleierungsverfahren sind zurzeit von großem Interesse, insbesondere bei der Übertragung von Videos über fehlerbehaftete Funkstrecken. Meist ist eine Zweitübertragung einer fehlerbehafteten Sequenz nicht möglich, und der Empfänger muss das Beste aus den Empfängerdaten machen. In diesem Artikel präsentieren wir zunächst eine vergleichende Leistungsanalyse verschiedener Verschleierungsverfahren (direktes Dekodieren, Verschleierung auf Slice-Ebene und Verschleierung auf Makro-Ebene), basierend auf einer Fehlererkennung, dessen exakte Fehlerposition in der Sequenz allerdings unbekannt ist. Zweitens präsentieren wir eine analytische Behandlung der Verschleierung auf Slice-Ebene, die zu einem exakten mathematischen Modell führt. Schließlich schlagen wir weitere Verbesserungeen basierend auf subjektiven Verfahren der visuellen Inspektion vor und vergleichen ihre Leistungsfähigkeit anhand von Simulationen.

Schlüsselwörter: Fehlerverschleierung; Fehlererkennung; Syntaxüberprüfung; subjektive Qualitätsverbesserung

Received March 12, 2012, accepted July 23, 2012

Abbreviations and symbols

AVC	Advanced Video Coding
BER	Bit Error Ratio
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
ecdf	empirical cumulative distribution function
FTP	File Transport Protocol
GOB	Groups Of Blocks
GOP	Group Of Picture
IDR	Instantaneous Decoding Refresh
IP	Internet Protocol
JM	Joint Model
JVT	Joint Video Team
LS	Least Squares
MB	MacroBlock
MBLC	MacroBlock Level Concealment
MSE	Mean Square Error
MTU	Maximum Transfer Unit
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
PSS	Packet Switched Streaming
PSS	Packet Switched Services
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
RTP	Real Time Protocol
SD	Straight Decoding
SLC	Slice Level Concealment
TCP	Transport Control Protocol
UDP	Universal Datagram Protocol
VCL	Video Coding Layer
Y-PSNR	Luminance Peak Signal to Noise Ratio

1. Introduction

The transmissions of data over wireless channels are affected by errors due to multiple causes, such as poor channel quality, interference and noise at the receiver. For some services, such as In-

This work has been funded by A1 Telekom Austria AG. Parts of this work (Section V) were published in a conference version L. Superiori and O. Nemethova and M. Rupp, "Detection of Visual Impairments in the Visual Domain," Picture Coding Symposium Proceedings, Lissabon, Portugal, Nov. 2007 (*Superiori et al. 2007a*).

Superiori, Luca, Dr., M.Sc., Institute of Telecommunications, Vienna University of Technology, Gusshausstraße 25/e389, 1040 Vienna, Austria (E-mail: luca.superiori@nt.tuwien.ac.at); **Nemethova, Olivia, Dr., M.Sc.,** Institute of Telecommunications, Vienna University of Technology, Gusshausstraße 25/e389, 1040 Vienna, Austria (E-mail: olivia@ieee.org); **Rupp, Markus, Univ.-Prof. Dipl.-Ing. Dr.,** Institute of Telecommunications, Vienna University of Technology, Gusshausstraße 25/e389, 1040 Vienna, Austria (E-mail: mrupp@nt.tuwien.ac.at)

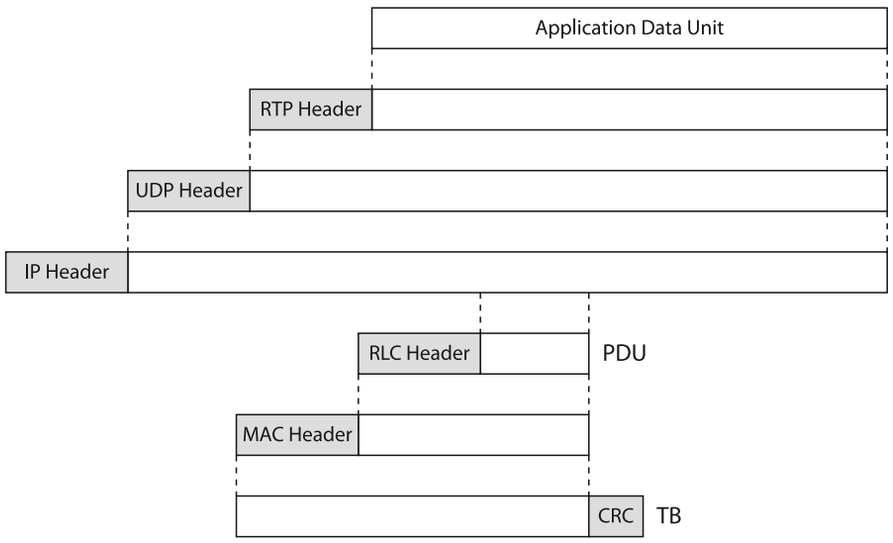

Fig. 1. Packet Switched Streaming (PSS) protocol stack

ternet browsing or File Transport Protocol (FTP) traffic, the application requires the retransmission of incorrectly received packets. For video streaming applications, however, missing and damaged packets have to be handled differently. In broadcast and multicast transmissions, a retransmission of damaged or not received packets is not possible. A single source is serving several users and, even assuming the presence of a feedback channel, it cannot react to the needs of a single user since the data channel is typically shared by the plurality of users. Although possible in unicast transmissions, retransmission can also be problematic there. In case of network congestion, an overloaded cell may not be able to retransmit the damaged packet within the delay limit required by the dimension of a playout buffer (typically 3–6 s) (*Sabeva et al. 2006; Levine et al. 2007*).

The video decoder has to cope with missing or damaged packets as well as with packets arriving too late (*Stockhammer et al. 2003*). The application reacts by *concealing* the missing information. Concealing an error basically means replacing the missing or invalid data with other information extrapolated from the already available video content. Video concealment techniques can be based on spatial, temporal or hybrid interpolation. See for example (*Sun and Reibman 2003; Nemethova 2009*) for good overviews of existing techniques. In the spatial case, the missing information is concealed by interpolating from the available neighbouring image portions. In the temporal concealment a missing part of the image is interpolated from other already received images. Motion compensation may be applied for selecting an appropriate replacement. Hybrid concealment methods consist of a mixture of the two approaches, spatial and temporal, depending on the frame type as well as on the picture properties. Often adaptive methods (*Nemethova et al. 2006*) can select the best means against particular error situations.

H.264 (*2005*) (known also as Advanced Video Coding (AVC)) and MPEG-4 part 10, standardized by the Joint Video Team (JVT) of experts from both ISO/IEC (International Electrotechnical Commission) and ITU (International Telecommunication Union) is currently the most recent video standard. Thus all our experiments utilize an H.264/AVC codec. The H.264/AVC design covers a Video Coding Layer (VCL), designed to efficiently represent the video content, and the Network Abstraction Layer (NAL), which formats the VCL representation of the video data and provides header information in a manner supportable by a variety of different communication protocol architectures or storage media (*Marpe et al. 2006*). The output of the NAL is the Network Abstraction Layer Unit (NALU).

In this paper we discuss the exploitation of the valid information contained in damaged packets. Since the exact location of the error is typically not known, the entire area detected as erroneous is typically concealed by means of interpolation. However, in general the larger the image portion to be interpolated, the higher the distortion of the concealed image. Our aim is to avoid concealing of the image parts which can still be recovered by an appropriate decoding. After briefly presenting the effects of errors at frame level and at packet level, in Sects. 2.1 and 2.2, respectively, a syntax based error detection mechanism is briefly explained in Sect. 3. In Sect. 4 eventually three error concealment strategies are compared. The proposed strategy, called slice level concealment has been analyzed in detail, achieving a highly reliable estimation of the observed results. Finally, in Sect. 5 we elaborate on recent methods for visual impairment detection capable of even more increasing the performance of the suggested error concealment strategies. Some conclusions round up the paper in Sect. 6.

2. Error impact on the video stream

The packetized video stream is transported over a network with a layered protocol structure, an example of which is shown in Fig. 1.

The application layer payload (in H.264/AVC called NALU) is a packet produced by an application such as a video and/or audio encoder for multimedia streaming. The application payload is further encapsulated into the Real Time Protocol (RTP) (*Group et al. 1996; Group and Schulzrinne 1996; Taferner and Bonek 2002; Wenger 2003; Wenger et al. 2005*). The RTP standardizes a packet format for the end-to-end delivering of audio and video through the Internet. The RTP is considered as a session oriented *unreliable* protocol, since it does not guarantee the delivery of the packet nor offer any mechanism for recovering packet losses. In the RTP headers (12 bytes), the sequence of the packets is marked by means of a unique *sequence number* and a *timestamp*, also useful to synchronize different sources such as video and audio. This allows the application to sort out the packet sequence, remove duplicated packets and, possibly, to react to missing packets or to packets arriving too late.

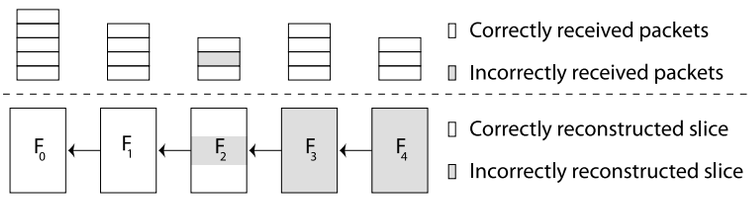

Fig. 2. Temporal error propagation

A *(de-)jitter buffer* is typically implemented at the application layer to fight the jitter, that is the variation of the packets arrival time. The size of the jitter buffer has to be selected carefully, since a small jitter buffer increases the packet loss rate while a large jitter buffer adds unnecessary delay.

The Universal Datagram Protocol (UDP) (*Postel 1980*) and the Transport Control Protocol (TCP) (*Postel 1981*) are the two transport layer protocols commonly used in the Internet. The TCP offers guaranteed transport services based on the retransmission of missing or damaged packets. As this causes unpredictable transport delays, TCP is not used in real-time communications. UDP, on the other hand, is a simple and unreliable datagram transport protocol. In the UDP header (8 bytes), the source and destination port are specified, as well as the length of the packet. Moreover, a sixteen bit long checksum is provided for verifying the correctness of the header and of the payload. The checksum enables detecting the presence of an error within the UDP packet. If an error is detected, its presence is signaled to higher layers, for instance, by setting a flag reserved for such a purpose. Typically, the packets signaled as erroneous are discarded by the higher layers. The missing image information is concealed by interpolating the corresponding image portion from other already decoded image data.

2.1 Effects of errors at frame and sequence level

The quality of the error concealment depends substantially on the size and position of the missing image information. In general, a video sequence comprises a plurality of images (called frames), each of which is typically subdivided into blocks. The image is then encoded block-wisely. A UDP packet over which the checksum is calculated typically carries a bitstream corresponding to the entire encoded image or to its part (called slice). One image slice contains a plurality of square regions, called MacroBlock (MB). For increasing the effectiveness of the coding, one macroblock can be subdivided further in submacroblocks. The encoding of image blocks may be based on spatial prediction, known as Intra (*I*) as it uses as reference elements belonging to the same image, or temporal prediction, known as Inter (*P*) as it uses as reference elements belonging to a previously encoded picture. A sequence of images consisting of one spatially predicted followed by a plurality of temporally predicted images up to the next image which is spatially predicted or not predicted, is called a Group Of Picture (GOP).

Thus, if a UDP packet is detected as erroneous, the corresponding missing image or slice of an image has to be somehow replaced. An estimation of the missing data is performed in order to display the video content with as little as possible impairment of the user experience. However, even if the reconstruction at the time instant t of the missing part of the picture may not be annoying for the users, the concealed picture is further used as a reference for prediction of the following temporally predicted frames. Since the prediction is computed at the encoder side (considering the error-free sequence as reference), if a concealed frame is used as a reference frame at the decoder side, a drift between encoder and decoder is introduced. This effect is illustrated in Fig. 2.

As a packet including a part of frame F_2 has been incorrectly received, the corresponding slice of the frame F_2 is incorrectly reconstructed or concealed. The frame F_3, although all its packets have been correctly received, is employing a corrupted reference for performing temporal prediction. This effect, denoted as *temporal error propagation*, involves all the following Inter encoded pictures until the end of the GOP. Only the next spatially predicted frame does not contain a reference to the previous pictures. If an Inter-GOP prediction is allowed, the insertion of Intra frames is not sufficient to stop the temporal error propagation. If the reference buffer size is larger than one, the first inter predicted frame of one GOP may use as reference a frame, belonging to the previous GOP. The temporal error propagation is then interrupted only in case the reference buffer contains only one picture or if the *I* frame empties the content of the buffer. In this last case the *I* frame is called Instantaneous Decoding Refresh (IDR) frame.

2.2 Effects of errors at packet level

Even though the UDP packet fails the checksum test, it can be still conveyed further to the RTP layer and the NALU can be handed as input to the video decoder. In this case, the Forbidden (F) bit of the NALU header is set to one for informing the decoder that it has possibly to cope with a damaged payload.

We will now discuss the effect of errors in the bitstream. As a reference bitstream, the bitstream generated by the Joint Model (JM) in accordance with the H.264/AVC standard has been considered. H.264/AVC supports two types of variable length (entropy) coding—Context Adaptive Binary Arithmetic Coding (CABAC) and Context Adaptive Variable Length Coding (CAVLC). The latter includes, in fact, various kinds of entropy coding applied to different information elements. Some information elements are encoded by an exp-Golomb code, or a similar integer code whereas other elements are coded by a code specified by particular codeword table(s) (in the following, this kind of coding will be denoted *tabelled VLC*). Depending on the entropy coding applied and on the affected syntax element, different effects arise. In the following, the case of codewords with variable length is analyzed based on exp-Golomb encoded codewords and subsequently generalized. The structure of an exp-Golomb encoded codeword has the following form:

$$\underbrace{0_1 \ldots 0_M}_{M}\, 1\, \underbrace{b_1 \ldots b_M}_{M}.$$

The codeword consists of M zeros, the *leading zeros*, one "1" followed by M bits, the *info field*. The number of bits in the info field, M, depends on the number of zeros preceding the first one. Thus, the number of leading zeros forces the length of the codeword to be equal to $2M+1$. The effect of even a single bit inversion in the first M zeros is depicted in Fig. 3. The original bitstream consists of four words of length 11 ($M=5$), 7 ($M=3$), 5 ($M=2$), and 1 ($M=0$), respectively. Assuming now an error inverting the third

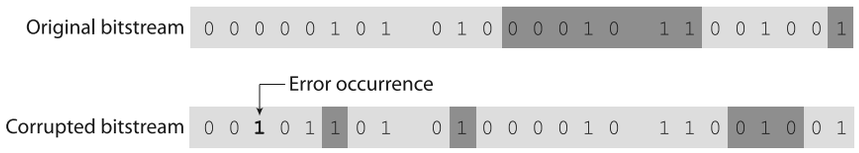

Fig. 3. Decoding desynchronization at bitstream level

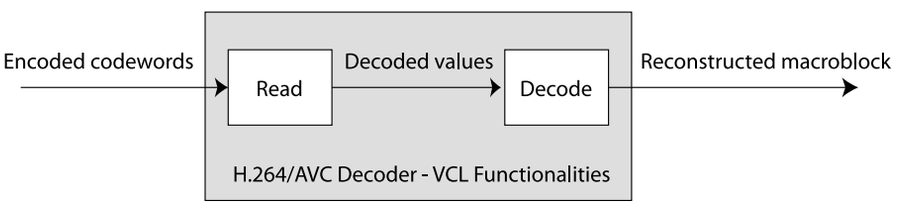

Fig. 4. H.264/AVC functional decoder blocks

bit, in the corrupted bitstream a "1" is found in the third position. The first decoded codeword has, therefore, $M = 2$ and length five. As the boundaries of the first codeword have been misinterpreted, the following codewords are incorrectly interpreted as well. As no resynchronization words are allowed in H.264/AVC, the *decoding desynchronization* may propagate up to the end of the NALU.

For exp-Golomb encoded codewords, a single error occurring in the leading zeros or in the first "1" causes decoding desynchronization, since the parameter M is evaluated incorrectly. In case the info field is corrupted while the parameter M is not changed, there is no direct desynchronization caused by the misunderstanding of the codeword boundaries. However, as the encoded value is incorrectly reconstructed, the decoder can misinterpret the meaning of the following decoded codewords. For instance, assume that an error is affecting the info field of a codeword, indicating the number of subblocks in a macroblock. Although no direct desynchronization occurs, the decoder will read the necessary parameters (such as motion vectors) for a wrong number of partitions, leading again to decoding desynchronization.

The described behavior is a known shortcoming associated to variable length coding. As the length of the codeword is embedded (or depends) on the content of the codeword itself, the efficiency of entropy coding is paid by a reduced robustness to errors if compared, for instance, to fixed length coding. One single bit inversion, in fact, causes both the wrong decoding of the current codeword as well as the misinterpretation of the boundaries of the current and possibly all of the following codewords.

3. Syntax-based error detection

After introducing the effects of errors both at sequence and packet level, in this section the proposed error handling mechanism will be discussed. It has been described how the errors only affect the decoding of the current and, possibly, following codewords. The information elements preceding the error occurrence can still be correctly decoded. Since the position of the error is not known, the definition of efficient error detection mechanisms is a task of major importance. In the following, an error detection mechanism based on the code syntax analysis will be presented. A similar method for the H.263 codec was proposed in *Barni et al.* (*2000*). An enhancement of the method for H.264/AVC was necessary both for dealing with the more complex bitstream structure and newly introduced entropy coding mechanisms, as well as because of specific synchronization words between Groups Of Blocks (GOB) allowed in H.263 and not in H.264.

The detection mechanism described in this section has been implemented in the standard reference software JM (*H.264/AVC JM Reference Software 2008*). The analyzed decoder functionalities have been subdivided into the following two elements:

(1) Read: Evaluate the value obtained when reading a codeword.
(2) Decode: Use the obtained value to reconstruct the macroblock.

This conceptual distinction is consistent with the two different logical functions defined in the JM: Read and Decode one macroblock, as indicated in Fig. 4. We have highlighted the two conceptual functions for two reasons: firstly, we will use them in the following discussion and secondly, the "read" function works at a "code" level, while the "decode" function works at a "pixel" level.

In the first function, "Read", each codeword is interpreted, obtaining the encoded value. In the second function, "Decode", such value is then employed to reconstruct the picture in the pixel domain, applying, for example, motion compensation or correction by means of residuals.

The original JM software relies on the detection mechanisms of the underlying protocol layers, which, upon detecting an erroneous packet, mark the packet by setting a flag. Thus, JM assumes that packets marked as error-free contain a validly encoded portion of the bitstream. If instead a portion of a bitstream, containing bit inversion(s) is introduced, the decoding will likely crash. Such crashes are due to unexpected codewords or values causing exception in the reading or decoding phase.

4. Concealment strategies

4.1 Error handling mechanisms

In the following, three different error detection and handling mechanisms are discussed. Independently of the way the error is detected, all the macroblocks that are marked as erroneous are concealed using the *copy-paste* strategy. Assuming that the macroblock in row i and column j of the f-th frame, $\mathrm{MB}_f(i,j)$ is labeled as corrupted, it will be replaced by the macroblock $\mathrm{MB}_{f-1}(i,j)$ occupying the same position in the previous picture.

(1) **Straight Decoding (SD).** This approach consists of the decoding of the corrupted bitstream by means of a modified H.264/AVC decoder. Without any detection mechanism, in case one of the syntax errors is detected, the decoder replaces the invalid value with its closest valid value. Since no error detection is

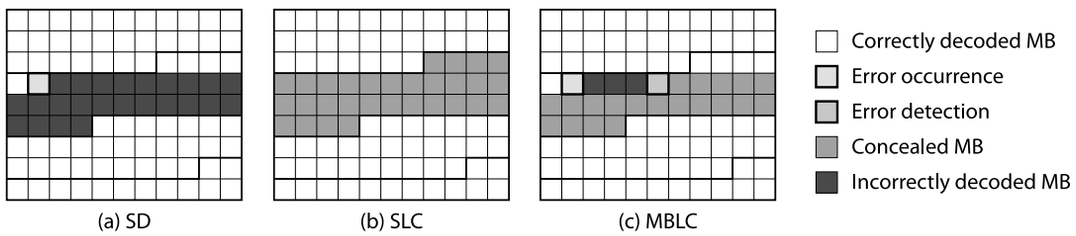

Fig. 5. Error handling mechanisms

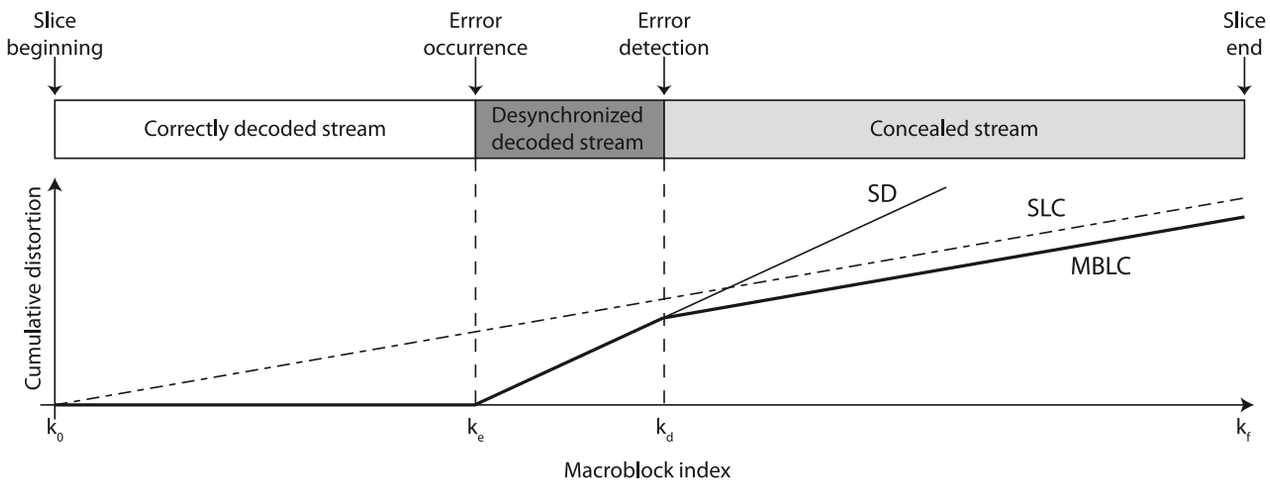

Fig. 6. Cumulative distortion using the three considered approaches

performed, the macroblocks preceding the error occurrence are correctly decoded, whereas the following ones are incorrectly decoded.

(2) **Slice Level Concealment (SLC).** This approach represents the standard error handling mechanism currently considered in the literature. In case a packet fails the UDP checksum test, it is discarded. All the MBs contained in the NALU are marked as corrupted and concealed. Also the MBs preceding the error are concealed, even if this is not necessary.

(3) **MacroBlock Level Concealment (MBLC).** This handling mechanism exploits the syntax analysis. Even though the packet fails the UDP checksum test, it is decoded. If a syntax error is detected, the current macroblock as well as all the following ones until the end of the slice are marked as erroneous. A detailed description of the syntax based error detection method is offered in *Superiori et al.* (*2006, 2007a, 2007b, 2009*). As discussed before, the error does not necessarily occur in the same macroblock where it is detected. This causes the macroblocks between the error occurrence and the error detection to be, possibly, incorrectly decoded. However, the information preceding the error detection is correctly exploited whereas the data after the error detection is marked as corrupted and concealed. The proposed method represents a compromise between the before mentioned two considered approaches.

In Fig. 5, the decoding of a picture using each of the three approaches is discussed. In this particular example, the same slice structure is used in all three cases, namely a fixed threshold on the amount of bytes in each packet. However, the particular choice should not play any role here. The presented discussion is valid for all slicing methods, such as fixed amount of MB on a packet, FMO and so on. The considered picture is an Intra predicted frame, consisting of four slices. An error has been introduced in the second slice, that contains the MBs with index from 30 to 59. The error will therefore propagate spatially until the end of the second slice, namely up to MB 59.

In Fig. 5(a) the decoding of the corrupted packet considering the straight decoding approach is shown. It can be noticed that, beginning from MB 35, the impairments propagate till the end of the slice. The MBs before the error occurrence are correctly decoded.

Figure 5(b) shows the result of the decoding when implementing the slice level concealment approach. Since the packet failed the UDP checksum test, the whole slice is concealed. MBs 30 to 34, even if correctly decodable, have been concealed as well.

The result of the proposed method (MBLC) is shown in Fig. 5(c). MBs 30 to 34 are correctly decoded. The error is detected in MB 39, therefore MBs 39 to 59 are concealed. In this example, a detection distance of four macroblocks has been assumed.

Figure 6 shows a qualitative comparison between the three proposed methods. The cumulative distortion between the slice decoded without errors and the corrupted slice handled by the three considered mechanisms is displayed. It can be noticed that, using SD, the distortion starts increasing as soon as the error occurs (k_0). The concealment method SLC is applied to the entire slice from k_0 to k_f and introduces a lighter grade of distortion all over the slice. The proposed approach, MBLC, limits the distortion caused by desynchronized decoding to the macroblocks between k_e, the point where the error actually occurs, and k_d, the point where the error has been detected. The distortion introduced by the concealment is spread from the error detection k_d until the end of the slice k_f. The performance of the proposed strategies depends strongly on the detection distance and on the position in which the error occurs. For

errors detected near the beginning of the slice, MBLC acts as SLC. The performance of MBLC depends on having "small" detection distances as increasing such distances worsens its performance.

4.2 Objective quality analysis

The first analysis performed regards the resulting quality in terms of Luminance Peak Signal to Noise Ratio (Y-PSNR). For a set of sequences obtained by encoding the Foreman video with different quantization parameters, the performance of the three strategies have been compared for error patterns characterized by different Bit Error Ratio (BER) values. Lower BERs have lower probability of errors, therefore the number of simulations increases with diminishing BERs.

Simulation Setup In order to evaluate the performance of the three considered methods, the decoding of differently corrupted sequences has been simulated. The sequences were encoded by the standard development encoder without any modification. Depending on the considered handling method, different features in the JM decoder were enabled. In case of SLC, once an error has been detected within the packet, all the macroblocks stored in the NALU are marked as erroneous. When considering SD, all the incorrect codewords or values are turned to a valid value. During the decoding process in MBLC, an error flag is raised in case invalid codewords or values have been detected. Once the flag has been raised, the decoder stops reading and decoding the macroblock and starts concealing it. The flags are turned back to zero at the beginning of the following slice. The sequence used for the simulation is the well-known "Foreman". It consists of 400 frames played at 30 frame per second with a resolution of 176×144 pixels (Quarter Common Intermediate Format (QCIF)). The GOP size was set to 10 and different Quantization Parameter (QP) values have been investigated. The variables listed in Table 1 facilitate the reading of this paragraph.

The graphs in Fig. 7 show the resulting objective quality in terms of Y-PSNR. The four lines drawn represent, respectively, the quality when considering the error free decoding, therefore not depending on the BER and considered as the reference quality, the straight decoding (SD), the slice level concealment (SLC), and the macroblock level concealment (MBLC).

The MBLC method performs better than the standard handling mechanism, particularly in the range of BERs from 10^{-4} to 10^{-6}. For lower BERs, the three mechanisms are close to the error free case, since the number of errors introduced is negligible. For higher BERs, such as 10^{-2}, the errors are introduced so frequently that the overall quality of the video is unsatisfactory, independently from the chosen handling mechanism.

The curve indicating the performance of SLC describes the behavior of the standard error handling approach. In the following, an analytical model is derived for approximating the empirical results. The simulations have been performed using Intra and Inter encoded pictures (Bi-directional temporal prediction was not allowed) and QP fixed to 28. With the considered settings, the size of an encoded *P* frame is smaller than the maximum packet size, that is 750 bytes, whereas one *I* frame requires six packets in average.

The bit error probability has been converted into the frame error probabilities P_I and P_P for the *I* and *P* frames, respectively. These

Table 1. Commonly used variables

Variable	Meaning
P_I, P_P	Frame error probability in *I* and *P* frames
P_{TP}	Probability of a damaged frame in a GOP
$P_{I,SC}(x)$	Probability of detected error in *I* frame at distance *x*
$P_{P,SC}(x)$	Probability of detected error in *P* frame at distance *x*
p_b	Bit error probability
n_b	Size of encoded frame in bits
$MSE_{P,TX}$	Mean Square Error (MSE) of *P* frames at decoder
$MSE_{P,EF}$	MSE of correctly decoded *P* frames
$MSE_{P,ERR}$	MSE of incorrectly decoded *P* frames
$MSE_{I,TX}$	MSE of *I* frames at decoder
$MSE_{I,EF}$	MSE of correctly decoded *I* frames
$MSE_{I,ERR}$	MSE of incorrectly decoded *I* frames
$MSE_{SI,ERR}$	MSE of incorrectly decoded *I* slices

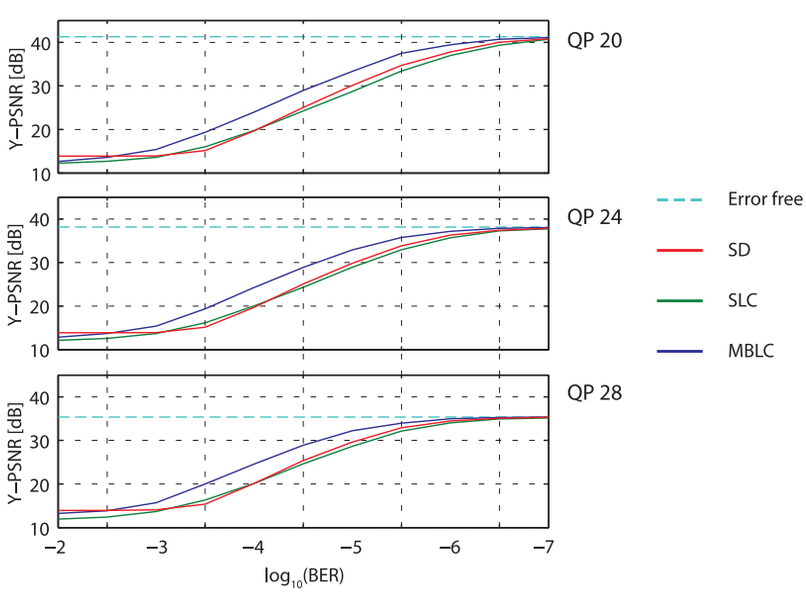

Fig. 7. Y-PSNR comparison using different handling mechanisms and QPs

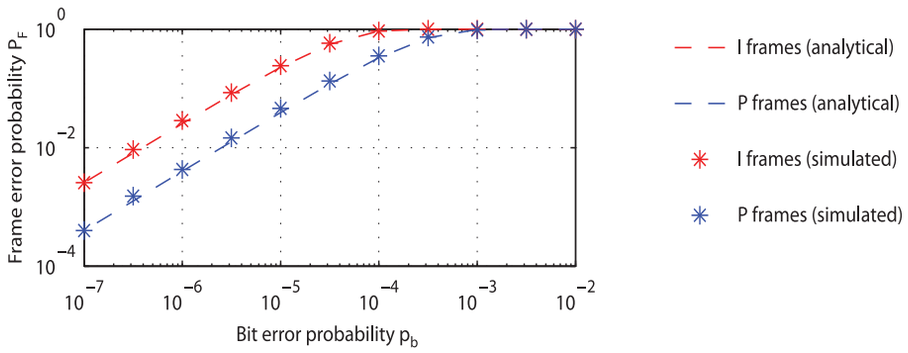

Fig. 8. Bit error probability mapped to packet error probability

probabilities are defined as follows:

$$P_F = 1 - (1 - p_b)^{n_b}, \tag{1}$$

where $P_F \in \{P_I, P_P\}$ indicates the generic frame error probability, p_b the bit error probability and n_b the size of the encoded frame. A comparison between the packet error probability calculated by (1) and those measured by simulation is shown in Fig. 8.

As the average size of an encoded *P* frame is smaller than the threshold set on the size of one packet, one *P* frame fits on a single packet. Therefore, the frame error probability coincides with the packet error probability. In the following discussion, all the quality measurements will be performed employing the MSE rather than the Y-PSNR, as the former considers linear values whereas the second a logarithmic scale. Y-PSNR and MSE are linked by the equation

$$\text{Y-PSNR} = 10 \cdot \log\left(\frac{255^2}{\text{MSE}}\right). \tag{2}$$

The average MSE of the *P* frames after decoding, $\text{MSE}_{P,\text{TX}}$ has been expressed as a function of the before mentioned probabilities:

$$\begin{aligned}&\text{MSE}_{P,\text{TX}}\\ &= \underbrace{P_I \cdot \text{MSE}_{I,\text{ERR}}}_{\text{I}}\\ &\quad + \underbrace{\frac{1-P_I}{\text{GOP}-1}\sum_{i=1}^{\text{GOP}-1}\left((1-P_P)^i \cdot \text{MSE}_{P,\text{EF}} + \left(1-(1-P_P)^i\right)\cdot \text{MSE}_{P,\text{ERR}}\right)}_{\text{II}}.\end{aligned} \tag{3}$$

The quality of the *P* frames strongly relies on the correctness of the previously decoded frames. Considering a single GOP, and then generalizing, the first underbraced term refers to the quality of the reference *I* frames. The value of $\text{MSE}_{I,\text{ERR}}$ defines the average MSE of an erroneous *I* frame. In case the *I* frame is correct (second underbraced term, labeled II) the average quality of the *P* frame depends on the previously decoded *P* frames. The quality of the *i*-th *P* frame can be expressed as the probability of the *i*-th frame and the previous *P* frames belonging to the same GOP to be correct, $(1-P_P)^i$, multiplied by the average MSE of the correctly decoded *P* frames, $\text{MSE}_{P,\text{EF}}$. This latter term consists itself of two summed elements: (i) the difference caused by lossy compression between the reconstructed and the original picture and (ii) the average MSE of the damaged *P* frames, $\text{MSE}_{P,\text{ERR}}$ multiplied by the probability that an error affects the current or the previous *P*. For averaging, all the possible GOP positions are considered and averaged.

Once an error has occurred, the following *P* frames belonging to the same GOP are affected by temporal error propagation. For obtaining the result in (3) and those following, two assumptions have been made:

(1) A second error occurring in a GOP that has already been affected by temporal error propagation, is not the cause of additional distortion.
(2) The quality degradation after the first error occurrence remains constant. This second argument is true only in average: in practice it may increase, decrease or remain constant depending on the sequence characteristics, and encoder settings (for instance, settings with respect to inserting the *I*-MBs into temporally predicted frames).

For the *I* frames, an equivalent formulation describing the average *I* frames quality, $\text{MSE}_{I,\text{TX}}$, has been defined:

$$\begin{aligned}&\text{MSE}_{I,\text{TX}}\\ &= \underbrace{(1-P_I)\cdot \text{MSE}_{I,\text{EF}}}_{I}\\ &\quad + \underbrace{P_I \cdot N(p_b)\cdot\left((1-P_{\text{TP}})\cdot \text{MSE}_{\text{SI,ERR}} + P_{\text{TP}}\cdot\frac{\text{MSE}_{P,\text{ERR}}\cdot\delta_{\text{ERR}}}{M_{\text{ave}}}\right)}_{\text{II}}.\end{aligned} \tag{4}$$

The first term of the equation considers the case of error free reconstruction of the *I* frame, the second refers to the case of reconstruction by means of error concealment. The first term describes the contribution of correctly reconstructed *I* frames, that occur with probability $(1-P_I)$.

As mentioned before, an *I* frame consists of more than a single packet. The size of the packet is fixed in number of bytes, the number of packets necessary for encoding the entire frame depends on the characteristics of each frame. In the considered scenario an *I* frame consists of 5.33 packets, in average. The factor $N(p_b)$ describes the average amount of corrupted slices with the bit error probability p_b. Each *I* packet is self-contained that is, it is reconstructed using only the information contained in the packet without exploiting any data belonging neither to the same nor to another frame. This effect is described in Fig. 9.

As the distortion of a single slice does not depend on the quality of the other slices belonging to the same frame, the distortion increases linearly with the number of damaged slices. In case one slice is concealed, the measured distortion (the parenthesis in the second term of (3)) depends on two terms. As the implemented concealment mechanism is simple copy-paste, in case the *I* frame is not correctly received the performance depends on the quality of the previous frame, this is the last *P* frame of the previous GOP.

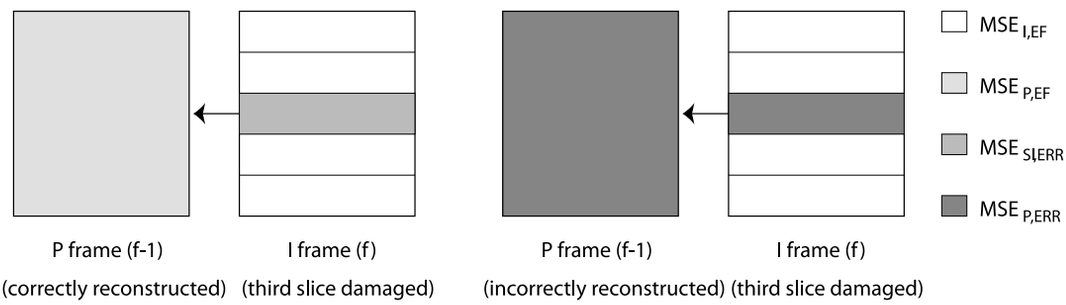

Fig. 9. MSE of corrupted *I* slices depending on the quality of the previous *P* frame

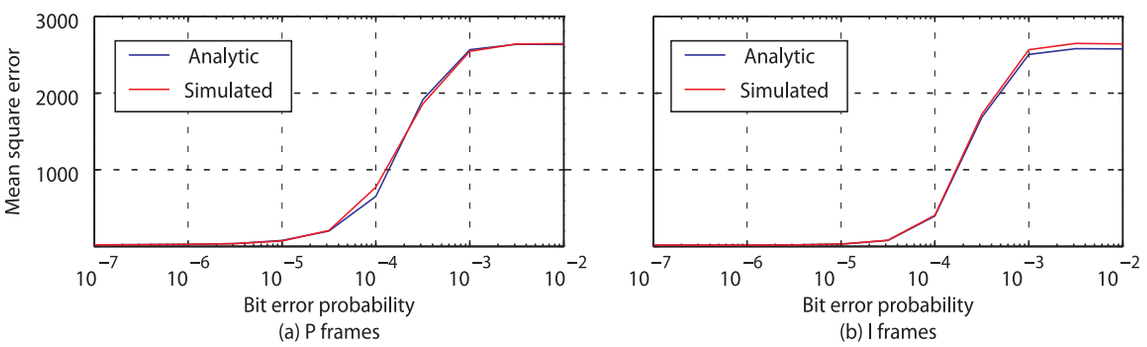

Fig. 10. MSE after transmission of *P* and *I* frames, $MSE_{P,TX}$ and $MSE_{I,TX}$, for QP = 28

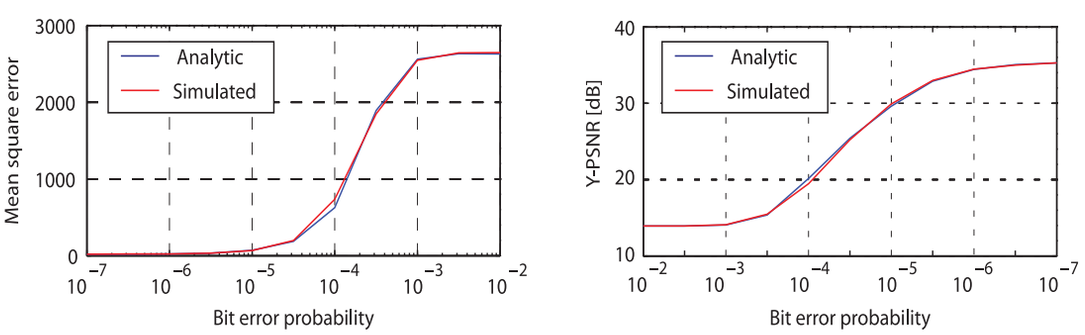

Fig. 11. Average MSE after transmission, MSE_{TX}, for QP = 28

The probability of the last frame not to be correctly reconstructed is given by

$$P_{TP} = P_I + \left(1 - (1 - P_P)\right)^{(GOP-1)}, \tag{5}$$

that is the probability of having a damaged frame in the GOP. In case the previous *P* frame is correct, $(1 - P_{TP})$, the distortion limited to a single slice is equal to $MSE_{SI,ERR}$. If the previous frame is damaged, P_{TP}, the distortion in the damaged slice is equal to the measured one in the corrupted *P* frames, but relatively limited to the concealed region. The term $MSE_{P,ERR}$, divided by the average amount of slices in an *I* frame M_{ave}, compensates a systematic error of the model, which occurs at higher BERs due to the previously introduced assumptions. For lower BERs, this term is equal to one. In particular, all the frames are, in average, damaged for BERs larger than 10^{-4}. This means that each frame is using an already corrupted picture for performing error concealment. When considering these conditions, the assumptions used for building the model are not in force anymore. However, as such a scenario does not provide any acceptable video quality anyhow, it has been preferred to concentrate the reliability range of the model to a reasonable BER interval.

In order to validate the proposed models, the empirical values obtained by simulation have been compared with the analytical estimations. Note that term I in (3) is equivalent to term II in (4). Equations (3) and (4) are used for calculating the average $MSE_{I,TX}$ and $MSE_{P,TX}$ of *P* and *I* frames, respectively, after transmission. The results are displayed in Fig. 10.

In order to obtain the average sequence mean square error, MSE_{TX}, the last two results have been averaged:

$$MSE_{TX} = \frac{1}{GOP} \cdot MSE_{I,TX} + \frac{GOP - 1}{GOP} \cdot MSE_{P,TX}. \tag{6}$$

The results are plotted in Fig. 11, both in terms of MSE (left) and Y-PSNR (right).

4.3 Detection performance

As discussed before, the detection probability as well as the detection distance represent key issues for the performance of the pro-

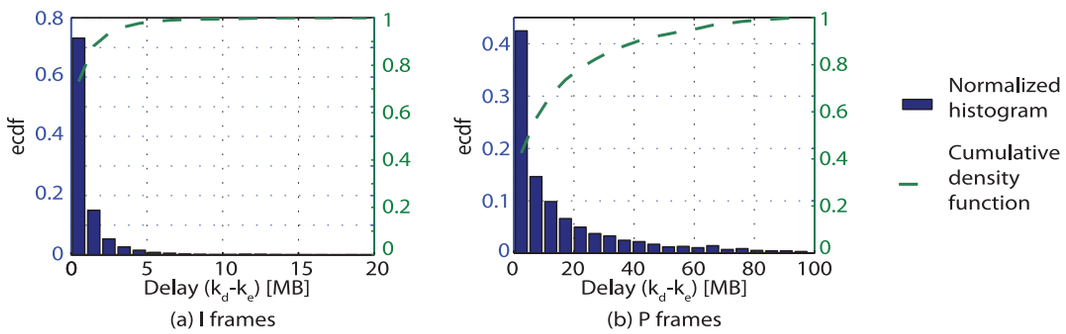

Fig. 12. Detected errors: distance between error occurrence and error detection, $k_d - k_e$

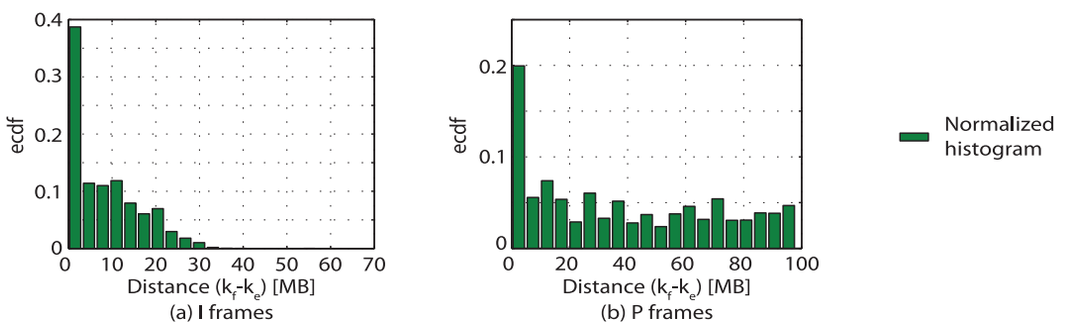

Fig. 13. Undetected errors: distance between error appearance and end of slice, $k_f - k_e$

posed method. Since the desynchronized decoding remains limited to the single NALU, new simulations were performed introducing a single error in each slice. It should be therefore underlined that the focus of these simulations is given on the detection performances and not on the quality, that would also be dependent on the temporal propagation of the resulting errors.

The simulations were performed on a sequence encoded with quantization parameter QP = 28 and NALU size limited to 700 bytes. These encoding parameters cause the *I* frames to be segmented in more than four slices, whereas the *P* frames usually consist of a single NALU.

The probability of detecting an error in the *I* frames is around 60 %, whereas for the *P* frames it is around 47 %. This difference can be explained considering the information elements encoded in the two different encoded slices. The *I* frames are self-contained: the encoded information is sufficient to reconstruct the picture without referencing any other previous decoded picture. This results in a less effective prediction and in a set of coefficients with higher information content. Such coefficients are much more sensitive to bit inversions and to desynchronized decoding. The packets containing encoded *P* frames, on the contrary, contain the information for reconstructing the picture applying motion compensation to the previously decoded pictures. Most of the information is contained in the motion vectors, describing the position of the best prediction of the considered block in the reference picture. Errors affecting the motion vectors are hardly detectable.

Figure 12 shows the detection distance, expressed in number of macroblocks, for the two types of predicted frames. Both normalized histograms (empirical cumulative distribution function (ecdf)) have an exponential trend. For the *I* frames, 90 % of the detected errors in the *I* frames are detected within 2 MBs. For the considered encoding settings the range varies between 0 and 20 MBs. An exponential trend describes well the probability of detecting the error in distance $k_d - k_e$ in *I* frames:

$$P_{I,SC}(k_d - k_e) = 1.552 \cdot \exp\big(-1.508 \cdot (k_d - k_e)\big). \tag{7}$$

The fit has been obtained by means of a non-linear Least Squares (LS)-fit.

For the *P* frames, this detection increases and, in average, the detection occurs within 7 MBs. The detection distance for the *P* frames is higher mainly because of a specific element encoded in the *P* frames, namely the `mb_skip_run`. It signalizes how many macroblocks have to be skipped, that is how many macroblocks can be reconstructed without further encoded information. Errors affecting this parameter will modify the number of skipped macroblocks. Since, in most of the cases, the encoded value is zero, an error will increase the number of the skipped macroblocks and, therefore, the detection distance. The model that has been found to best approximate the probability of detecting errors in *P* frames at distance $k_d - k_e$ in the normalized histogram has the form

$$P_{P,SC}(k_d - k_e) = 0.757 \cdot \exp\big(-0.386 \cdot (k_d - k_e)\big) + 0.146 \cdot \exp\big(-0.046 \cdot (k_d - k_e)\big), \tag{8}$$

the coefficients were again obtained by a non-linear LS-fit.

For the errors that have not been detected by the syntax analysis, a similar investigation has been performed. In this case, the distance between the error occurrence and at the end of the slice has been measured. The histogram of the distribution is plotted in Fig. 13. For the *I* frames, 50 % of the undetected errors are located in the last two macroblocks of the slice. Also for the *P* frames a peak is recognized for short distances between the error occurrence and the end of the slice. This shows how a considerable number of errors cannot be detected because the number of macroblocks to be decoded after the error occurrence is smaller than the average detection distance.

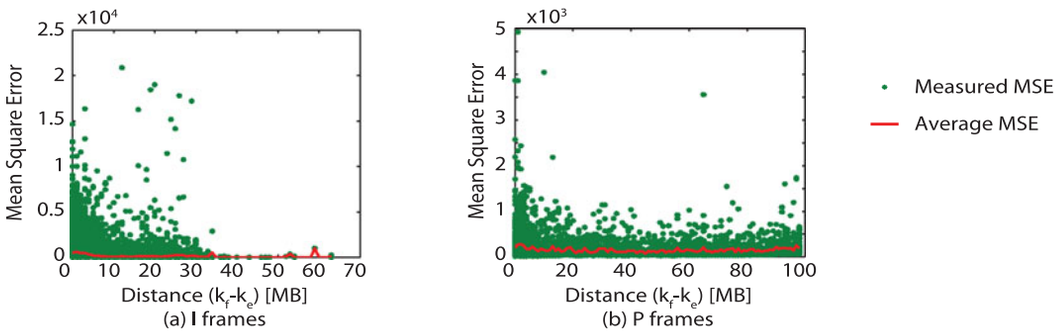

Fig. 14. Undetected errors: MSE calculated in the range $k_f - k_e$

In order to evaluate the influence of the undetected errors, their impact in terms of pixel-wise differences has been measured. In Fig. 14, the average distortion introduced by the missed detection is drawn as a function of the distance between the error occurrence and the end of the slice.

The distortion is measured in terms of MSE between the macroblocks affected by decoding desynchronization and the same macroblocks reconstructed in an error free environment. Small MSE values signalize that, even if the error has not been detected, the effects do not impair the decoded picture. Since the MSE is calculated over the whole area, possibly affected by decoding desynchronization, one might expect increasing MSE values for increasing detection distances. However, the performed simulations demonstrate that the resulting MSE does not depend on the number of affected macroblocks. This leads to the conclusion that, in average, the undetected errors do not affect significantly the decoded picture.

The previous reasoning is not valid for small values of $k_f - k_e$ that, in average, causes the highest MSE value. As the MBLC suffers from a detection distance, these errors were not detected for being too close to the end of the slice. In other words, these undetected errors cause harmful artifacts, resulting in increasing MSE, and would be detected by MBLC if enough macroblocks were following.

5. Visual artifacts detection

The syntax analysis still suffers when error detection distance increases. In the region between error occurrence and error detection, the encoded information is incorrectly decoded. This may result in visual impairments in the pixel domain. The detection of such visual artifacts can further improve the performance of the syntax analysis, by reducing the detection distance. However, the detection of impairments in the video domain calls for refined preprocessing techniques at pixel level. An analysis over video sequences obtained by decoding corrupted bitstreams has been performed and is described in the following. It is worth noticing that the effect of desynchronization was significantly different depending on the type of frame, Inter or Intra.

5.1 Inter frame impairments

When detecting errors in the f-th frame, we assume the frame $f-1$ to be correct. To detect errors in P frames, we analyze the pixel-wise difference map $\Delta_f(i,j)$ between frame f and frame $f-1$:

$$\Delta_f(i,j) = \left|F_f(i,j) - F_{f-1}(i,j)\right|. \tag{9}$$

Since we aim to detect artifacts with the resolution of a macroblock, the difference map Δ_f is then reshaped considering the average difference in 16×16 pixels. The difference map is not only dependent on possible visual artifacts, but also on the movement between the two consecutive pictures. The artifacts represent isolated, out of context, square regions of pixels. We therefore propose to implement a simple edge detector to highlight edginess in the picture. It has been noticed that observing the edge characteristics of 8×8 pixels sub-macroblocks represents the best compromise between false and missed detections.

The final decision whether one block k is detected as erroneous or not, is then taken considering the information about the difference map $D_f(k)$, defined as

$$D_f(k) = \sum_{(i,j)\in k} \left|\Delta_f(i,j)\right|, \tag{10}$$

as well as the edginess map $E_f(k)$, defined as

$$\begin{aligned} E_f(k) = {} & \frac{1}{8}\cdot\sum_{l=0}^{7}\left|F_f(i_1, j_1+l) - F_f(i_1-1, j_1+l)\right| \\ & + \frac{1}{8}\cdot\sum_{l=0}^{7}\left|F_f(i_1+l, j_1) - F_f(i_1+l, j_1-1)\right| \\ & + \frac{1}{8}\cdot\sum_{l=0}^{7}\left|F_f(i_1+7, j_1+l) - F_f(i_1+8, j_1+l)\right| \\ & + \frac{1}{8}\cdot\sum_{l=0}^{7}\left|F_f(i_1+l, j_1+7) - F_f(i_1+l, j_1+8)\right|, \end{aligned} \tag{11}$$

being i_1 and j_1 the coordinates of the upper-left pixel of the considered subblock.

For each block, $E_f(k)$ represents the average difference between the rows (resp. column) at the border of the macroblock and its neighbors belonging to a surrounding macroblock. They are both compared with an adaptive threshold considering the movement characteristic of the whole picture.

5.2 Intra frame impairments

In the Intra predicted frames the correlation between neighboring macroblocks belonging to the same picture is exploited. Each block is predicted considering the luminance and chrominance component of the confining already encoded blocks. At the decoder side, the quality of a single block strongly depends on the correctness of the neighbors. The spatial propagation of the errors in the I frames may occur also in case no desynchronization in decoding takes place. As an example, assume the codeword of a Variable Length Coding (VLC) residual to be affected by a bit inversion in the info field; therefore not causing decoding desynchronization. The considered block will be reconstructed differently from the block available at the encoder. The following blocks exploiting that macroblock as a reference will suffer of a spatial error propagation, since the obtained

(a) I frames (b) P frames

Fig. 15. Visual impairments caused by undetected errors

prediction will not be consistent with that considered at the encoder side.

Therefore, as shown in Fig. 15(a), an error in the *I* frames usually corrupts the affected macroblocks as well as its successors until the end of the slice. The different behaviors observed in the two kinds of frame call for the design of two different detection mechanisms.

To ensure robustness to the detection mechanism, the decision is taken considering a voting system. Similarly to the detection performed in the inter frames, the input to the voting system are the block difference map $D_f(k)$ and the edge map $E_f(k)$. Since the detection performance of the syntax analysis in the Intra frames was significantly better, the error position as detected by the syntax analysis is considered as well.

A scoring procedure is initialized each time the difference and edginess value of a block surpass a second threshold. This block is considered as the root of the artifacts sequence. The following blocks are further investigated: in case their difference and edginess characteristics are compatible with the artifact's features, the score is increased, otherwise decreased. A sequence of possible artifacts is terminated in the following cases:

(1) The score of the sequence surpasses a given threshold. In that case, the root of the sequence is assumed to be the macroblock where the error occurred. The following macroblocks until the end of the slice are marked as corrupted.
(2) The score of the sequence remains below a given threshold. In that case, the sequence is handled as a false positive. The detection is restarted and a new possible root is searched for in the following macroblocks.
(3) The syntax analysis signalizes that an error was detected in the current macroblock. The root of the sequence is considered as the macroblock where the error occurred. The following macroblocks until the end of the slice are marked as corrupted.
(4) The end of the slice has been reached, and none of the previous conditions were fulfilled. In this case the current score is compared with a second threshold. In case the current score exceeds it, the root is considered as the macroblock affected by the error. Otherwise, the whole sequence is treated as a false positive.

All the considered thresholds, as discussed for the Intra predicted frames, are adaptive and depend on the movement characteristic of the sequence.

5.3 Results

In order to measure the detection performance, the same simulation setup as described in Sect. 4.3 was considered. The results in terms of detection distance are shown in Fig. 16. For Inter encoded frames, the syntax analysis suffered of a significant detection distance. The detected errors were spotted, in average, after more than seven MBs. Performing the detection of visual impairments, such distance is slightly reduced to 6.8 MBs. The average motion compensation measured in the neighboring macroblocks is applied to those that are skipped, therefore they do not result as out of context macroblocks. The probability of detecting an error at distance $k_d - k_e$ in a *P* frame by means of visual detection, $P_{\mathrm{P,VD}}$, can be described as

$$P_{P,\mathrm{VD}}(k_d - k_e) = 0.788 \cdot \exp\big(-0.456 \cdot (k_d - k_e)\big) + 0.197 \cdot \exp\big(-0.057 \cdot (k_d - k_e)\big). \quad (12)$$

Even though the performance of the syntax analysis for Intra encoded frames was satisfactory, by means of visual artifact detection it has been further improved. The average detection distance, in particular, has been reduced from 1.39 MBs to 0.92 MBs. The normalized histogram of the detection distance for *I* frames follows an exponential trend. The probability of detecting an error at distance $k_d - k_e$ in an *I* frame by means of visual detection, $P_{\mathrm{I,VD}}$, is:

$$P_{I,\mathrm{VD}}(k_d - k_e) = 3.499 \cdot \exp\big(-2.714 \cdot (k_d - k_e)\big). \quad (13)$$

The average detection probability for *I* frames has been increased from 54.35 %, with the MBLC, to 59.99 % by means of visual error detection. Although the overall detection probability does not exceed 60 %, it has to be noted that, usually, the errors that do not cause desynchronization, do not produce any visible artifacts on the decoded picture. Also in the pixel domain, an error in the trailing ones can be barely spotted, since it would influence only high frequency components. Moreover, it remains questionable whether the detection of such errors would influence positively the resulting quality. The following macroblocks, in fact, can be correctly decoded and possible drifts in the spatial prediction would result in negligible distortion. Marking these macroblocks as corrupted would cause the concealment not to exploit the available valid information. More details are available under (*Superiori et al. 2007a*, *2007b*).

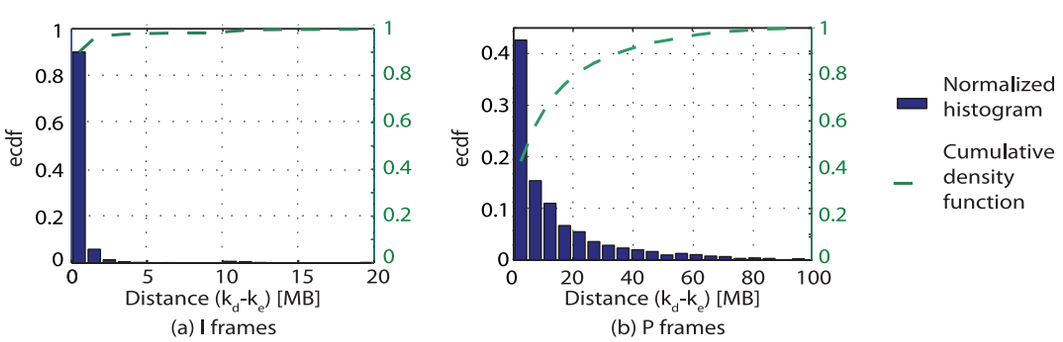

Fig. 16. Errors detected by visual impairments detection: $k_d - k_e$

6. Conclusions

Different error concealment strategies have been presented in this paper. The aim of these investigations is exploiting the fractions of the payload of erroneous packets that may still contain valid information. In order to reduce the protocol overhead, an Internet Protocol (IP) packet size is employed as close as possible to the network's Maximum Transfer Unit (MTU). In case the entire damaged packet is discarded, the decoder has to cope with a significant amount of missing information. As the effectiveness of the error concealment strongly depends on the size of the missing region, it is of major importance to keep this region as small as possible.

The proposed error detection strategies allow the decoder to recognize the damaged payload's region. Subsequently the data stored before the error occurrence can be exploited. The effectiveness of the strategies has been measured considering two metrics: (i) error detection probability and (ii) detection distance. The first index measures the amount of detected errors over the total error occurrences. The second index expresses the distance between the error occurrence and the error detection in multiples of macroblocks.

However, the error detection probability is not capable of distinguishing between errors that harm the perceived quality and those that does not cause any visible artifact. In this paper, we have also suggested a mechanism for detecting errors which are visually perceptible for a user. This enables to employ the analyzed error concealment strategies with higher efficiency, meaning that the artifacts which are not perceptible and thus not annoying to the user, are not concealed.

References

I-T H.264 (2005): Series H: Audiovisual and multimedia systems, infrastructure of audiovisual services–coding of moving video, advanced video coding for generic audiovisual services. Tech. Rep.

H.264/AVC JM Reference Software (2008): Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG. http://iphome.hhi.de/suehring/tml/.

Barni M., Bartolini F., Bianco P. (2000): Performance of syntax-based error detection in H.263 video coding: a quantitative analysis. In Vasudev B., Hsing T. R., Tescher A. G., Stevenson R. L. (Eds.), Image and video communication and processing 2000, vol 3974, no. 1, pp 949–956. http://link.aip.org/link/?PSI/3974/949/1.

Group A.-V. T. W., Schulzrinne H. (1996): RTP profile for audio and video conferences with minimal control. RFC 1890 (Proposed Standard), Internet Engineering Task Force, obsoleted by RFC 3551. http://www.ietf.org/rfc/rfc1890.txt.

Group A.-V. T. W., Schulzrinne H., Casner S., Frederick R., Jacobson V. (1996): RTP: a transport protocol for real-time applications. RFC 1889 (Proposed Standard), Internet Engineering Task Force, obsoleted by RFC 3550. http://www.ietf.org/rfc/rfc1889.txt.

Levine D., Lynch W., Le-Ngoc T. (2007): Observations on error detection in H.264. In 50th midwest symposium on circuits and systems, MWSCAS'2007, pp 815–818. doi:10.1109/MWSCAS.2007.4488698.

Marpe D., Wiegand T., Sullivan G. (2006): The H.264/MPEG4 advanced video coding standard and its application. IEEE Commun. Mag. 44(8):134–143. http://iphome.hhi.de/wiegand/assets/pdfs/h264-AVC-Standard.pdf.

Nemethova O. (2009): Principles of video coding. In Rupp M. (Ed.), Video and multimedia transmissions over cellular networks: analysis, modelling and optimization in live 3G mobile networks, pp 126–158.

Nemethova O., Al-Moghrabi A., Rupp M. (2006): An adaptive error concealment mechanism for H.264 encoded low-resolution video streaming. In Proceedings of 14th European signal processing conference (EUSIPCO), Florence, Italy. http://publik.tuwien.ac.at/files/pub-et_11103.pdf.

Postel J. (1980): User datagram protocol. RFC 768 (Standard), IETF, Tech. Rep. 768. http://www.ietf.org/rfc/rfc768.txt.

Postel J. (1981): Transmission control protocol. RFC 793 (Standard), IETF, Tech. Rep. 793, updated by RFCs 1122, 3168. http://www.ietf.org/rfc/rfc793.txt.

Sabeva G., Jamaa S. B., Kieffer M., Duhamel P. (2006): Robust decoding of H.264 encoded video transmitted over wireless channels. In Proceedings of MMSP, pp 9–13.

Stockhammer T., Hannuksela M., Wiegand T. (2003): H.264/AVC in wireless environments. IEEE Trans. Circuits Syst. Video Technol. 13(7):657–673. doi:10.1109/TCSVT.2003.815167.

Sun M., Reibman A. (2003): Chapter 3: Error concealment. In Compressed video over networks, signal processing and communications series, pp 217–250.

Superiori L., Nemethova O., Rupp M. (2006): Performance of a H.264/AVC error detection algorithm based on syntax analysis. In Proceedings of international conference on advances in mobile computing and multimedia (MoMM), Yogyakarta, Indonesian, pp 1–10. http://publik.tuwien.ac.at/files/pub-et_11446.pdf.

Superiori L., Nemethova O., Rupp M. (2007a): Detection of visual impairments in the visual domain. In Picture coding symposium proceedings, Lissabon, Portugal. http://publik.tuwien.ac.at/files/pub-et_12809.pdf.

Superiori L., Nemethova O., Rupp M. (2007b): Performance of a H.264/AVC error detection algorithm based on syntax check. J. Mob. Multimedia 3(4):314–330. http://publik.tuwien.ac.at/files/pub-et_12377.pdf.

Superiori L., Weidmann C., Nemethova O. (2009): Error detection mechanisms for encoded video streams. In Rupp M. (Ed.), Video and multimedia transmissions over cellular networks: analysis, modelling and optimization in live 3G mobile networks, pp 112–118.

Taferner M., Bonek E. (2002): Wireless Internet access over GSM and UMTS. Springer, Berlin.

Wenger S. (2003): H.264/AVC over IP. IEEE Trans. Circuits Syst. Video Technol. 13(7):645–656. doi:10.1109/TCSVT.2003.814966.

Wenger S., Hannuksela M., Stockhammer T., Westerlund M., Singer D. (2005): RTP payload format for H.264 video. RFC 3984 (Proposed Standard), IETF, Tech. Rep. 3984. http://www.ietf.org/rfc/rfc3984.txt.

Authors

Luca Superiori

received his B.Sc. and M.Sc. degrees in electronic engineering in 2002 and 2005, respectively, both from the University of Cagliari, Italy. In 2006 he joined the Institute of Communications and Radio-Frequency Engineering at the Vienna University of Technology (now: Institute of Telecommunications), where he received his Ph.D. in 2010. His areas of expertise are focused on video streaming over wireless networks, with a specific interest on error detection mechanisms for H.264/AVC streams, content-aware encoding for specific video contents (such as football) as well as cross-layer optimization of low-resolution video streams transmitted over 3G networks. He is currently employed as a consultant in the field of Communication, Media and Technology.

Olivia Nemethova

received her B.Sc. and M.Sc. degrees from Slovak University of Technology in Bratislava in 1999 and 2001, respectively, both in Informatics and Telecommunications. She received her Dr. techn. (Ph.D.) in electrical engineering from Vienna University of Technology with distinction in 2007. From 2001 until 2003 she was with Siemens as a systems engineer. She worked on UMTS standardization within 3GPP TSG RAN2 as a Siemens delegate. In parallel she worked within an International Property Rights management team responsible for evaluation of IPRs regarding RANs. In 2003 she joined the Institute of Communications and Radio-Frequency Engineering (now: Institute of Telecommunications) at the Vienna University of Technology as a research and teaching assistant. Her current research interests include error resilient transmission of multimedia over wireless networks, video processing and mobile communications.

Markus Rupp

received his Dipl.-Ing. degree in 1988 at the University of Saarbruecken, Germany and his Dr.-Ing. degree in 1993 at the Technische Universitaet Darmstadt, Germany, where he worked with Eberhardt Haensler on designing new algorithms for acoustical and electrical echo compensation. From November 1993 until July 1995, he had a postdoctoral position at the University of Santa Barbara, California with Sanjit Mitra where he worked with Ali H. Sayed on a robustness description of adaptive filters with impact on neural networks and active noise control. From October 1995 until August 2001 he was a member of technical staff in the Wireless Technology Research Department of Bell-Labs at Crawford Hill, NJ, where he worked on various topics related to adaptive equalization and rapid implementation for IS-136, 802.11 and UMTS, including the first MIMO prototype for UMTS. Since October 2001 he has a full professor for Digital Signal Processing in Mobile Communications at the Vienna University of Technology where he founded the Christian-Doppler Laboratory for Design Methodology of Signal Processing Algorithms in 2002 at the Institute for Communications and RF Engineering. He served as Dean from 2005 to 2007. He was associate editor of IEEE Transactions on Signal Processing from 2002 to 2005, is currently associate editor of JASP EURASIP Journal of Advances in Signal Processing and JES EURASIP Journal on Embedded Systems. He is elected AdCom member of EURASIP since 2004 and served as president of EURASIP from 2009 to 2010. He authored and co-authored more than 400 scientific papers and patents on adaptive filtering, wireless communications, and rapid prototyping, as well as automatic design methods.