DATA HIDING USING BIT- STREAM LEVEL THROUGH SELECTIVE EMBEDDING AND MODULATION

Venkata Pavan Kumar M[1], Dr. C.P.V.N.J. Mohan Rao[2], Satya P Kumar Somayajula[3]

1Final Mtech Student, Avanthi Institute of Engineering and Technology
2Principal, Avanthi Institute of Engineering and Technology
3Assistant Professor, Avanthi Institute of Engineering and Technology

Abstract

Hiding the data in the Video is still an important research issue,but the problem with the traditional mechanisms it leads to the distortion and error prone. However, most of the video data hiding methods utilize uncompressed video data. Recent video data hiding techniques are focused on the characteristics generated by video compressing standards. Motion vector based schemes have been proposed for MPEG Algorithms. In this paper we present a new data hiding scheme for H.264, MPEG encoded video sequences. Embedding takes place during the encoding process and utilizes the advanced inter prediction features of the encoder. The simulation results show that the proposed approach performs superior to conventional approaches for concealing the errors in binary symmetric channels, especially for higher bit rates and error rates.

Index Terms: *Data hiding, forbidden zone data hiding, quantization index modulation, repeat accumulate codes, selective embedding, synchronization, error concealment.*

---------- *** ----------

1. INTRODUCTION

Data hiding has been used in various applications like copyright protection, authentication, fingerprinting, error concealment, broadcast monitoring, covert communication, etc. Each application imposes different types of constraints in terms of capacity, security and robustness. Privacy is protected by obfuscating images of individuals from the video and the original data is preserved by hiding it in the compressed bit stream of the modified video. This is particularly useful when a condition arises to prove the authenticity of the modified video. In general, visual and arual media are preferred due to their wide presence and the tolerance of human perceptual systems involved. Although the general structure of data hiding process does not depend on the media type. The transmission of video signals over noisy wireless channels may cause inevitable errors that might severely degrade the visual message. In wireless communication systems, in order to handle such errors, some error concealment techniques have been proposed in three major groups. These techniques try to recover the lost data either by an interaction between the encoder and decoder, as a re-send signal, or post-processing operations at the decoder to recover lost information, or leaving some extra redundancy at the encoder to minimize the reconstruction error.

In the encoder and decoder interactive error concealment techniques, encoder and decoder cooperate, if a backward channel from decoder to encoder is available. Based on the feedback information, source coding parameters, the amount of Forward Error Correction (FEC) bits, and retransmission bandwidth can be changed. However, re-transmission leads to decoding delays, which is not desirable in some real-time systems.

Post-processing error concealment techniques use the correlation between the damaged block and its neighboring blocks in the same frame and/or previous frame. These techniques are based on the smooth variation of the intensity values of spatial and temporally adjacent pixels. However, in the regions with sharp edges, a satisfying reconstruction may not be achieved by post-processing operations.

The error concealment techniques in the third group utilize the redundant data on the bit stream, which is added at the encoder side after source coding. Video source can be coded in layers or in multiple descriptions during the source coding and some amount of FEC can be applied. The major drawback of these methods is the increasing transmission overhead.

All these approaches can be merged together by hiding some imperceptible information to be useful during error

concealment. During source coding, some information about the video can be embedded into certain parts of the video itself and the decoder can make use of this hidden information in error concealment. In this way, hidden information is not only transmitted through a secret channel from encoder to decoder by "sending back" some lost information, but also alleviates some burden on post-processing. Hiding some data into a video slightly degrades the visual quality and causes a minor increase in the coding bit rate. On the other hand, the extra hidden information and its small visual loss might be equivalent to decreasing the source bit-rate for obtaining the same visual quality and utilizing error control codes as a result of the bit savings at the encoder.

Data hiding in video sequences is performed in two major ways: bit stream-level and data-level. In bit stream-level, the redundancies within the current compression standards are exploited. Typically, encoders have various options during encoding and this freedom of selection is suitable for manipulation with the aim of data hiding. However, these methods highly rely on the structure of the bit stream; hence, they are quite fragile, in the sense that in many cases they cannot survive any format conversion or transcoding, even without any significant loss of perceptual quality. As a result, this type of data hiding methods is generally proposed for fragile applications, such as authentication. On the other hand, data level methods are more robust to attacks. Therefore, they are suitable for a broader range of applications.

However, most of the video data hiding methods utilize uncompressed video data. Sarkar et al. Proposed a high volume transform domain data hiding in MPEG-2 videos. They applied quantization index modulation (QIM) to low frequency DCT coefficients and adapted the quantization parameter based on MPEG-2 parameters. Furthermore, they varied the embedding rate depending on the type of the frame. As a result, insertions and erasures occur at the decoder, which causes de-synchronization. They utilized repeat accumulate (RA) codes in order to withstand erasures. Since they adapted the parameters according to type of frame, each frame is processed separately.

RA codes are already applied in image data hiding. In adaptive block selection results in de-synchronization and they utilized RA codes to handle erasures. Insertions and erasures can be also handled by convolutional codes. The authors used convolutional codes at embedder. However, the burden is placed on the decoder. Multiple parallel Viterbi decoders are used to correct de-synchronization errors. However, it is observed that such a scheme is successful when the number of selected host signal samples is much less than the total number of host signals samples.

In this paper we propose a new block – based selective embedding type data hiding framework that encapsulates forbidden zone data hiding (FZDH) and RA codes in accordance with an additional temporal synchronization mechanism. RA codes are already used in image and video data hiding due to their robustness against erasures. This robustness allows handling de-synchronization between embedder and decoder that occurs as a result of the differences in the selected coefficients. In order to incorporate frame synchronization markers, we partition the blocks into two groups. One group is used for frame marker embedding and the other is used for message bits. By means of simple rules applied to the frame markers, we introduce certain level of robustness against frame drop, repeat and insert attacks. We utilize systematic RA codes to encode message bits and frame marker bits. Each bit is associated with a block residing in a group of frames

Random interleaving is performed spatio-temporally; hence, dependency on local characteristics is reduced. Host signal coefficients used for data hiding are selected at four stages. First, frame selection is performed. Frames with sufficient number of blocks are selected. Next, only some predetermined low frequency DCT coefficients are permitted to hide data. Then the average energy of the block is expected to be greater than a predetermined threshold. In the final stage, the energy of each coefficient is compared against another threshold. The unselected blocks are labeled as erasures and they are not processed. For each selected block, there exists variable number of coefficients. These coefficients are used to embed and decode single message bit by employing multidimensional form of FZDH that uses cubic lattice as its base quantizer.

In addition to that, intra- and inter-coded frames are considered separately from the error concealment point of view. For concealing the errors in intra-coded frames, mainly edge direction data of a Micro Block (MB) and coded MB bit length value are used. On the other hand, the coded motion vector bits are utilized to conceal the errors for inter-coded frames. All these data are embedded into the video at the encoder and then extracted at the decoder as an auxiliary data for error concealment.

2. RESTRICTED AREA FOR DATA CONCEALMENT

Restricted Area for Data Concealment is nothing but Forbidden Zone Data Hiding (FZDH) which was introduced in [8]. The method depends on the Forbidden Zone (FZ) concept, which is defined as the host signal range where no alteration is allowed during data hiding process. FZDH makes use of FZ to adjust the robustness-invisibility trade-off.

Let s (bold denoting a vector) be the host signal in R^N and m Є {0, 1} be the data to be hidden. Then the marked signal x is obtained as given in (1).

$$\boldsymbol{x} = \begin{cases} \boldsymbol{s}, & \boldsymbol{s} \in \boldsymbol{FZ_m} \\ \boldsymbol{M_m(s)}, & \boldsymbol{s} \in \boldsymbol{AZ_m} \end{cases} \quad (1)$$

where FZ_m, Allowed Zone (AZ_m) pair defines the host signal zones where alteration is allowed or not and $M_m(.)$ is a mapping from R^N to a suitable partition of R^N. The requirement on these zones and partitions is simply based on the constraint that they should be mutually exclusive for different m. The key point of FZDH is the determination of the zones and the partitions.

There could be infinite ways to achieve this; however, a practical design can be performed by using quantizers. Such a simple parametric form is given in (2),here the mapping function is defined as:

$$M_m(s) = \{s + e_m(1 - \frac{r}{e_m}) \quad (2)$$

Here r is the control parameter, $q_m(.)$ is a quantizer indexed by m and e is defined as the difference vector between the host signal and its quantized version:

$$e_m \triangleq Q_m(s) - s \quad \textbf{(3)}$$

The mapping function in (2) states that the host signal is modified by adding an additional term, which is a scaled version of the quantization difference. In 1-D, this additional term is scalar, whereas in N-D host signal is moved along the quantization difference vector and towards the reconstruction point of the quantizer. Hence, embedding distortion is reduced and became smaller than the quantization error.

FZ_m and AZ_m are defined using the control parameter and the difference vector:

$$FZ_m = \{s \mid \|e_m\| \leq r\},\ AZ_m = \{s \mid \|e_m\| > r\} \quad \textbf{(4)}$$

In order to fulfill the requirement of mutual exclusion, the reconstruction points of the quantizers that are indexed by different m should be non-overlapping, which can be achieved by using a base quantizer and shifting its reconstruction points depending on m, similar to Dither Modulation [10]. A typical embedding function that uses a uniform quantizer is shown in Fig. 1.

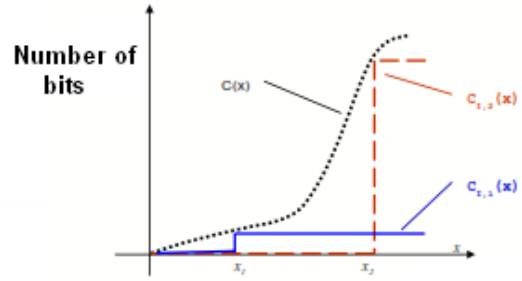

Fig. 1. A sample embedding function of FZDH in 1D. c_i is a reconstruction point of the quantizer.

During data extraction step, the generic minimum distance decoder is utilized to decode the hidden data:

$$\hat{m} = arg.min\ d(y, y_m) \quad (5)$$

where y is the received signal, y_m is equal to its FZDH embedding operation applied version as in (1), and d(.,.) is a suitable distance metric.

The decoder and embedder should be synchronized in terms of the zones, partitions and system parameters.

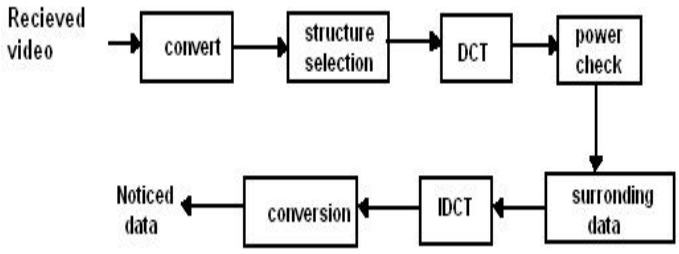

Fig. 2. Embedder flowchart of the proposed video data hiding framework for a single frame.

2. PROPOSED VIDEO DATA HIDING FRAMEWORK

We propose a block based adaptive video data hiding method that incorporates FZDH, which is shown to be superior to QIM and competitive with DC-QIM[9], and erasure handling through RA Codes. We utilize selective embedding to determine which host signal coefficients will be used in data hiding as in [3] and [4]. We employ block selection (Entropy Selection Scheme) and coefficient selection (Selectively Embedding in Coefficients Scheme[5]) together. The de-synchronization due to block selection is handled via RA Codes. The de-synchronization due to coefficient selection is handled by using multi-dimensional form of FZDH in varying dimensions. The frames are processed independently.

It is observed that intra and inter frames do not yield significant differences. Therefore, in order to overcome local bursts of error, we utilize 3-D interleaving, which does not utilize selective embedding, but uses the whole LL subband of Discrete Wavelet Transform. Furthermore, as in [6], we equip the method with frame synchronization markers in order to handle frame drop, insert or repeat attacks.

Hence, it can be stated the original contribution of this work is to devise a complete video data hiding method that is resistant to de-synchronization due to selective embedding and robust to temporal attacks, while making use of the superiority of FZDH as in [1].

A. Framework

The embedding operation for a single frame is shown in Fig. 2. Y-channel is utilized for data embedding. In the first step, frame selection is performed and the selected frames are processed block-wise. For each block, only a single bit is hidden. After obtaining 8x8 DCT of the block, energy check is performed on the coefficients that are predefined in a mask. Selected coefficients of variable length are used to hide data bit m. m is a member of message bits or frame synchronization markers. Message sequence of each group is obtained by using RA codes for T consecutive frames. Each block is assigned to one of these groups at the beginning. After the inverse transform host frame is obtained

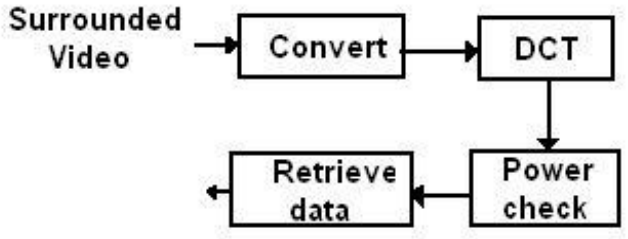

Fig. 3. Decoder flowchart of the proposed video data hiding framework for a single frame

Decoder is the dual of the embedder, with the exception that frame selection is not performed. Fig. 3 shows the flowchart for a single frame. Marked frames are detected by using frame synchronization markers. Decoder employs the same system parameters and determines the marked signal values that will be fed to data extraction step. Non-selected blocks are handled as erasures. Erasures and decoded message data probabilities (o_m) are passed to RA decoder for T consecutive frames as a whole and then the hidden data is decoded.

B. Selective Embedding

- Host signal samples, which will be used in data hiding, are determined adaptively. The selection is performed at four stages: frame selection, frequency band determination, block selection, and coefficient selection.
- Frame Selection: Selected number of blocks in the frame is counted. If the ratio of selected blocks to all the blocks is above to certain value (T_0) the frame is processed. Otherwise this frame is skipped.
- Frequency Band: Only certain DCT coefficients are utilized. Middle frequency band of DCT coefficients are shown in Fig. 4 is utilized is similar to [3].
- Block Selection: Energy of the coefficients in the mask is computed. If the energy of the block is above the certain level (T_1) then the block is processed. Otherwise it is skipped.
- Coefficient Selection: Energy of each coefficient is compared to other threshold (T_2). If the energy of the above T_2, then it is used during data embedding together with other selected coefficients in the same block.

C. Block Partitioning

Two disjoint data sets are embedded: message bits (m_1) and frame synchronization markers (m_2). The block locations of m_2 are determined randomly depending on a random key. The rest of the blocks are reserved for m_1. The same partitioning is used for all frames. m_2 is embedded frame by frame.
On the other hand, m_1 is dispersed to T consecutive frames. Both of them are obtained as the outcomes of the RA encoder.

D. Erasure Handling

Due to adaptive block selection, de-synchronization occurs between embedder and decoder. As a result of attacks or even embedding operation decoder may not perfectly determine the selected blocks at the embedder. In order to overcome this problem, error correction codes resilient to erasures, such as RA codes are used in image[4] and video[3] data hiding in previous efforts.

RA code is a low complexity turbo-like code[12]. It is composed of repetition code, interleaver and a convolutional encoder. The source bits (u) are repeated R times and randomly permuted depending on a key. The interleaved sequence is passed through a convolutional encoder with a transfer function 1/(1+D), where D represents a first order delay.

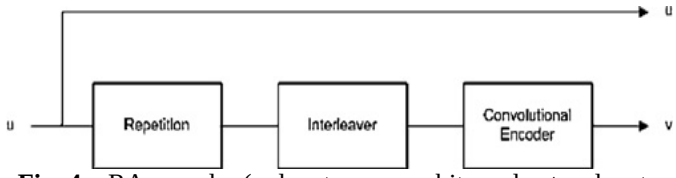

Fig. 4. RA encoder (u denotes source bits and u + v denote encoded bits

In systematic RA code, input is placed at the beginning of the output as shown in Fig. 4. In this work, we utilize systematic RA codes to obtain m_1 as u_1+v_1 and m_2 as u_2+v_2. Here, u_1 denotes the uncoded message bits and u_2 is the uncoded frame synchronization marker bits. RA code is decoded using sum-product algorithm

E. Frame Synchronization Markers

Each frame within a group of T consecutive frames is assigned a local frame index starting from 0 to T-1. These markers are used to determine the frame drops, inserts and repeats, as well as the end of the group of frames at which point all necessary message bits are available for RA decoder.

Frame indices are represented by K_2 bits. After RA encoder RK_2 bits are obtained. Hence, RK_2 blocks are reserved for frame markers. $K_2 >> \log_2 T$, so that a small 2^{k2} portion of codewords is valid. Therefore, we can detect the valid frames with higher probability. Using the sequential frame index information, the robustness increases. Furthermore, RA code spreads the output codewords of the adjacent frame indices; hence, errors are less likely to occur when decoding adjacent frame indices.Once one reserves RK_2 blocks for frame markers, $T(N-RK_2)$ blocks remain for message bits.

F. Soft Decoding

At the decoder, a data structure of length RK_1 is kept for channel observation probability values, o_m. The structure is initialized with erasures (o_m= 0.5 for m=0 and m=1). At each frame, frame synchronization markers are decoded first. Message decoding is performed once the end of the group of frames is detected.

Two frame index values are stored: current and previous indices. Let f_{cur} and f_{pre} denote the current and previous frame indices, respectively. Then the following rules are used to decode u_1.

If $f_{cur} > T$, then skip this frame. (This case corresponds to unmarked frame).

If $f_{cur} = f_{pre}$, then, skip this frame. (This case corresponds to frruame repeat).

Otherwise, process the current frame. Put O_m values in the corresponding place of the data structure. Non-selected blocks are left as erasures.

If $f_{cur} < f_{pre}$, then the group of the frames is reached. Decode the message bits and initialize the data structure.

G. Intra-frame Error Concealment

In order to achieve a successful error recovery, 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 final step is the reconstruction of the intensities for the damaged block to finalize error recovery. Therefore, the three main issues for a successful error recovery are error detection, resynchronization and reconstruction (recovery) of the damaged block. In the remaining parts of this section, the proposed system is briefly explained. While bit-length value is strictly necessary for synchronization, edge direction information is suitable for the reconstruction of the damaged block. Finally, these two data can be used together to detect the bit-errors.

However, the reconstruction of all the damaged blocks does not always give promising results, which causes an over concealment case. For determining such cases and deciding for the recovery of a damaged block, a 2-bit over concealment parity, which is obtained from DCT coefficients of the block, is proposed to accompany the edge direction information. However, the hidden data is not capable of detecting all bit errors. In order to provide a full detection capability, a single parity bit is proposed to use with the synchronization data.

In order to embed edge orientation, the block is first classified as an edge block by applying an edge detection algorithm. For each pixel in the block, its gradient vector magnitude and gradient vector angle are calculated by using Robert gradient operator.

The angles of the pixels, whose gradient magnitudes are above a threshold, are quantized into 16 equally spaced directions (i.e. represented with 4 bits) and the gradient magnitudes with the same direction are summed up. The direction with largest gradient magnitude sum is selected as the final single edge direction of the whole block (Fig. 5). Obviously, a single message bit should also be hidden to indicate the type of the block, i.e. an edge or a smooth block. Hence, this approach requires only 5 bits per block to embed the edge direction information to the DCT coefficients of the upper MB, which is used to recover the intensities of the blocks.

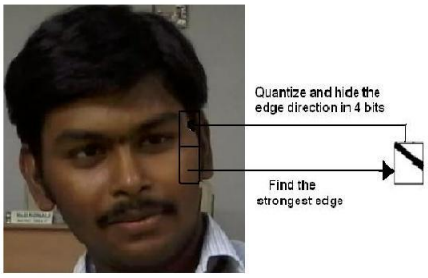

Fig. 5 Hiding edge direction data for intra-frame error concealment.

For hiding bit length data, the number of bits required for encoding the current block is determined and this value is embedded into the DCT coefficients of the previous block after being converted into binary representation during encoding (Fig. 6). The number of bits used in this representation should be pre-calculated by considering the maximum bit lengths of typical blocks. The proposed method requires 9 to 13 bits (i.e. for each block, bit length value varies from 511 to 8191) depending on the bit rate of the utilized video encoder (i.e. quantization parameter). By looking at the quantization parameter, decoder can determine, the number of bits used for the bit length data.

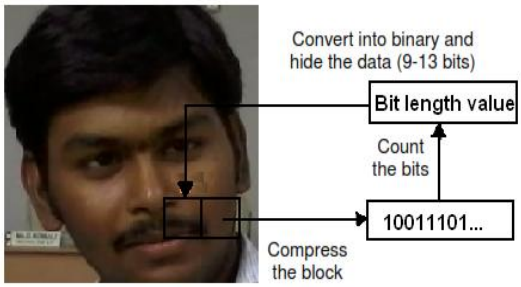

Fig. 6. Hiding bit length value for intra-frame error concealment.

i) Error Detection Using Parity:

In case of consistency in the bit length values, a single parity bit of the macroblock bit-stream is used in error detection. This parity bit is obtained by taking XOR of all the bits for the coded macroblock and hidden into previous block's DCT coefficients as extra hidden information (Fig. 6). If the bit length value check and parity bit check do not give any error, then one can be sure that there is no error in the decoded macroblock and the data hidden in it, as well. Obviously, this is based on an assumption that there is a single bit error in the bit stream of the current block. This assumption can be tolerable in practice, not only for binary symmetric, but also even for fading channels, if appropriate interleaving of the bitstream can be achieved.

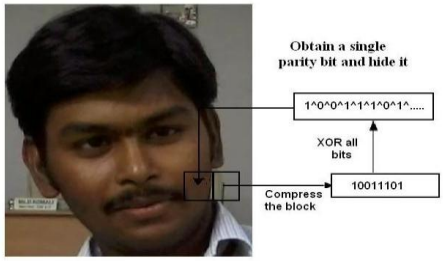

Fig. 6 Hiding a single parity bit for intra-frame error detection

ii) Resynchronization:

After error detection, in order to resynchronize at the decoder, simply the bit length data is utilized. During decoding, the system is not allowed to decode bits more than the number that is dictated as the hidden value in the previous block. At the time of resynchronization, it is certain that the number of decoded bits is smaller or equal to the hidden value in the previous block. The difference between the hidden and decoded bit numbers is calculated and the decoder skips the calculated amount of bits, in order to start decoding from the next undamaged block (Fig.7). In this way, without having macroblock headers that can guarantee synchronization, the system is able to synchronize itself at the start of each macroblock**.**

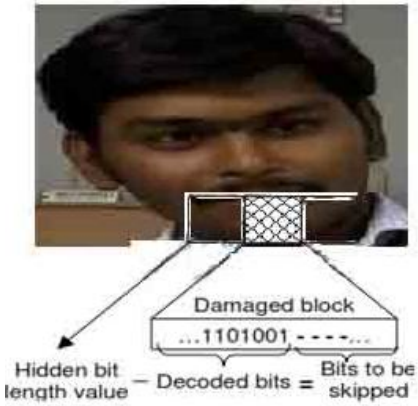

Fig. 7. Resynchronizing the decoder to the next macroblock.

iii) Reconstruction:

The importance of interpolation along edge direction is illustrated in Fig.8. In Fig. 8, the center block contains a strong edge with four neighboring blocks and then, this edge block is damaged in Fig. 10 (b). Afterwards, the damaged block is reconstructed by bilinear and edge directed interpolation techniques in (c) and (d), respectively. The results show intuitively the superiority of the edge-based interpolation, especially for the blocks with a major edge.

A. As soon as the error is detected and the synchronization is obtained, the final step is to recover the single block in which an error has occurred. For this purpose, edge direction information is extracted from the blocks in the upper slice for every block (note that the edge direction information for the blocks of the first slice is hidden into the blocks of the last slice). The first hidden bit, which indicates the type of the lost block, is tested to check whether it is an edge or a smooth block. If it is found out to be an edge block, then it is interpolated from two neighboring blocks along its edge direction (Fig. 11). Otherwise, for a smooth block, simple bilinear interpolation technique is applied.

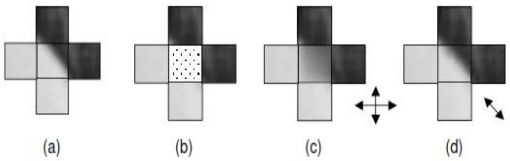

Fig. 8. Interpolation along edge direction: (a) an edge block with four neighboring blocks, (b) damaged block, (c) result after bilinear interpolation, (d) result after interpolation along edge direction.

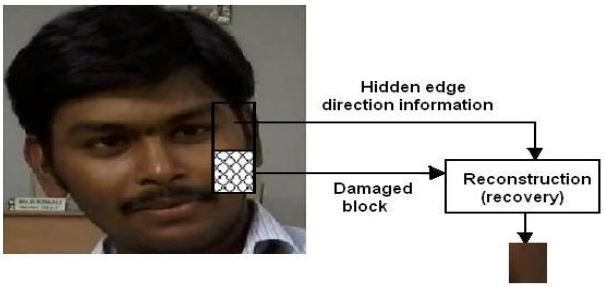

Fig. 9. Reconstruction of the damaged block in intra-frame errors.

iv) Overall System:

All the necessary information for intra-frame error concealment can be seen in Fig. 10. While the bit length, block parity, and overconcealment bits of each block are hidden into its previous block on the left of current block, the edge direction data is embedded into its upper block. All these data are concatenated and a short bit stream is obtained. Finally, this bit stream is hidden into the neighbor block (Fig.10).

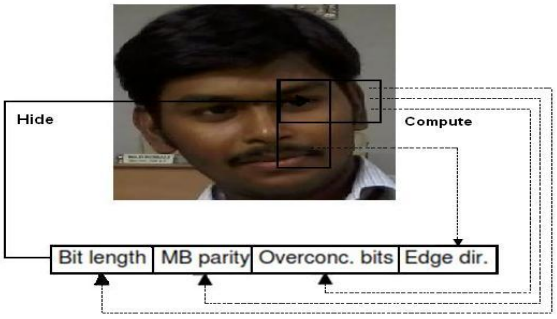

Fig. 10. Obtaining and hiding all the necessary bits for intra-frame error concealment.

CONCLUSION

In this paper, we propose a new video data hiding framework that makes use of erasure correction capability of RA codes, superiority of FZDH and Intra-Frame error concealment technique. The method is also robust to frame manipulation attacks via frame synchronization markers.

First, we compare FZDH and QIM as the data hiding method of the proposed framework. We observe that FZDH is superior to QIM, especially for low embedding distortion levels.

The framework is tested with MPEG-2, H.264 compression, scaling and frame-rate conversion attacks. Typical system parameters are reported for error-free decoding. The results indicate that the framework can be successfully utilized in video data hiding applications. For instance, Tardos fingerprinting, which is a randomized construction of binary fingerprint codes that are optimal against collusion attack, can be employed within the proposed framework with the following settings. We also compared the proposed framework against the canonical watermarking method, JAWS [15] and [16], and a more recent quantization based method in [3]. The results indicate a significant superiority over JAWS and a comparable performance.

The experiments also shed light on possible improvements on the proposed method. Firstly, the framework involves a number of thresholds (T_0, T_1, and T_2), which are determined manually. The range of these thresholds can be analyzed by using a training set. Then some heuristics can be deduced for proper selection of these threshold values.

Additionally, incorporation of Human Visual System based spatio-temporally adaptation of data hiding method parameters remains as a future direction.

Intra-coded frames are considered separately from the error concealment point of view, For concealing the errors in intra-coded frames, mainly edge direction data of an MB and coded MB bit length value are used. On the other hand, the coded motion vector bits are utilized to conceal the errors for inter-coded frames. All these data are embedded into the video at the encoder and then extracted at the decoder as an auxiliary data for error concealment.

REFERENCES

[1] Ersin Esen and A. Aydin Alatan, "Robust Video Data Hiding Using Forbidden Zone Data Hiding and Selective Embedding", in IEEE transactions on Circuits and Systems for Video Technology, vol.21, NO. 8, Aug 2011

[2] S. K. Kapotas, E. E. Varsaki, and A. N. Skodras, "Data Hiding in H-264 Encoded Video Sequences," in IEEE 9th Workshop on Multimedia Signal Processing, MMSP 2007, pp. 373—376.

[3] A. Sarkar, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Adaptive MPEG-2 Video Data Hiding Scheme," in Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX, 2007.

[4] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, , and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," IEEE Transactions on Image Processing, vol.13, Dec. 2004, pp. 1627--1639.

[5] M. Schlauweg, D. Profrock, and E. Muller, "Correction of Insertions and Deletions in Selective Watermarking," in IEEE International Conference on Signal Image Technology and Internet Based Systems, SITIS '08, 2008, pp.277—284.

[6] H.Liu, J.Huang, and Y. Q. Shi, "DWT-Based Video Data Hiding Robust to MPEG Compression and Frame Loss," Int. Journal of Image and Graphics, vol. 5, pp. 111-134, Jan. 2005.

[7] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video I. Fundamental issues and solutions," IEEE Transactions on Image Processing, vol. 12, pp. 685—695, June 2003.

[8] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video II: Designs and applications," IEEE Transactions on Image Processing, vol. 12, pp. 696—705, June 2003.

[9] E. Esen and A. A. Alatan, "Forbidden zone data hiding," in IEEE International Conference on Image Processing, 2006, pp. 1393— 1396.

[10] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," IEEE Transactions on Information Theory, vol. 47, May 2001, pp. 1423-1443, May 2001,.

[11] E. Esen, Z. Doğan, T. K. Ates, and A. A. Alatan, "Comparison of Quantization Index Modulation and Forbidden Zone Data Hiding for Compressed Domain Video Data Hiding,"in IEEE 17th Signal Processing and Communications Applications Conference SIU, 2009.

[11] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo-like codes," in Proc. 36th Allerton Conf. Communications, Control, and Computing, 1998, pp. 201—210.

[12] M. M. Mansour, "A Turbo-Decoding Message-Passing Algorithm for Sparse Parity-Check Matrix Codes," IEEE Transactions on Signal Processing, vol. 54, pp. 4376—4392, Nov. 2006.

[13] Z. Wei, K. N. Ngan, "Spatio-Temporal Just Noticeable Distortion Profile for Grey Scale Image/Video in DCT Domain," IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, pp. 337—346, Mar. 2009.

[14] M. Maes, T. Kalker, J. Haitsma, and G. Depovere, "Exploiting Shift Invariance to Obtain a High Payload in Digital Image Watermarking," in IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), vol. 1, 1999.

[15] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes, "Video watermarking system for broadcast monitoring," in Security and watermarking of multimedia contents Conference, SPIE Proceedings vol. 3657 , 1999, pp. 103—112.

[16] M. Maes, T. Kalker, J. -P. M. G., J. Talstra, F. G. Depovere, and J. Haitsma, "Digital watermarking for DVD video copy protection," IEEE Signal Processing Magazine, vol. 17, pp. 47—57, Sep. 2000.

[17] K. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, "Complete video quality-preserving data hiding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, pp. 1499—1512, Oct. 2009.

[18] G. Tardos, "Optimal probabilistic fingerprint codes," in Proceedings of the thirty fifth annual ACM symposium on Theory of computing (STOC '03), New York, NY, USA, 116—125.

[19] B. Skoric, T. U. Vladimirova, M. Celik, and J. C. Talstra, "Tardos fingerprinting is better than we thought," IEEE Transactions on Information Theory, vol. 54, no. 8, pp. 3663—3676, 2008

BIOGRAPHIES

Mr. M. V. Pavan Kumar, completed B-Tech(CSE) in Pydah College of Engineering and Technology, Visakhapatnam. Later he is studying M-Tech (Software Engineering) in Avanthi Institute of Engineering and Technology. His interested areas are Information Security, Image Processing

Dr.C.P.V.N.J Mohan Rao is Professor in the Department of Computer Science and Engineering and Principal of Avanthi Institute of Engineering & Technology - Narsipatnam He did his PhD from Andhra University and his research interests include Image Processing, Networks, Information security, Data Mining and Software Engineering. He has guided more than 50 M.Tech Projects and currently guiding four research scholars for Ph.D. He received many honors and he has been the member for many expert committee member of many professional bodies and Resource person for various organizations.

Mr. Satya P Kumar Somayajula is working as an Asst.Professor, in CSE Department, Avanthi Institute of Engg & Tech, Tamaram, Visakhapatnam, A.P., India. He has received his M.Sc (Physics) from Andhra University, Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM University), Visakhapatnam, A.P., INDIA. He published 17 papers in reputed International journals & 5 National journals. His research interests include Image Processing, Networks security, Web security, Information security, Data Mining and Software Engineering.