Robust Video Data Hiding Using Forbidden Zone Data Hiding and Selective Embedding

Ersin Esen and A. Aydin Alatan, *Member, IEEE*

***Abstract*—Video data hiding is still an important research topic due to the design complexities involved. We propose a new video data hiding method that makes use of erasure correction capability of repeat accumulate codes and superiority of forbidden zone data hiding. Selective embedding is utilized in the proposed method to determine host signal samples suitable for data hiding. This method also contains a temporal synchronization scheme in order to withstand frame drop and insert attacks. The proposed framework is tested by typical broadcast material against MPEG-2, H.264 compression, frame-rate conversion attacks, as well as other well-known video data hiding methods. The decoding error values are reported for typical system parameters. The simulation results indicate that the framework can be successfully utilized in video data hiding applications.**

***Index Terms*—Data hiding, digital watermarking, forbidden zone data hiding, quantization index modulation, repeat accumulate codes, selective embedding.**

I. Introduction

DATA HIDING is the process of embedding information into a host medium. 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 host media type, the methods vary depending on the nature of such media. For instance, image and video data hiding share many common points; however video data hiding necessitates more complex designs [6], [7] as a result of the additional temporal dimension. Therefore, video data hiding continues to constitute an active research area.

Data hiding in video sequences is performed in two major ways: bitstream-level and data-level. In bitstream-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 bitstream; 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.

Despite their fragility, the bitstream-based methods are still attractive for data hiding applications. For instance, in [1], the redundancy in block size selection of H.264 encoding is exploited for hiding data. In another approach [17], the quantization parameter and discrete cosine transform (DCT) coefficients are altered in the bitstream-level.

However, most of the video data hiding methods utilize uncompressed video data. Sarkar *et al.* [2] 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 [3], 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 as in [4]. 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 [4] that such a scheme is successful when the number of selected host signal samples is much less than the total number of host signal samples.

In [5], 3-D DWT domain is used to hide data. They use LL subband coefficients and do not perform any adaptive selection. Therefore, they do not use error correction codes robust to erasures. Instead, they use BCH code to increase error correction capability. The authors performed 3-D interleaving in order to get rid of local burst of errors. Additionally, they proposed a temporal synchronization technique to cope with temporal attacks, such as frame drop, insert, and repeat.

In this paper, we propose a new block-based selective embedding type data hiding framework that encapsulates forbidden zone data hiding (FZDH) [8] and RA codes in accor-

Manuscript received September 25, 2010; revised January 13, 2011; accepted February 24, 2011. Date of publication April 5, 2011; date of current version August 3, 2011. This paper was recommended by Associate Editor R. C. Lancini.

E. Esen is with the TUBITAK UZAY Space Technologies Research Institute, Middle East Technical University, Ankara 06531, Turkey (e-mail: ersin.esen@uzay.tubitak.gov.tr).

A. Aydin Alatan is with the Department of Electrical and Electronics Engineering, Middle East Technical University, Balgat, Ankara 06531, Turkey (e-mail: alatan@eee.metu.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2134770

dance with an additional temporal synchronization mechanism. FZDH is a practical data hiding method, which is shown to be superior to the conventional QIM [9]. RA codes are already used in image [3] and video [2] 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 multi-dimensional form of FZDH that uses cubic lattice as its base quantizer.

We describe the utilized data hiding method in Section II. Then the proposed video data hiding method is presented in Section III. Experiment results are given in Section IV, which is followed by the concluding remarks.

II. Forbidden Zone Data Hiding

Forbidden zone data hiding (FZDH) is 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 tradeoff.

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

$$\mathbf{x} = \begin{cases} \mathbf{s}, & \mathbf{s} \in FZ_m \\ \mathrm{M}_m(\mathbf{s}), & \mathbf{s} \in AZ_m \end{cases} \tag{1}$$

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),

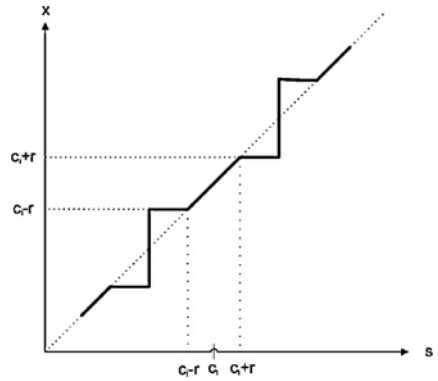

Fig. 1. Sample embedding function of FZDH in 1-D. c_i is a reconstruction point of the quantizer.

where the mapping function is defined as

$$\mathbf{M}_m(\mathbf{s}) = \left\{ \mathbf{s} + \mathbf{e}_m \left(1 - \frac{r}{\|\mathbf{e}_m\|} \right) . \right. \tag{2}$$

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

$$\mathbf{e}_m \stackrel{\Delta}{=} q_m(\mathbf{s}) - \mathbf{s}. \tag{3}$$

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 toward 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

$$FZ_m = \{\mathbf{s}| \, \|\mathbf{e}_m\| \leq r\} , \quad AZ_m = \{\mathbf{s}| \, \|\mathbf{e}_m\| > r\}. \tag{4}$$

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 [9]. A typical embedding function that uses a uniform quantizer is shown in Fig. 1.

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

$$\hat{m} = \arg\min_m d(\mathbf{y}, \mathbf{y}_m) \tag{5}$$

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.

For soft decoding, channel observation probabilities (o_m) are computed using the distances

$$o_m = \frac{d(\mathbf{y}, \mathbf{y}_m)^{-1}}{d(\mathbf{y}, \mathbf{y}_0)^{-1} + d(\mathbf{y}, \mathbf{y}_1)^{-1}}. \tag{6}$$

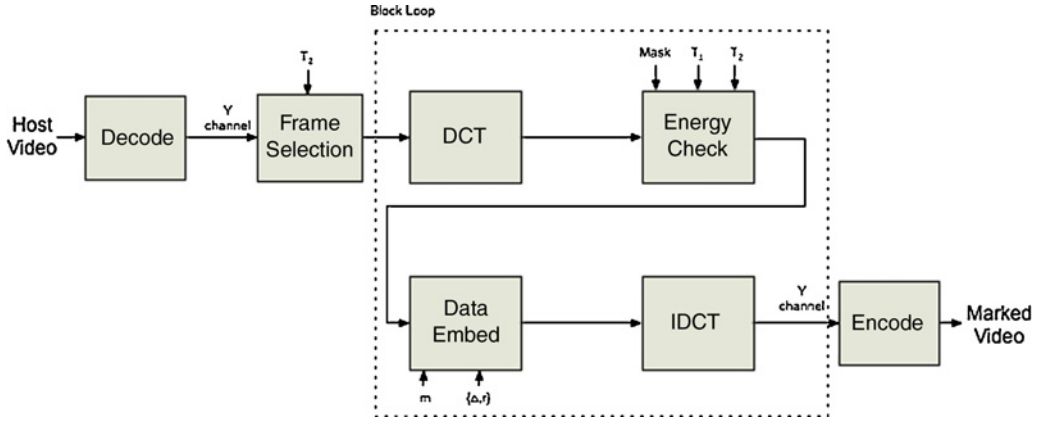

Fig. 2. Embedder flowchart of the proposed video data hiding framework for a single frame.

III. 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 [8], 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]. Unlike the method in [3], we employ block selection (entropy selection scheme [3]) and coefficient selection (selectively embedding in coefficients scheme [3]) together. The de-synchronization due to block selection is handled via RA Codes as in [2] and [3]. The de-synchronization due to coefficient selection is handled by using multi-dimensional form of FZDH in varying dimensions. In [2], the frames are processed independently. It is observed that [10] intra and inter frames do not yield significant differences. Therefore, in order to overcome local bursts of error, we utilize 3-D interleaving similar to [5], which does not utilize selective embedding, but uses the whole LL subband of discrete wavelet transform. Furthermore, as in [5], 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 paper 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.

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 8×8 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.

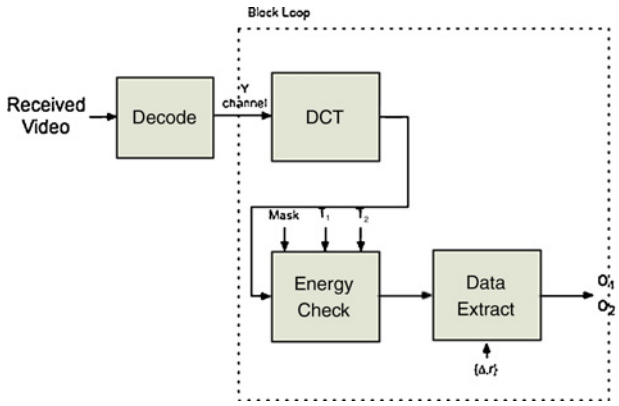

Fig. 3. Decoder flowchart of the proposed video data hiding framework for a single frame.

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.

1) Frame selection: selected number of blocks in the whole frame is counted. If the ratio of selected blocks to all blocks is above a certain value (T_0) the frame is processed. Otherwise, this frame is skipped.
2) Frequency band: only certain DCT coefficients are utilized. Middle frequency band of DCT coefficients shown in Fig. 4 is utilized similar to [2].
3) Block selection: energy of the coefficients in the mask is computed. If the energy of the block is above a certain

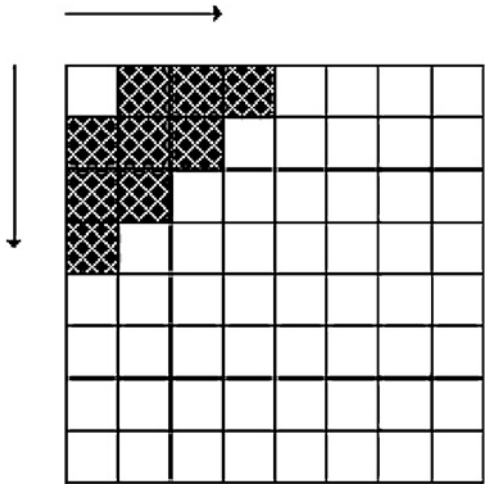

Fig. 4. Sample coefficient mask denoting the selected frequency band.

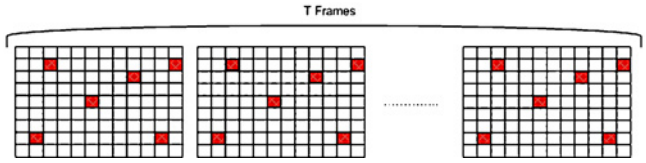

Fig. 5. Typical block partitioning for message bits and frame synchronization markers.

value (T_1) then the block is processed. Otherwise, it is skipped.

4) Coefficient selection: energy of each coefficient is compared to another threshold T_2. If the energy is 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. A typical partitioning is shown in Fig. 5. 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 [3] and video [2] data hiding in previous efforts.

RA code is a low complexity turbo-like code [11]. 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.

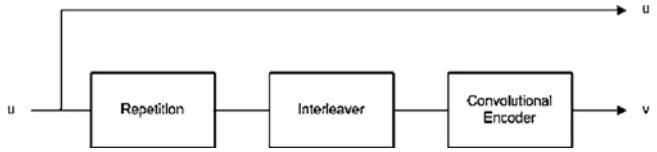

Fig. 6. RA encoder (u denotes source bits and $u+v$ denote encoded bits).

In systematic RA code, input is placed at the beginning of the output as shown in Fig. 6. In this paper, 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. We utilize the message passing algorithm given in [12].

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 portion of 2^{K_2} 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. Then, the actual number of message bits (K_1) becomes equal to $\lfloor T(N-RK_2)/R \rfloor$, where $\lfloor \cdot \rfloor$ denotes floor operation. The remaining blocks at the end of last frame are left untouched.

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.

1) If $f_{\text{cur}} > T$, then skip this frame. (This case corresponds to unmarked frame.)
2) If $f_{\text{cur}} = f_{\text{pre}}$, then skip this frame. (This case corresponds to frame repeat.)
3) 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 $f_{\text{cur}} < f_{\text{pre}}$, then the end of the group of frames is reached. Decode the message bits and obtain u_1. Initialize data structure.

TABLE I
DATA HIDING PARAMETERS

Average Embedding Distortion	QIM (Δ)	FZDH (Δ, r)
48 dB	30	40, 4
51 dB	15	20, 2

IV. EXPERIMENTS

We perform experiments in three stages. First, we compare QIM and FZDH by means of their raw decoding error performances without any error correction. Second, we observe the performance of the proposed framework against various common video processing attacks. Third, we compare the proposed video data hiding framework against JAWS [14], [15] and the method in [2] by using MPEG-2 compression attack.

A. FZDH Versus QIM

We utilize MPEG-2 DVB-S videos from five different TV channels. The total duration of the host video set is equal to 60 min (approximately 90 000 frames). The resolution of the videos is 720 by 576. Initial bitrates of the videos range from 6 Mb/s to 9 Mb/s. The marked videos are re-encoded at various bitrates and decoding errors are computed. The raw channel performance is measured by hiding the same data bit (i.e., constant m) to the whole video. Additionally, frame selection is not active. Hence de-synchronization due to selective embedding is not effective.

QIM and FZDH are compared at the same embedding distortion and data hiding rate. Two different embedding distortion values are utilized: 48 dB and 51 dB average PSNR. Embedding distortion is computed as the average PSNR between host and marked frames. The data hiding parameters that yield these values are tabulated in Table I. We should note that different pairs of (Δ, r) may yield the same embedding distortion. We make use of typical values determined manually. T_1 is selected as 2000 and T_2 is set to 1000. A typical host and marked frame pair for FZDH (at 48 dB) is shown in Figs. 7 and 8.

Comparison results against MPEG-2 compression attack are shown in Fig. 9 for Intra and Inter frames, respectively, for 48 dB and 51 dB cases. We observe that FZDH is superior to QIM, especially at low compression bitrates and small embedding distortion values. These conditions correspond to low WNR values; hence, the results comply with the reported results for AWGN in [8]. Furthermore, we observe the Intra and Inter frames do not yield significant differences.

B. Common Video Processing Attacks

At the second stage, we apply error correction and assess the performance of FZDH against some common video processing attacks. We utilize a typical TV broadcast material of 10 min. We prefer a smaller duration, which is still accurate to draw conclusions, due to the computational burden of RA decoding. The format of the test video is MPEG-2 at 9 Mb/s and its resolution is 720 by 576.

Fig. 7. Typical host frame.

Fig. 8. Corresponding marked frame using FZDH with $\Delta = 40$ and $r = 4$.

The system parameters are tuned manually. We utilize the following values during the experiments: $T_0 = 0.05$, $T_1 = 1000$, $T_2 = 500$, $K_2 = 10$, $T = 3$. One should note that threshold values are selected for this resolution and block size of 8 by 8. Different dimensions might require some other threshold values. Typical R values are used according to the attack. Once these values are set, the embedding rate is determined. For instance for $R = 150$, K_1 is 99, i.e., 33 bits are hidden per frame.

First, we observe the effect of the parameters on the number of selected block rate. Results for a 4 s video segment are shown in Fig. 10. The number of the selected blocks depends on the content and varies slowly with time. The abrupt changes correspond to shot boundaries. We observe that embedder and decoder select different number of blocks. Interestingly, for low rates the decoder can select higher number of blocks. However, due to frame selection at embedder (with respect to T_0), the decoder can correctly determine the group of frame and extract the hidden data as a result of the frame synchronization markers.

Second, we observe the decoding error performance against compression attack. We utilize typical bitrates for this resolution. We increase R to the point where one obtains error-free

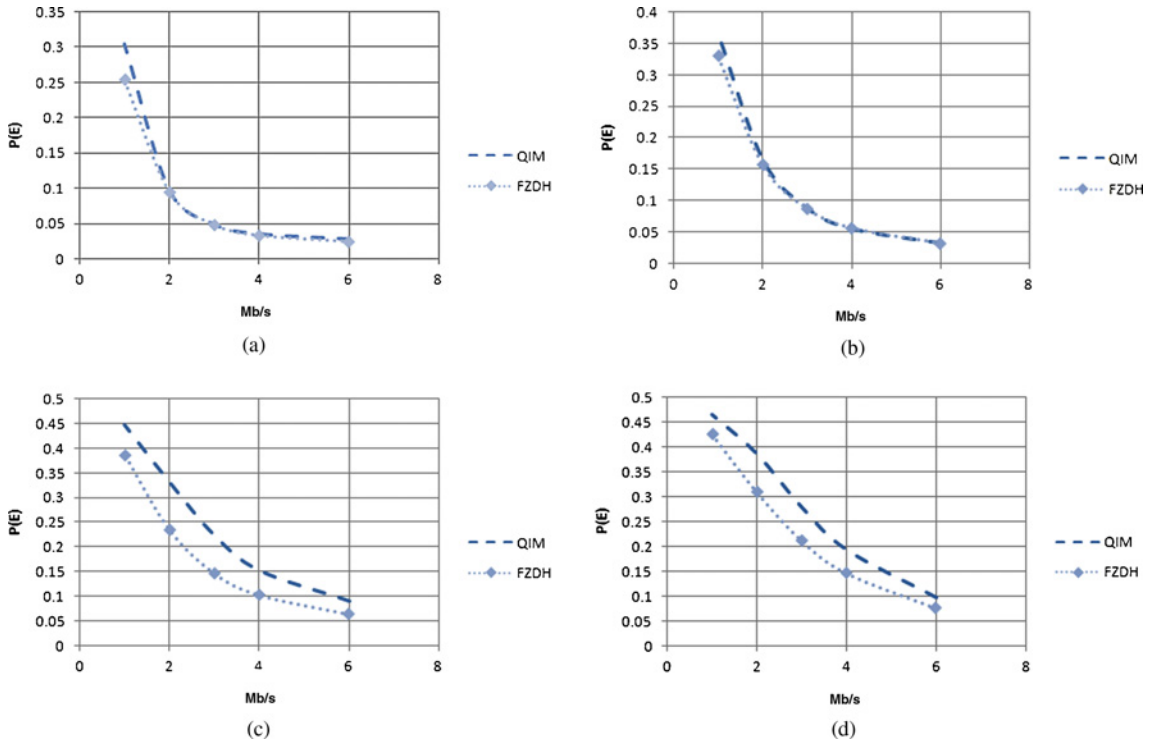

Fig. 9. FZDH versus QIM. (a) Intra frames, 48 dB average embedding distortion. (b) Inter frames, 48 dB average embedding distortion. (c) Intra frames, 51 dB average embedding distortion. (d) Inter frames, 51 dB average embedding distortion.

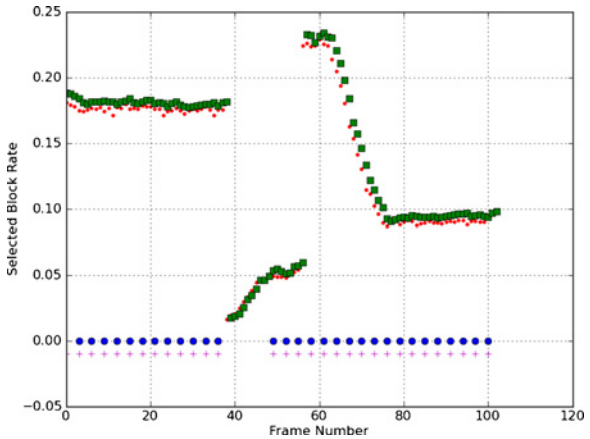

Fig. 10. Typical selected block rates and decoding errors for $T_0 = 0.05$, $T_1 = 1000$, $K_1 = 10$, $T = 3$, $R = 150$, $\Delta = 80$, $r = 8$ and MPGE-2 4 Mb/s compression. *Circle* denotes decoding error, *square* denotes selected block rate at embedder, *dot* denotes selected block rate at decoder, and *plus* denotes the frame locations where message is embedded.

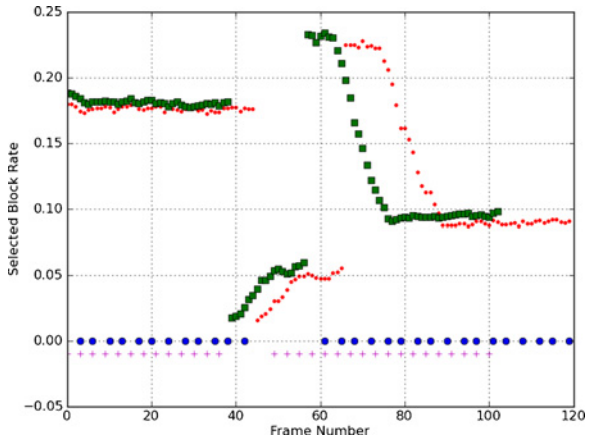

Fig. 11. Typical selected block rates and decoding errors for T_0=0.05, T_1=1000, K_1=10, T=3, R=150, δ=80, r=8, and MPEG-2 4 Mb/s compression against frame rate conversion from 25 f/s to 30 f/s. Circle denotes decoding error, square denotes selected block rate at embedder, dot denotes selected block rate at decoder, and plus denotes the frame locations where message is embedded.

decoding. The results are tabulated in Table II for two different embedding distortion values. 46.0 dB and 40.46 dB embedding distortion values are obtained with the following FZDH parameters $\{\Delta = 40, r = 4\}$, $\{\Delta = 80, r = 8\}$, respectively. The results indicate that we need repetition number higher than the erasure rate. The reason for this observation is due to the fact that decoding errors occur as a result of compression as well as the erasures due to the block selection. Furthermore, we observe that H.264 appears to be a stronger attack compared to MPEG-2. Therefore, we need higher repetition for error-free decoding.

Third, we test the performance of the method against another common video processing: frame-rate conversion. The frame rate of the original video is 25 f/s. We change this frame-rate to a higher as well as a lower value and measure the decoding error rate. We should note that frame-rate conversion could be achieved in various ways, some of which could be quite complex. However, we utilize an open source codec (ffmpeg[1]), which performs frame-rate conversion by frame drop/repeat and re-encoding. First, we present the selected

[1]Available at http://www.ffmpeg.org.

TABLE II
DECODING ERROR FOR MPEG-2 AND H.264 COMPRESSION ATTACKS AT 4 MB/S AND 2 MB/S

		Embedding Distortion (46.0 dB)		Embedding Distortion (40.68 dB)	
		$R = 100$	$R = 150$	$R = 100$	$R = 150$
MPEG-2	4 Mb/s	0.0170089	0.014457	0.000098	0
	2 Mb/s	0.207233	0.136266	0.0349629	0.0281465
H.264	4 Mb/s	0.327757	0.230392	0.007989	0.000469
	2 Mb/s	0.494789	0.482247	0.151802	0.070663

TABLE III
DECODING ERROR FOR FRAME RATE CONVERSION AT 40.68 DB EMBEDDING DISTORTION AND MPEG-2 4 MB/S

Frame Per Second	$R = 150$	$R = 200$
30	0.000221199	0
23.98	0.00020057	0

TABLE IV
DECODING ERROR FOR DOWNSCALING AT MPEG-2 4 MB/S AND 40.6 DB AVERAGE EMBEDDING DISTORTION

Size	$R = 200$	$R = 250$
CIF (352×288)	0.492966	0.443106
VGA (640×480)	0.0136099	0.00362576
SVGA (800×600)	0.0150746	0.00525774

block rates at embedder and decoder in Fig. 11 by using the same video segment in Fig. 10. The frame rate of the marked video is changed to 30 f/s from 25 f/s. We observe that even if the message locations are shifted, we can successfully decode the message bits as a result of the frame synchronization markers.

The total decoding error results are tabulated in Table III. We observe that different rates have similar results. Frame insertions and drops do not differ as long as they can be detected correctly by synchronization markers. We should note that for lower f/s value of 23.98, frame drop rate is quite small and at most one frame per group of frames can be dropped. Additionally, we require higher repetitions than compression attack.

Finally, we test the scaling performance. For this purpose, we downscale the marked video, and then, upscale the attacked video to its original size. Scaling operations are performed again using ffmpeg library. The decoding error values for three different dimensions are given in Table IV.

Scaling test results indicate that CIF resolution attack totally removes the hidden data. On the other hand, we can obtain better results for VGA and SVGA resolutions. However, error-free decoding is not possible with the utilized system parameters. One should increase the repetition rate, embedding distortion, or number of frames in order to achieve error-free decoding.

C. Proposed Framework Against JAWS and Sarkar

We compare the proposed framework against the canonical video watermarking methods JAWS and a more recent quantization based method [2].

JAWS is a spatial domain additive spread spectrum based watermarking method [14]. It is utilized for DVD copyright protection [16] and broadcast monitoring [15]. In JAWS, luminance channel of the frame in tiles. A pseudo-random watermark pattern is generated. The size of this pattern matches the size of the tile elements. The payload is increased by using shifted versions of the base watermark. The amount of payload depends on the number of shifts and possible shift locations determined by a fixed grid. The superposition of the base watermark and its shifted versions is added to each tile element of the luminance channel spatially. The strength of the watermark is adjusted by means of a global scaling parameter. At the decoder, the received frame is folded by averaging tile elements and the same base watermark is generated. Correlation is performed to detect the locations of the peaks in Fourier domain. The relative positioning of the peaks gives the decoded message bits.

TABLE V
JAWS DECODING ERROR FOR MPEG-2 COMPRESSION

Global Scaling Parameter	Average Embedding Distortion	2 Mb/s	4 Mb/s	6 Mb/s
0.25	40.2 dB	0.30375	0.1436	0.1164
0.5	34.44 dB	0.15425	0.10455	0.0987

We compare the proposed framework and JAWS at the same data hiding rate and embedding distortion. We utilize the same host video in Section IV-B. For JAWS, we utilize a tile with four elements, four shift locations inside a grid of 32 by 32. The resultant data hiding bit rate is 30 bits per frame. This rate is assured with $R = 150$ (33 bits per frame) in the proposed framework. The global scaling parameter of JAWS is varied to adjust the embedding distortion. The results of JAWS against MPEG-2 compression attack are given in Table V. When we compare Tables II and V, we observe that at the same embedding distortion (40 dB PSNR), the proposed framework is significantly superior to JAWS. The results indicate that for high payload applications JAWS cannot achieve an acceptable performance level.

We compare the proposed framework and adaptive quantization based scheme of Sarkar *et al.* [2] by means of the results reported in [2]. In this method, conventional QIM is applied in uncompressed domain to selected low-frequency DCT coefficients. The quantization parameter is adaptively adjusted according to the type of the frame. The resulting desynchronization due to coefficient selection is handled by means of RA codes. The different utilization of I/P/B frames results in the unexpected situation that the decoding error decreases with decreasing embedding distortion. However, we base our comparison with the best result obtained in [2].

TABLE VI
MINIMUM SEGMENT DURATION (IN S) REQUIRED FOR TARDOS FINGERPRINTING

R	$\varepsilon_1 = 10^{-9}$		$\varepsilon_1 = 10^{-12}$	
	$C_o = 10$	$C_o = 20$	$C_o = 10$	$C_o = 20$
100	62.6	250	83.5	332
150	103.8	414	138.4	551
200	153.3	611	204	815

We utilize the same host video as in [2]. Since the host video is QVGA (320 by 240) size, we adjust method parameters accordingly. First, we reduce the repetition number, R, to 4 and obtain embedding rate of 300 bits per frame, whereas the best result is obtained for 294.2 bits per frame in [2]. Second, we utilize $\Delta = 70$, $r = 10$ with $T = 3$ and obtain an embedding distortion of 37.06 dB PSNR, whereas the corresponding value is 37.02 dB in [2].

In these conditions, the proposed method yields 0.006 decoding error rate for MPEG-2 compression attack at 4 Mb/s. On the other hand, the performance of [2] is given in terms of "frame error rate," which corresponds to the frames with non-converging RA decoder, and the best result reported in [2] is 0.008. Assuming zero error for converging RA decoder, one can claim that they have comparable performances.

However, we should note that at this level of embedding distortion visible artifacts occur and hence for a more realistic comparison embedding distortion should be decreased to an acceptable level.

V. CONCLUSION

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

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

The framework was 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 [18], 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. The length of the Tardos fingerprint is $AC_0^2 \ln \frac{1}{\varepsilon_1}$ [19], where A is a function of false positive probability (ε_1), false negative probability, and maximum size of colluder coalition, (C_o). The minimum segment durations required for Tardos fingerprinting in different operating conditions are given in Table VI.

We also compared the proposed framework against the canonical watermarking method, JAWS, and a more recent quantization based method [2]. The results indicate a significant superiority over JAWS and a comparable performance with [2].

The experiments also shed light on possible improvements on the proposed method. First, 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 as in [13] remains as a future direction.

REFERENCES

[1] S. K. Kapotas, E. E. Varsaki, and A. N. Skodras, "Data hiding in H-264 encoded video sequences," in *Proc. IEEE 9th Workshop Multimedia Signal Process.*, Oct. 2007, pp. 373–376.

[2] A. Sarkar, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Adaptive MPEG-2 video data hiding scheme," in *Proc. 9th SPIE Security Steganography Watermarking Multimedia Contents*, 2007, pp. 373–376.

[3] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," *IEEE Trans. Image Process.*, vol. 13, no. 12, pp. 1627–1639, Dec. 2004.

[4] M. Schlauweg, D. Profrock, and E. Muller, "Correction of insertions and deletions in selective watermarking," in *Proc. IEEE Int. Conf. SITIS*, Nov.–Dec. 2008, pp. 277–284.

[5] H. Liu, J. Huang, and Y. Q. Shi, "DWT-based video data hiding robust to MPEG compression and frame loss," *Int. J. Image Graph.*, vol. 5, no. 1, pp. 111–134, Jan. 2005.

[6] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video: I. Fundamental issues and solutions," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 685–695, Jun. 2003.

[7] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video: II. Designs and applications," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 696–705, Jun. 2003.

[8] E. Esen and A. A. Alatan, "Forbidden zone data hiding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1393–1396.

[9] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[10] 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 *Proc. IEEE 17th Signal Process. Commun. Applicat. Conf.*, Apr. 2009, pp. 404–407.

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

[12] M. M. Mansour, "A turbo-decoding message-passing algorithm for sparse parity-check matrix codes," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4376–4392, Nov. 2006.

[13] Z. Wei and K. N. Ngan, "Spatio-temporal just noticeable distortion profile for grey scale image/video in DCT domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, 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 *Proc. IEEE ICMCS*, vol. 1. Jul. 1999, pp. 7–12.

[15] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes, "Video watermarking system for broadcast monitoring," in *Proc. SPIE Security Watermarking Multimedia Contents Conf.*, vol. 3657. 1999, pp. 103–112.

[16] M. Maes, T. Kalker, J.-P. M. G. Linnartz, J. Talstra, F. G. Depovere, and J. Haitsma, "Digital watermarking for DVD video copy protection," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 47–57, Sep. 2000.

[17] K. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, "Complete video quality-preserving data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1499–1512, Oct. 2009.

[18] G. Tardos, "Optimal probabilistic fingerprint codes," in *Proc. 35th Annu. ACM STOC*, 2003, pp. 116–125.

[19] B. Skoric, T. U. Vladimirova, M. Celik, and J. C. Talstra, "Tardos fingerprinting is better than we thought," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3663–3676, Aug. 2008.

Ersin Esen received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 1998 and 2000, respectively, and the Ph.D. degree in electrical and electronics engineering from Middle East Technical University, Ankara, in 2010.

Since 2001, he has been with the TUBITAK UZAY Space Technologies Research Institute, Ankara. He has been involved in various projects related to his research interests, which include data hiding, multimedia archive management, content based copy detection, and audio-visual concept detection.

A. Aydin Alatan (S'91–M'07) was born in Ankara, Turkey, in 1968. He received the B.S. degree from Middle East Technical University, Ankara, in 1990, the M.S. and DIC degrees from the Imperial College of Science, Medicine and Technology, London, U.K., in 1992, and the Ph.D. degree from Bilkent University, Ankara, in 1997, all in electrical engineering.

He was a Post-Doctoral Research Associate with the Center for Image Processing Research, Rensselaer Polytechnic Institute, Troy, NY, from 1997 to 1998, and with the New Jersey Center for Multimedia Research, New Jersey Institute of Technology, Newark, from 1998 to 2000. In August 2000, he joined the faculty of the Department of Electrical and Electronics Engineering, Middle East Technical University.