Proceedings of the 6th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China, April 16-18, 2006 (pp19-24)

The new Hybrid approach to protect MPEG-2 video header

*YUK YING CHUNG, *XIANG ZHANG, *XIAOMING CHEN,
*MOHD AFIZI MOHD SHUKRAN, **CHANGSEOK BAE
*School of Information Technologies, University of Sydney, NSW 2006, AUSTRALIA
** Post-PC Research Group, ETRI, 161 Gajeong Dong, Yuseong Gu, Daejeon, KOREA

Abstract: - This paper has proposed and implemented a new approach to detect MPEG-2 header errors using a MD5 hash function and information hiding technique. The video picture header information is protected by a MD5 hash technique, and any change in the header level is detected immediately. The information hiding technique is used to locate the error block number in the corrupted MPEG-2 video. The results show that the new approach can detect 100% of header error at both block level and higher levels without increasing the transmission bit rate.

Key-Words: - MPEG-2, video header, MD5, information hiding.

1 Introduction

The transmission error (cell loss/error) can be detected by the ATM adaptation layer (AAL) protocol [2]-[5] when using the support of network transmission functionality, such as Asynchronous Transfer Mode (ATM) transmission scheme, where. Thus, the blocks belonging to the damaged cells and all the subsequent cells before VLC resynchronizations are discarded. This results in a waste of the resource and reduces the real-time quality. Cuenca, P [3] presents an algorithm for solving the problem. He divides the MPEG-2 bitstream into two sub-bitstreams. One is termed base layer and the other the enhancements layer. All the headers and all the important information at microblock level such as motion vectors, motion types, microblock types and the DCT coefficients before a breakpoint will be transmitted on to the base layer. The rest of the DCT coefficients between the breakpoint and the end of block (EOB) and other less important information will be transmitted on to the enhancements layer. Different error concealment algorithms are applied to the two layers. If any cell loss or cell error on base layer is detected by AAL, discarding and resending will be executed immediately. If it happens at the enhancements layer, the lost parts will be simply replaced with zeros. Another algorithm presented by W. Luo [5] uses packetization techniques that rely on interleaving at the slice level. However this increases the complexity of decoder.

We can use the decoder's functionality to check if the information arrived is corrected or not. The following information must be checked at the decoder side.

1. Sequence header: it contains information such as the picture display size and chroma format,
2. Frame header: it contains intra DC precision and motion vector ranges, the DCT scan patterns, the picture structure, and the picture coding type, and
3. Other important information such as program specific information (PSI), Transport packet header, slice header [6].

A simple way to reduce the probability of degradation of the received images is to send duplication of the images or some of the data with high importance. But this results in a higher transmission bit rate. **F**orward **E**rror **R**ecovery (***FER***) techniques are developed to send these high import data to increase the quality [1] [6].

We can perform FER on both encoder side and decoder side. Some error-correction information is used to encode the images or add to the encoded images. Then the transmission error can be detected at the decoder side. The channel coding approach [7][8] uses this scheme. Aign. S [7] considered that the channel coding for terrestrial transmission should be based on a concatenate coding scheme with an outer systematic RS code and an inner convolutional code. The decoder side will receive the information and apply error concealment. However channel coding will increase the transmission bit rate because of the error-correction information. From above we find that most of the previous work [3][4][6][7] only focused on the detection of errors at the block level.

In this paper, a new hybrid approach is proposed and implemented to detect and locate the errors from corresponding corrupted MPEG-2 images at both block level and higher levels by using a MD5 hash function and information hiding technique to hide the error-correction information images without

increasing the transmission bit rate. Two detection functions have been carried out. First is to find out the picture header errors and locate them. The second is to locate the error happens on block levels. Section 2 explains the MD5 hash function. Section 3 explains the Information Hiding Technique. Section 4 gives the details about the new MPEG2 Header Protection System. Section 5 and section 6 show the experimental results and conclusion respectively.

2 Introduction to MD5 hash function

MD5 (Message-Digest algorithm 5) is a widely-used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. A hash function takes a long string (or message) of any length as input and produces a fixed length string as output, sometimes termed a message digest or a digital fingerprint.

MPEG-2 compressed video is very sensitive to data loss. To ensure the quality of the whole frame we need to make sure every bit in the header part is correct. One of the important characteristic of the hash function is that even a small change in the message will (with overwhelming probability) result in a completely different hash. The 128-bit (16-byte) MD5 hashes (also termed message digests) are typically represented as 32-digit hexadecimal numbers. The following demonstrates a 43-byte ASCII input and the corresponding MD5 hash function:

MD5
("The quick brown fox jumps over the lazy dog")=
9e107d9d372bb6826bd81d3542a419d6

Example 1: Changing d to c:
MD5
("The quick brown fox jumps over the lazy cog") =
1055d3e698d289f2af8663725127bd4b

Example 2: The hash function value of the zero-length string is:
MD5 ("") =
d41d8cd98f00b204e9800998ecf8427e

As we do not want increase the transmission bit rate, the hashed code will be hidden in the frame using the watermark technique so a fixed code length is used in the codec.

3 Information Hiding Technique

3.1 Encoder

The following will show how we embed the watermark information into MPEG-2 I-frames:

We have the following assumptions:
1) The test host image is a X*Y pixels image,
2) The watermark is a 128 bits MD5 hash code,
3) X*Y>=8*8*128/2 ; and
4) We take this 352*288 image as a example.

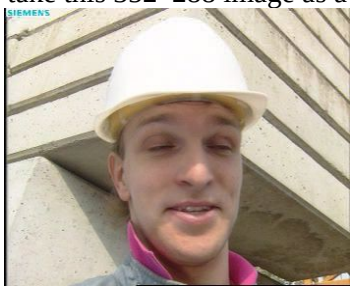


Figure 1 Test frame [0]

Figure 2 shows how to embedding the watermark information into a still image.

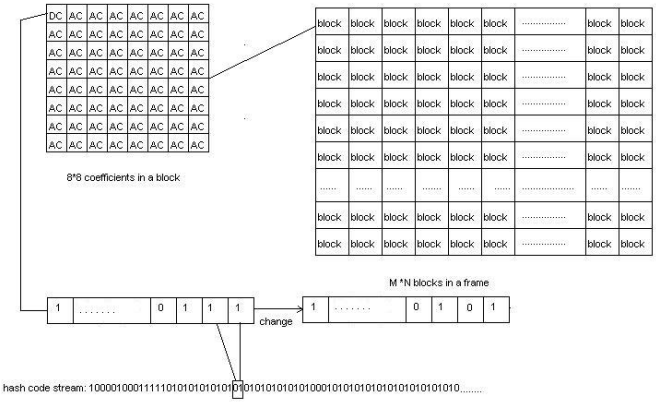


Figure 2 The process of embedding the watermark information into a image

We have divided the watermark embedding process into three steps. The following explains these three steps:

Step 1:
We divide the picture into 8*8 pixels blocks.
The example image can be divided into
352*288/64=1584 blocks

Step 2: Forward DCT transform

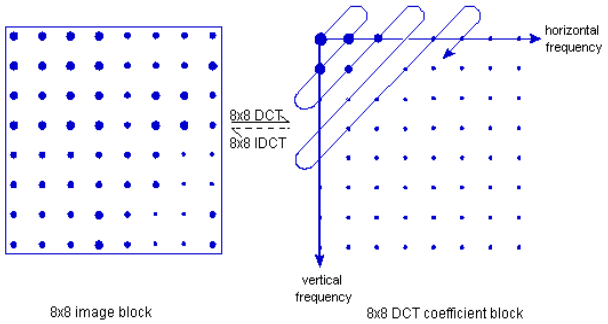


Figure 3 Forward DCT transform

Step 3:
Embed watermark:

1) Transfer the hashed code into one-zero string,
2) Zig-Zag scanning,

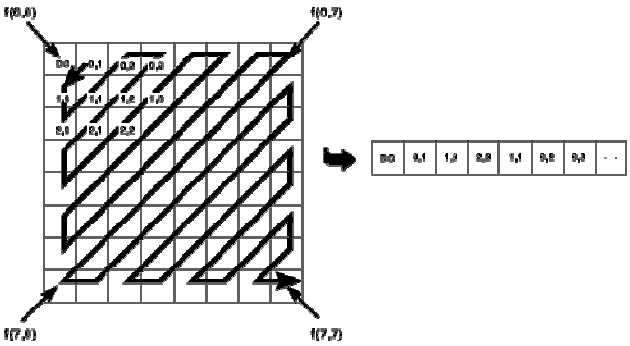


Figure 4 Zig-Zag Scanning

3) Embed the binary watermark into every DC coefficients' last two digits as Figure 2; and
4) Then the quantization and coding process will be finished.

3.2 Decoder

The following four steps explain how to obtain watermark information from the MPEG-2 picture frames.

1) Perform the same Zig-Zag scan as achieved by the encoder in section 3.1 step 3,
2) Obtain every DC coefficients' last 2 digits and put them in a 128 characters String,
3) Transfer the one-zero string into 128 bits code; and
4) The code extracted from (3) is the watermark information.

4 The New Header Protection System

This paper has proposed a new algorithm to detect MPEG-2 picture header error using the MD5 hash function and watermark information embedding technique. This algorithm consists of two implementations:

(1) The picture header was protected by the MD5 hash technique. Any change in the header level will be detected immediately and this is the most significant feature of this new hybrid algorithm when compared with other detection and recovery algorithms; and
(2) The information embedding in the blocks can be used to judge which block had been corrupted when the MPEG-2 video had been decoded.

The following block diagram describes how the new hybrid approach can protect the MPEG-2 video header.

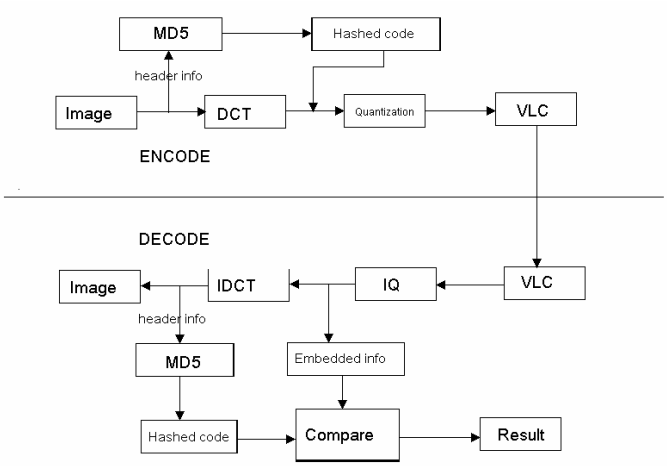


Figure 5 The block diagram of the new hybrid approach to protect the MPEG2 header

According to Figure 5, the header information was extracted from the images and passed through MD5 hash function to generate a 128-bits hash code. Then the codes were embedded into each 8*8 block of all I-frames after DCT transformation.

The frames underwent quantization and VLC functions were encoded as MPEG-2 video. The MPEG-2 video travelled through a noisy channel and reached the decoder. During this period some errors may occur on the video file.

At the decoder the MPEG-2 video was decoded to 300 ppm frames. The header information was hashed into 128-bits code. When this is compared with the embedding information the error or the lost blocks will be detected and located immediately.

5 Experimental Results

Section 4 has explained the new header protection system based on the MD5 hash function and information embedding technique. We tested this system by embedding the watermark information into DC and different AC coefficients. The results in Figure 6 show that we can only embed the watermark information into DC coefficient. The watermarked video I-frames with the AC coefficient changed will corrupt the video pictures as shown in column 3 and column 4 of Figure 6.

We have tested this watermark information embedding system by modifying 1 bit to 4 bits of each 8 x 8 pixels sub-block of the MPEG2 video I-frame pictures. The results are shown in Figure 7. The watermarked host pictures achieve the best result if we only modify 1-bit information in each 8x8 pixels sub-block and have the worst result if we modify 4-bits information in each 8x8 pixel sub-block. However the visual quality is good if we embed 2 bit information into each 8x8 pixels sub-block.

The testing results also show that the performance of this new hybrid approach can detect 100% of header errors.

6 Conclusion

Some research work had been done on MPEG-2 error detection, correction and recovery. However most studies were at the block level. Their methods are only able to achieve good performance on block recovery by using neighbor block information. There are still many things need to be improved to protect the picture header information satisfactorily. In this paper a new approach was proposed and implemented to detect the MPEG-2 header errors using the MD5 hash function and information hiding technique. The video picture headers were protected by the MD5 hash technique and any change in the header level was detected immediately. The information hiding technique was used to locate the error block number in the corrupted MPEG-2 video. The results in section 5 showed that the new approach can detect 100% header error on block level and at higher levels without increasing the transmission bit rate.

References:

[1] Pascal Frossard: “AMISP: A Complete Content-Based MPEG-2 Error-Resilient Scheme”, IEEE transactions on circuits and systems for video technology, Vol. 11, No. 9, pp. 989-998 Sep 2001.

[2] Delicado. F: “Error resilient in MPEG-2 video transmission over wireless ATM networks”, IEEE Conference on High Performance Switching and Routing, pp. 343-351, June 2000.

[3] Cuenca, P: “An efficient protocol architecture for error-resilient MPEG-2 video communications over ATM networks”, IEEE Transactions on Broadcasting, Volume: 45, Issue: 1, pp 129-140, Mar 1999.

[4] Salama, P: “Error concealment in MPEG video streams over ATM networks”, IEEE Journal on Selected Areas in Communications, Volume 18, Issue 6, pp. 1129–1144, June 2000

[5] W. Luo and M. El Zarki: “Analysis of error concealment schemes for MPEG-2 video transmission over ATM based networks”, in Proc. SPIE Conf. Vis. Communication Image Processing Taipei, Taiwan, vol. 1605, pp. 1358-1368, May 1995.

[6] B. Prabhakaran and Q. Qiang: “A forward error recovery technique for real-time MPEG-2 video transport and its performance over wireless IEEE 802.11 LAN”, Ninth International Conference on Computer Communications and Networks, pp 497-502, Oct 2000.

[7] Aign. S: “Error concealment improvements for MPEG-2 using enhanced error detection and early re-synchronization”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol 4, pp 2625-2628 ,Apr 1997.

[8] Chung-Lin Huang; Sling Liang: “A model-driven joint source and channel coder for MPEG-2 video transmission”, International Conference on Information Technology: Coding and Computing, pp 404-409. Apr 2002.

The first five I-frames decoded from the watermarked MPEG2 video when we embed the watermark information into DC and different AC coefficients.

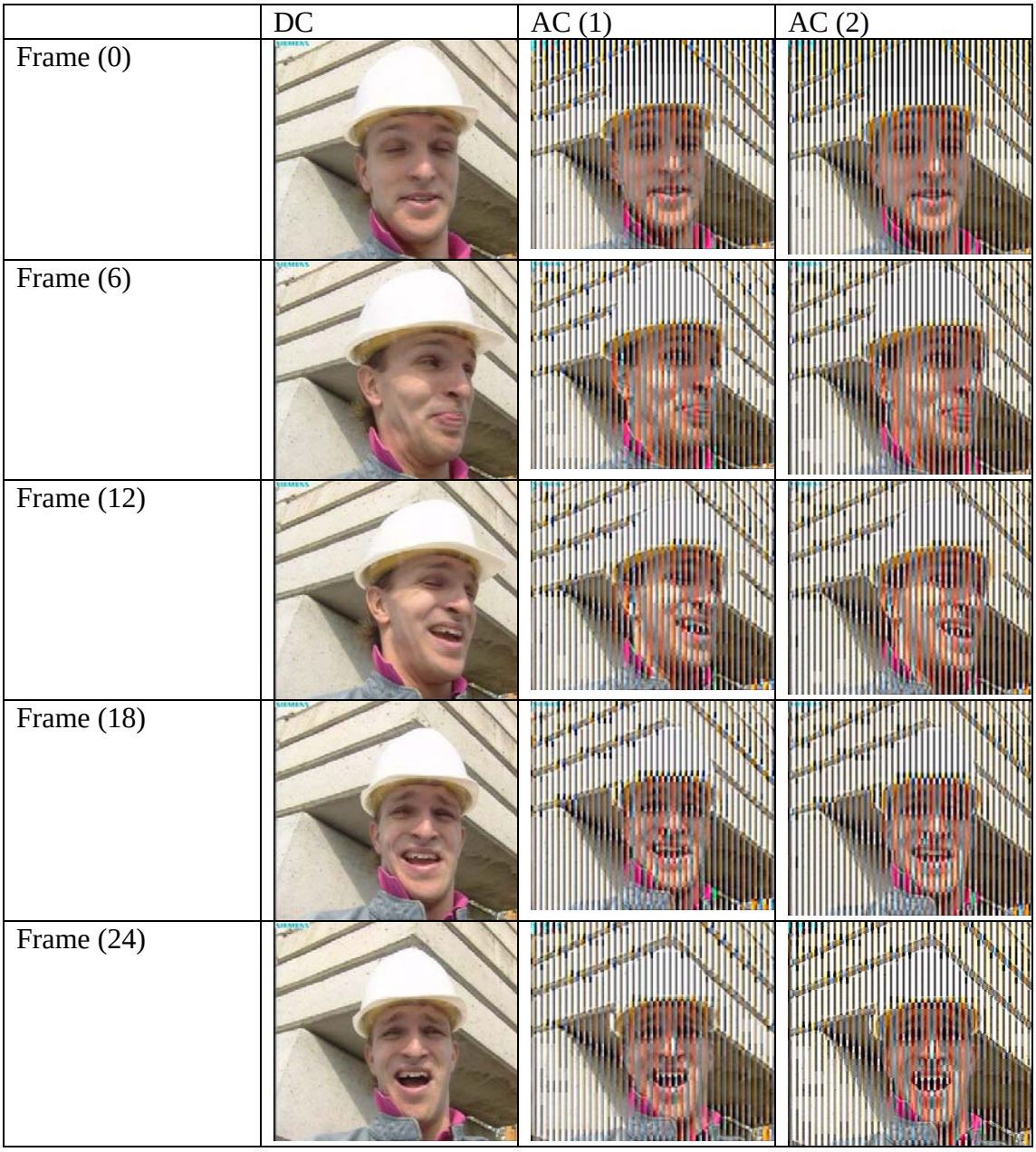


Figure 6 Testing results for embedding information into DC and different AC coefficients

The first six I-frames decoded from the watermarked MPEG2 video when we embed the watermark information by modifying 1 bit to 4 bit of DC coefficients in each 8 x 8 pixels sub-block.

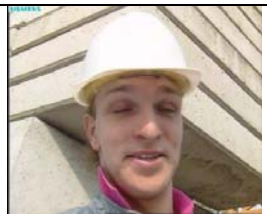
	1 bit	2 bit	3 bit	4 bit
Frame (0)				
	PSNR = 43.94dB MSE = 2.63	PSNR = 41.64dB MSE = 5.09	PSNR = 37.19dB MSE = 12.41	PSNR = 31.60dB MSE = 45
Frame (6)				
	PSNR = 43.47dB MSE = 2.92	PSNR = 41.46dB MSE = 4.65	PSNR = 37.62dB MSE = 11.25	PSNR = 31.65dB MSE = 44.98
Frame (12)				
	PSNR = 43.66dB MSE = 2.80	PSNR = 41.07dB MSE = 5.09	PSNR = 37.15dB MSE = 12.54	PSNR = 32.26dB MSE = 38.61
Frame (18)				
	PSNR = 43.30dB MSE = 3.04	PSNR = 41.47dB MSE = 4.65	PSNR = 37.83dB MSE = 10.72	PSNR = 31.78dB MSE = 43.21
Frame (24)				
	PSNR = 43.71dB MSE = 2.773	PSNR =41.31dB MSE = 4.81	PSNR =37.37dB MSE = 11.91	PSNR = 32.01dB MSE =40.96
Frame (30)				
	PSNR = 43.50dB MSE = 2.90	PSNR = 41.06dB MSE = 5.10	PSNR = 37.47dB MSE = 11.63	PSNR = 31.85dB MSE = 42.51

Figure 7 Testing results for embedding information by modifying 1 bit to 4 bits of DC coefficient