COMBINED CONCEALMENT, SYNCHRONIZATION, AND ERROR DETECTION USING DATA HIDING

A. YILMAZ[1], E. ESEN[1,2] AND A. A. ALATAN[1,2]
[1]*Department of Electrical and Electronics Engineering, M.E.T.U.,*
[2]*TÜBİTAK BİLTEN, Balgat 06531 Ankara, Turkey*
e-mail: {ayhan, alatan}@eee.metu.edu.tr, ersin.esen@bilten.metu.edu.tr

Utilization of data hiding for concealment of video transmission errors provides a reserve information about the video to the receiver while unchanging the transmitted bit-stream syntax; hence, improves the reconstruction video quality without extra channel utilization. A spatial domain error concealment technique, which hides edge orientation information of a block, and a resynchronization technique, which embeds bit length of a block into other blocks are composed. The proposed method also uses these techniques for detecting errors. Simulation results show that these approaches cooperate quite well in concealing the errors.

1. Introduction

In order to handle the transmission errors of digital signals over noisy wireless channels at the receiver side, some error concealment techniques have been proposed [1]. These techniques try to recover the lost data by an interaction between the encoder and decoder, as a re-send signal [1], or post-processing operations to recover lost information at the decoder [5], or leaving some extra redundancy at the encoder to minimize the reconstruction error [1]. Instead of permitting some redundancy during source coding, one option is to hide some imperceptible information to be useful during error concealment.

Data hiding is a new generation technique of making imperceptible modifications on the media [3] and the hidden information can be transmitted without a bit-rate overhead in the bit-stream of the compression standard being used. Hiding information, not only works as a hidden channel from encoder to decoder, but also alleviates some burden on post-processing.

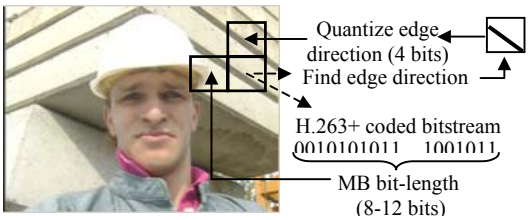
2. Related Work On Error Concealment By Data Hiding

Conventional error concealment methods are achieved in a post-processing stage and are mainly based on the smoothness property for the image frames. Recovery of the lost blocks is usually performed in spatial [5] or frequency domain [1]. However, if a lost block is not sufficiently smooth, then the interpolation of this block from its neighbors is a suboptimal solution. Data

hiding, which simply serves for transporting this lost information to the receiver, can be a good alternative solution with some bottlenecks.

Since interpolation along the edge direction gives superior results compared to conventional approaches, edge direction information is worth to hide [5]. In [2], the edge orientation of each block is embedded into a *companion* block (Fig. 1) by the help of any data hiding technique. Another problem arising in error concealment is due to the loss of synchronization in the bit stream at the decoder. One solution is to hide the bit length data of a block into neighbors [4].

Figure 1. Embedding edge orientation and bit length



3. Proposed Method

In order to achieve successful error concealment, the exact location of the error, i.e. damaged block, should be detected as a first step. After detecting the damaged block, synchronization must be established back in order to prevent the propagation of the error to the other blocks. The next step is the reconstruction of the intensities for the damaged block.

In the proposed method, edge direction information and bit length data are necessary to realize the error recovery in all 3 issues. Edge direction information is embedded into one upper block of the current block and bit length data is hidden into the previous block. For embedding the edge orientation, the block is first classified as an *edge block* by computing the gradient magnitude. The angles, whose gradient magnitudes are above a threshold, are quantized into 16 equally spaced directions (4 bits). Obviously, a single message bit should also be hidden to indicate the type of the block, i.e. an edge or a smooth block (Fig.1). For hiding the bit-length data, the number of bits used for the current block is determined during encoding and this value is embedded after a conversion into binary representation (Fig.1). The proposed method requires 8 to 12 bits according to the bit rate (i.e. quantization parameter). Data hiding is achieved by simple “even-odd” signaling [2,4].

3.1. *Error Detection*

Both edge direction information and bit length data is used for error detection. For each decoded block, its edge direction information is calculated once again

at the decoder and compared with the information hidden in the upper block. After the edge direction test, the bit length data is checked as second stage. After a block is decoded, total number of bits read from the bit stream and the value hidden in the previous block are compared.

3.2. *Resynchronization*

In order to resynchronize at the decoder, the bit length data is utilized again. The difference between the hidden and decoded bit numbers is calculated and the decoder skips that amount of bits in order to start decoding from a new undamaged block. In this way, without having macroblock headers, the system is able to synchronize itself at the start of each macroblock.

3.3. *Reconstruction*

For every block, edge direction information is extracted from the blocks in the upper slice. When an error is detected in a block, its edge direction information is checked whether it is an edge or a smooth block. If it is an edge block, then it is interpolated from two neighboring blocks along its edge direction. Otherwise, for a smooth block, simple bilinear interpolation technique is applied.

4. Simulations

ITU H.263+ codec is utilized for *Foreman* sequence during the experiments. After H.263+ compression, single bit errors are imposed on the resultant bit-stream. It should be noted that the current system considers only errors that affect the intra frames. A system, considers inter frames by hiding motion vector information, is still under development.

In Fig. 2(a), (b) and (c), the first frames of the original, compressed and data hidden videos are shown respectively. In Fig. 2(d), a typical result due to a random 1-bit error in the bit stream (intra-frame) is given. The concealment in Fig. 2(e) is made by bilinear interpolation only and H.263+ decoder itself detects the error. Fig. 2(f) is concealed by the proposed method. The decoder perfectly detects the error, synchronizes the next block and reconstructs the damaged block. Table 1 summarized the average PSNR values for the whole (22 times single-bit-error) experiments.

5. Conclusions

In this paper, a novel error concealment method using data hiding is proposed. The system combines two previous methods in this area in order to obtain much better reconstruction quality considering detection, resynchronization and

concealment together. However, all blocks do not have the same characteristics from concealment point of view. The simulations show that the blocks without a major single edge (such as highly textured areas) cannot be interpolated successfully via edge-interpolation.

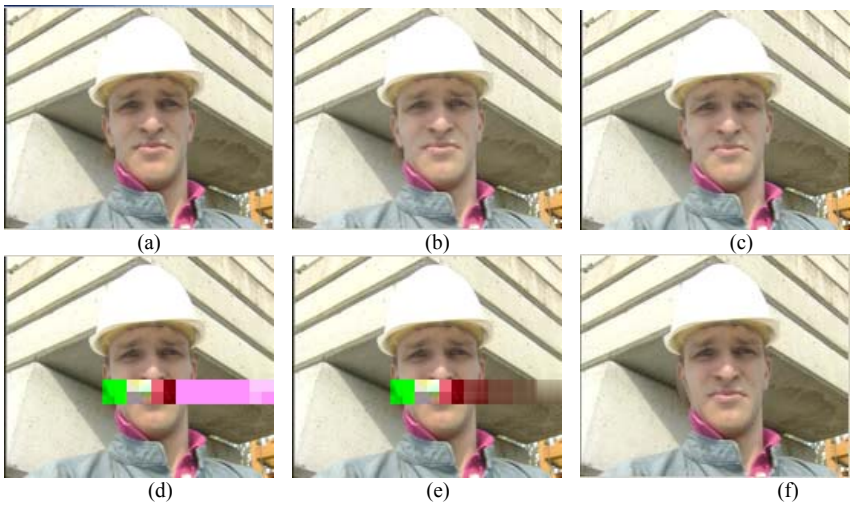


Figure 2. (a): original, (b) compressed, (c) data hidden, (d) 1-bit error, (e) bilinear interpolation, (f) proposed method.

Table 1. Average PSNR values.

Average PSNR (dB)	**Y**	**U**	**V**
After H.263+ compression (900 Kbit/s)	43.58	44.70	46.72
After data hiding	42.93	44.47	45.76
After transmission with errors	29.08	29.52	31.02
After concealment using bilinear interpolation	32.81	36.70	37.41
After concealment using edge based interpolation	41.38	44.38	44.88

References

1. Y. Wang and Q.-F. Zhu, *Proc. of the IEEE* **86**, 974 (1998).
2. P. Yin, B. Liu, and H. H. Yu, *Proc. of. ICASSP* **3**, 1453 (2001).
3. S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House (2000).
4. D. L. Robie and R. M. Mersereau, *Proc. of ICIP*, 930 (2001).
5. W. Zeng, and B. Liu, *IEEE Transactions on Circuits and Systems for Video Technology* **9**, 648 (1999).