Picture Coding Symposium 2007, 7-9 November 2007, Lisboa, Portugal

DETECTION OF VISUAL IMPAIRMENTS IN THE PIXEL DOMAIN OF CORRUPTED H.264/AVC PACKETS

Luca Superiori, Olivia Nemethova, Markus Rupp

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology, Austria
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: {lsuper, onemeth, mrupp}@nt.tuwien.ac.at

ABSTRACT

In this article we investigate the utilization of the corrupted IP packets (indicated by checksum) at H.264/AVC decoders. The position of the error within a packet has to be pre-localized by a syntax analysis. The impairments remaining after syntax analysis are further detected in the pixel domain, by means of a voting system, using difference frames and knowledge about the artifact appearance. The blocks considered as erroneous are concealed. The proposed detection of artifacts in the pixel domain considerably improves the probability of error detection achieved by the syntax analysis. The combined system clearly outperforms the typically employed slice discarding in terms of quality measured by mean squared error.

1. INTRODUCTION

The H.264/AVC (Advanced Video Coding) [1] is nowadays the best performing video coding standard. Due to its enhanced coding gain and error resilience, it is widely employed for transmission of real-time video over packet networks.

The units containing H.264/AVC stream fractions are typically encapsulated into RTP/UDP/IP packets. The UDP contains a simple checksum for detecting the errors. If an error is detected at the receiver, the entire packet is usually discarded and the missing part of picture is concealed. In literature several works dealing with the improvement of code robustness against errors were presented. By the exploitation of knowledge about the channel characteristics, in [2] several Joint Sources Channel Coding (JSChC) approaches based on residual redundancy are described. Error resilience tools at the encoder side are presented in [3]. The corrupted packet may still contain correctly received information. If channel decoding results in few residual errors only, the utilization of the corrupted packets may be extremely beneficial.

The variable codeword length of the H.264/AVC entropy code can lead to the *desynchronization* of the decoding after an error — in case the borders of codewords are not determined correctly. Desynchronized decoding results on considerable visual impairments. An optimal error handling strategy consists on detecting the position of the errors within the packet and conceal the macroblocks considered as erroneous. There are several approaches aiming the detection of errors in corrupted IP packets. Additional parity check bits increase the rate and are not compatible with the H.264/AVC standard. In [4], inserting of fragile watermarks allows for detecting the error position at the receiver. Such method cause the overall quality degradation of the stream. Sequential decoding [5] is capable of even correcting errors, however it increases the decoder complexity. In [6] the author proposed an error handling strategy based on syntax analysis. The proposal has negligible complexity and provides an essential improvement compared to the common slice rejection mechanism. The syntax analysis still suffer for undetected errors and errors detected after they occurrence. Such drawbacks result on annoying impairments on the decoded frame. In this work we face the detection of visual error effects.

The impairments caused by desynchronized decoding and later/missed detection are detected in the pixel domain. A rather complex, human visual system based method for detection of impairments was already proposed in [7] for estimation of video quality. However, the artifacts left by syntax check possess some characteristic features that allow for designing a low-complexity impairment detection mechanism.

This paper is organized as follows. Section 2 briefly summarizes the syntax analysis method for detecting the errors in entropy encoded video stream. Section 3 describes the proposed mechanism for detecting the impairments after decoding in the pixel domain. In Section 4 the experimental results are presented and discussed. Section 5 contains conclusions and some final remarks.

2. SYNTAX BASED ANALYSIS

The H.264/AVC standard [1] is conceptually subdivided in two interacting blocks: the Video Coding Layer (VCL) and

The authors thank mobilkom austria AG for technical and financial support of this work. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG.

the Network Abstraction Layer (NAL). The first deals with the hybrid block-based video coding functionalities. The encoded stream is then partitioned by the NAL into basic elements called NALUs (NAL Units). The NALUs can be flexibly adapted for both packet-oriented and bitstream-oriented transport system. Each VCL NALU contains a segment of the frame (called *slice*) that can be decoded independently of the other slices belonging to the same frame. For the considered network application, each NALU is further encapsulated into the IP/UDP/RTP transport protocol stack.

The coding functionalities are intended to reduce the video redundancy, both temporal and spatial. The VCL makes extensive use of lossless entropy coding. Entropy coding assigns codewords to symbols depending on their probability of occurrence: the higher the probability, the shorter the associated codeword. Depending on the syntax component, two families of Variable Length Coding (VLC) strategies are used: the universal *exponential Golomb code* [8] and the Context Adaptive VLC (CAVLC). They are both *prefix* codes characterized by a regular logical structure. The word length is contained into the *prefix* field. Possible errors affecting this field will cause the misinterpretation of the codeword boundaries and, therefore, *decoding desynchronization*.

In [6] the author proposed a mechanism for detecting errors in H.264/AVC sequences. Similarly to other approaches for different codecs [9, 10], by means of bitstream syntax analysis, different resulting error categories were defined and analyzed. The detection of errors occurs during the reading and interpretation of the code associated to a macroblock. If an error is detected while decoding a macroblock, all the following ones up to the end of the slice are marked as flawed. These macroblocks can be properly concealed.

There are still errors that cannot be detected by means of syntax analysis. Moreover an error is usually not detected immediately, but after some desynchronized decoding. In such cases, it propagates over the slice causing the following macroblocks to be decoded erroneously. As a consequence the resulting decoded image will be affected by visible artifacts.

3. ERROR LOCALIZATION IN PIXEL DOMAIN

Figure 1 shows the typical effects of desynchronized decoding on intra predicted (I) and inter predicted (P) frames. Since the artifacts are visually noticeable, we decided to analyze their characteristics in the pixel domain. Their detection calls for a refined post-decoding video processing. In [5], a method for detecting impairments remaining after a sequential decoding in I frames was proposed. In [11] a method for detecting transmission artifacts on image transmitted over wireless channel was presented. Our detection is based on a voting system that analyzes the difference frames and checks the size, shape and propagation of the candidate artifacts. A schematic block representation is drawn in figure 2.

In [6] the position of the detected syntax errors are used to conceal the macroblocks marked as flawed. In this work we discuss the possibility of performing the detection of visual artifacts directly on the desynchronized decoded frames using the syntax analysis as additional information. The concealment will be performed on the macroblocks where visual impairments have been detected. As in [6], this paper deals with the detection of errors independently from the concealment method. The latter can be chosen according to the error and frame characteristic. In Fig. 1 we observe notable difference between desynchronized decoding in I and P frame. Therefore, we decided to perform the detection separately by means of two different processing algorithms. The inputs of the visual impairment detection blocks are the desynchronized decoded video and the informations produced by the syntax analysis, namely the position of the detected errors and the dimension of the slices in macroblocks. The detection is only performed over the packets failing the UDP checksum, reducing the complexity and the possibility of false positive.

(a) I frames (b) P frames

Fig. 1. Visual impairments caused by undetected errors.

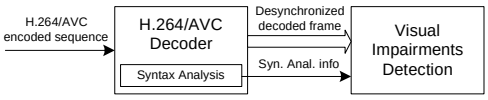


Fig. 2. Improved detection scheme

3.1. Detection in Inter Frames

Inter predicted frame exploits the temporal correlation between consecutive frames. The code associated to inter predicted macroblocks consists of the prediction information and residual levels. The residuals are indicative of the difference between the block to be coded and its prediction. Since the residuals are typically small or even skipped, their errors have often a negligible influence on the quality and on error propagation. The errors in motion vectors and/or reference frames cause artifacts similar to those of temporal error concealment, i.e. spatial shifting of affected macroblocks. The VLC desyncronizes rarely.

In addition, the absence of spatial prediction within a frame results in lack of direct artifact propagation within the same frame, as shown in Fig. 1(b). Even if the entire frame belongs to the same slice, the artifacts remain isolated and are spatially interleaved with sequences of consistent macroblocks.

Artifacts could be noticed observing the elementwise difference frame $D_n = |F_n - F_{n-1}|$ (calculated over all the color components), where the frame F_{n-1} is assumed to be correct. Besides impairments, the difference map can contain high magnitude components corresponding to movement. However, the considered artifacts lead to visual blockiness. Blockiness can be detected observing horizontal and vertical edges E_n of the considered frame F_n.

The detection is therefore performed separately for each 8×8 block, basing the decision on the combined observation of edge and difference characteristics. In order to handle sequences characterized by fast movements, the threshold is adapted to the statistic of the average difference magnitude.

3.2. Detection in Intra Frames

Intra predicted frames are encoded reducing the spatial redundancy of the image. The luminance and chrominance components of each macroblock are predicted from the previously encoded surrounding blocks. Since I frames encoding uses only self-reference, they act as sequence refreshers. This enables effectiveness of temporal prediction after scene change and terminates temporal propagation.

Due to spatial prediction and VLC decoding desynchronization, artifacts in I frames typically propagate until the end of the slice 1(a). In addition, they also affect the following frames referencing the macroblocks erroneously decoded in the I frames. The temporal propagation usually involves all the inter predicted frames until the next I frame.

For I frames, the detection works on top of the Syntax Analysis (SA). To ensure more resistance against false positives, the decision is made depending on the result of a voting system. Input to the decision mechanism is the block average difference $D_n(i,j)$, the edge frame $E_n(i,j)$ and the error position detected by the SA (SE). A procedure is initialized if the considered $D_n(i,j)$ lies over a threshold thr_1, and the block has edge. The initial vote depends on $D_n(i,j)$.

A procedure is initialized once the considered $D_n(i,j)$ lies over a threshold thr_1 and if edges are recognized, or $E_n(i,j) = 1$. If such conditions persist, then the characteristic of the considered block are compatible with the one of a visual artifacts. The block $B(i,j)$ is considered being the root of the sequence. An initial vote is assigned to the sequence depending on $D_n(i,j)$. Consistently with the previous discussion, we assumed that the artifacts in I frame do not appear isolate, but rather propagate. Therefore subsequent blocks are processed following the raster-scan order, as described in the following pseudo-code.

```
while (mb-idx != end-of-slice)
  if (mb HAS SE)
     syn-found = 1; QUIT;
  elseif ((mb HAS edge) AND (mb-diff > thr1))
     vote++;
  elseif (mb-diff > thr2) vote+;
  else vote--;
  end
  if (vote > upper-thr)
     vis-found = 1; QUIT;
  elseif (vote < lower-thr) clear-seq;
  end
  mb-idx+;
end
```

For a given packet, the procedure is terminated if the analyzed macroblock was recognized as damaged by the syntax check. Statistically, some desynchronized decoding will precede the error detection. A more exhaustive visual impairments detection will be therefore performed on the blocks preceding the error detected by means of syntax analysis.

If the sequence vote exceed a threshold thr_{HIGH}, the algorithm recognized a series of visual impairments and quit the procedure for the considered packet. The index corresponding to the macroblock where the threshold was overcome does not necessarily represent the error detection. Similarly to the previous analysis, all the blocks that contributed positively to the vote are examined in detail. Once the vote goes under a thr_{LOW}, the algorithm recognizes the current sequence as a false positive. The current sequence is deleted and the voting system is reinitialized.

In case the end of the slice has been reached and none of the previous cases has occurred, a potential existing sequence is evaluated with respect to a final threshold thr_{END}. In case the vote was bigger than the considered threshold, the algorithm considers the sequence originated by an actual error and performs further analysis on the sequence elements. Otherwise, according to the algorithm, any artifact was present on the considered the slice.

The considered thresholds are adapted to the statistical characteristics of the considered previous frames. Fast moving sequences will cause the thresholds to increase, since the difference picture will contain more component due to the movement. The macroblocks recognized as damaged are not taken into account while calculating the average difference of the considered frame. As mentioned, due to the encoding mechanism, an error in I frame will propagate up to the end of the slice. Therefore, for I frame, the chosen concealment mechanism will be extended also to all the macroblocks following the one considered as erroneous.

4. EXPERIMENTAL RESULTS

The performance of the proposed algorithm are tested decoding and processing a damaged H.264/AVC code. We used "Foreman" sequence in QCIF resolution, encoded in baseline profile with QP = 28. To simulate an error-prone binary symmetric channel, we inserted memoryless random errors during the decoding. In order to maximize the simulation set, these simulations were performed inserting one error each slice. We used the Joint Model H.264/AVC reference software (v.10.2)

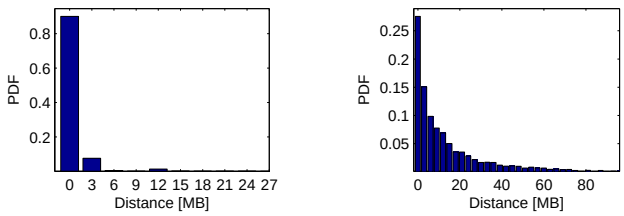
[12] adapted as in described in 3.

We evaluated the improvement of the proposed method compared to the syntax analysis. The comparison of each detection strategy (DS) (syntax analysis (SA) and artifacts detection (AD)) is performed separately for the P and I frames (FT). For both we calculate the detection probability (P_{det}). For the detected errors, we measured the average distance between the error occurrence and the error detection (Δ). The mean square error MSE_D caused by desynchronized decoding in that interval is considered aswell. For the undetected errors, we consider the average mean square error MSE_U calculated for the whole desynchronized decoded area, namely between the error occurrence and the end of the slice.

DS	FT	P_{det}	Δ	MSE_D	MSE_U
SA	P	48.75%	15.091	279.5	162.6
AD	P	48.54%	13.741	181.9	126.7
SA	I	54.35%	1.390	622.6	282.0
AD	I	59.99%	0.925	105.3	90.3

Table 1. Performance comparison

The simulations for P frame show for the proposed method a smaller detection probability, since we detect only errors that result in visual impairments. Both the MSE and the detection distance were improved. Since the proposed method for I frames is based on the SA, we observe tangible improvements. The average MSEs for both the detected and undetected errors are much smaller. Detection probability rises to 60% and the detected errors are spotted usually within one macroblock. The normalized histogram of the detection distance is plotted in figure 3 for I and P frames. As final result



(a) I frames (b) P frames

Fig. 3. Detction Distance

we evaluated the performance of the proposed method, simulating the transmission of an encoded sequence over a channel with Bit Error Rate (BER) equal to 10^{-5}. The graph in Fig. 4 shows the comparison between the Artifact Detection and the Syntax Analysis averaging the 20 considered simulations. An average improvement of 1.36 dB was obtained. As expected, the quality enhancement is particularly notable in case the error affects the I frame.

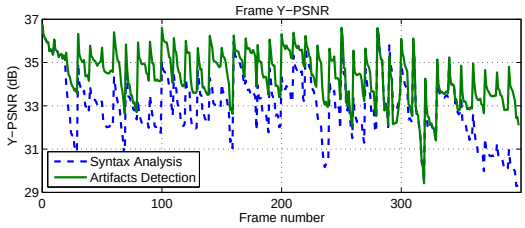


Fig. 4. Comparison of decoding quality

5. CONCLUSIONS

An application for detecting errors in the pixel domain has been presented in this study. Alternatively to the common slice rejection, we propose an hybrid error detection mechanism performed in the code and pixel domain. The proposed method performs the detection at macroblock level, allowing the decoder to exploit the correct information. The proposed method essentially improves the performance of the syntax analysis. Moreover, it does not require any additional overhead and relies on the decoder implementation and post-processing only.

6. REFERENCES

[1] ITU-T Rec. H.264 / ISO/IEC 11496-10, "Advanced Video Coding," Final Committee Draft, Document JVTE022, Sept. 2002.

[2] R.E. van Dyck, D.J. Miller, "Transport of wireless video using separate, concatenated, and joint source channel coding," IEEE Proceedings, vol. 87, no. 10, pp. 1734-1750, Oct. 1999.

[3] S. Wenger, "H.264/AVC Over IP," IEEE Trans. On Circuits And Systems for Video Techn., 13, no. 7, pp. 645-656, Jul. 2003.

[4] M. Chen, Y. He, R.L. Lagendijk, "A Fragile Watermark Error Detection Scheme fot Wireless Video Communications," in IEEE Trans. on Multimedia, April 2005.

[5] C. Weidmann, P. Kadlec, O. Nemethova, and A. Al-Moghrabi, "Combined sequential decoding and error concealment of H.264 video," in Proc. of MMSP, Italy, Sep. 2004.

[6] L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," in Proc. of MoMM, Indonesia, Dec. 2006.

[7] S. Winkler, "Digital video quality," Wiley, 2005

[8] J. Wen, J.D. Villasenor, "Reversible Variable Length Codes for Efficient and Robust Image and Video Coding," in Proc. of IEEE Data Compression Conf., Snowbird, Utah, USA, Apr. 1998.

[9] B. Yan, K. Wing, "Analysis and detection of MPEG-4 visual transmission errors over error-prone channels," IEEE Trans. on Cons. Elec. Vol. 49, Iss. 4, Nov. 2003 Page(s):1424 – 1430.

[10] E. Khan, H. Gunji, S. Lehmann, M. Ghanbari, "Error Detection and Correction in H.263 coded video over Wireless Network," Proc. of PV 2002, USA, Apr. 2002.

[11] S. Ye, X. Lin, Q. Sun, "Content based Error Detection and Concealment for Image Transmission over Wireless Channel," Proc. of ISCAS, Thailand, May 2003.

[12] H.264/AVC Software Coordination, "Joint Model Software," ver.10.2, available in `http://iphome.hhi.de/suehring/tml/`.