

Matrix Encoding for Data Hiding Using Multilayer Video Coding and Transcoding Solutions

Tamer Shanableh

Department of Computer Science and Engineering

American University of Sharjah

Fax: +971 6 515-2979

tshanableh@aus.edu

Abstract

This paper proposes a multitier data hiding solution in MPEG video using matrix encoding. In the first tier of the proposed solution, raw video is encoded into two layers using multilayer Signal to Noise Ratio (SNR) encoding. During the multilayer encoding process, the quantization scales and coding parameters are stored. In the second tier, message bits are embedded in the stored quantization scales using matrix encoding. In the third and last tier, the raw video is encoded again into two layers using the modulated quantization scales and coding parameters of the first encoder. The proposed multitier solution is extended by replacing the multilayer encoder with a multilayer transcoder to accommodate situations in which the video exists in pre-coded format only. By the use of multilayer encoder or transcoding, we show that the message payload can be doubled without adversely affecting the coding quality or bitrate of the host video. We also show that message embedding using matrix encoding can be extended to modulate motion vectors. The paper also proposes two solutions for motion vector modulation based on the availability of the raw video. In comparison to existing work, the proposed solutions have higher message payloads, lower degradation in video quality and result in reduced excessive bitrate.

Keywords: data hiding, steganography, matrix encoding, video transcoding, scalable video coding.

1. Introduction

There are a number of conflicting requirements in data hiding using compressed media. Data should be embedded whilst causing minimal distortions and retaining full compatibility with the compressed media decoder. The embedding process might result in visual distortions; such distortions are required to remain subjectively imperceptible. Moreover, depending on the purpose of data hiding, some applications require embedding high payload data which is likely to result in excessive bitrate when the host media is compressed. Such an excessive bitrate is undesired and is required to remain within reasonable bounds.

In general, data hiding has a number of applications such as watermarking [1,2], copyright protection [3], content authentication [4], transaction tracking [5], access control [6], content annotation and error resiliency and concealment [7]. Data hiding is also used in less obvious applications such as enhancing image and video compression. For instance, the work in [8] used data hiding to hide color information in the luminance component of an image. The hiding is performed in the wavelet domain of the image prior to compression. And [9] reported a similar approach where the DCT domain was used instead of the wavelet domain to hide the color information. More recently [10] proposed the use of data hiding in the H.264/AVC[11] codec to hide motion vector coding parameters into selected DCT coefficients. As such, reducing the overall bitrate of the compressed video.

Video bitstream syntax can be used for data hiding. For instance the work in [12] proposed the use of intra prediction modes to hide message bits. It was shown that 1 bit can be hidden in each candidate 4x4 intra block. Additionally, the work in [13] utilized the block types and modes of intra-coded blocks of H.264/AVC to hide message bits. The sign of the trailing ones in the context-adaptive variable length coding (CAVLC) of the H.264/AVC encoder can also be used for data hiding as reported in [14]. Lastly, the ‘group slice id’ of macroblocks coded using flexible macroblock ordering in H.246/AVC was also used for hiding data [15].

Typical data hiding venues in compressed video are the DCT coefficients [16], motion vectors and quantization scales. The use of motion vectors for data hiding was first reported in [17]. The work in [18] used the phase angles of motion vectors as a hiding venue. Whereas, the work in [19] and [20] used the magnitude of motion vectors as a hiding venue. Only a subset of the coded motion vectors is used for hiding data. This subset is chosen carefully to reduce the excessive bitrate as a result of data hiding. Likewise, the quantization scale can be used as a venue for data hiding as reported in [21] and [22]. More recently, the author proposed the use of multivariate regression to predict the

value of message bits hidden in quantization scales [15]. Additionally, data hiding in quantization scales is also used in wavelet-based video coding for error concealment purposes as proposed in [23].

In this work, we propose the use of matrix encoding [24] to hide data in quantization scales and motion vectors of compressed MPEG videos. We refer to data hiding as message embedding, we also refer to data hiding in quantization scales and motion vectors as quantization scale modulation and motion vector modulation respectively. Since the host video might already be compressed and the original raw video is not available, we propose the use of video transcoding for message embedding. In comparison to [22], our work doubles the payload of embedded messages in the quantization scales through the use of multilayer video encoding and transcoding. And in comparison to [20], our work embeds messages in motion vectors using a non-iterative procedure that is suitable for message embedding regardless of the availability of the original raw video.

This paper is organized as follows. Section 2 introduces the quantization scale modulation solution using multilayer video encoding and transcoding. Section 3 reviews matrix encoding and its use in data hiding or message embedding. Section 4 introduces the use of matrix encoding for motion vector modulation in encoding and transcoding contexts. The experimental results are reported in Section 5. Lastly, Section 6 concludes the paper.

2. Quantization scale modulation:

Message embedding using quantization scale modulation is limited by the number of quantization scales available in coded video frames. In an attempt to increase the payload of the embedded messages, we propose the use of multilayer video encoding and transcoding. In multilayer encoding, a digital video can be coded into a base layer with a low resolution and one or more enhancement layers to enhance the temporal, spatial and/or quality resolution. The quality resolution enhancement is also known as Signal to Noise Ratio (SNR) multilayer encoding. Detailed information about multilayer video encoding can be found in [25].

In this work, we use SNR multilayer encoding to generate a two layer video with the same temporal and spatial resolutions. The two layers are referred to as the base and the enhancement layers. As such, each macroblock is coded into two layers with two different quantization scales, therefore potentially doubling the payload of embedded messages.

A general block diagram for SNR multilayer MPEG encoding is shown in Figure 1. In general, there is more than one approach for implementing SNR multilayer encoding. Figure 1 shows the simplest approach in which the difference between an original and a reconstructed image is computed and

fed into a second video encoder. The difference between the two images is due to quantization error. Another approach to SNR scalable encoding is to feed the dequantized DCT coefficients of the enhancement layer into the frame buffers of the base layer. As such, forming a more accurate source of prediction for the base layer frames. This helps in enhancing the overall video quality as elaborated upon in the experimental results section.

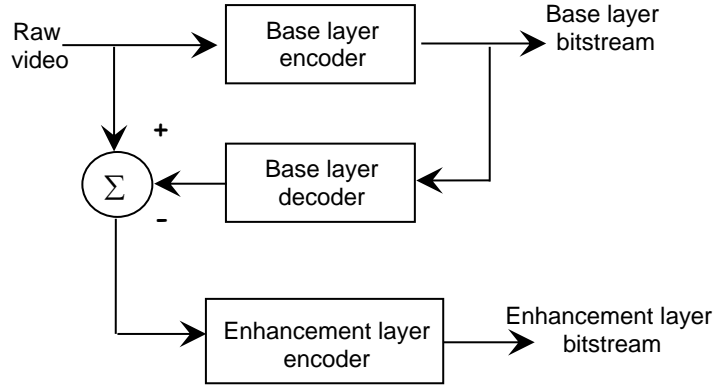


Figure 1. Simplified SNR multilayer MPEG encoder.

If the video is already coded and the original raw video is not available, then video transcoding techniques are needed. In previous work, the author proposed a number of MPEG video transcoders to convert a single layer video into multilayers with temporal, spatial and/or SNR scalability [26]. Similar work was then reported for H.264/AVC-SVC in [27]. A two layer SNR transcoder is shown in Figure 2.

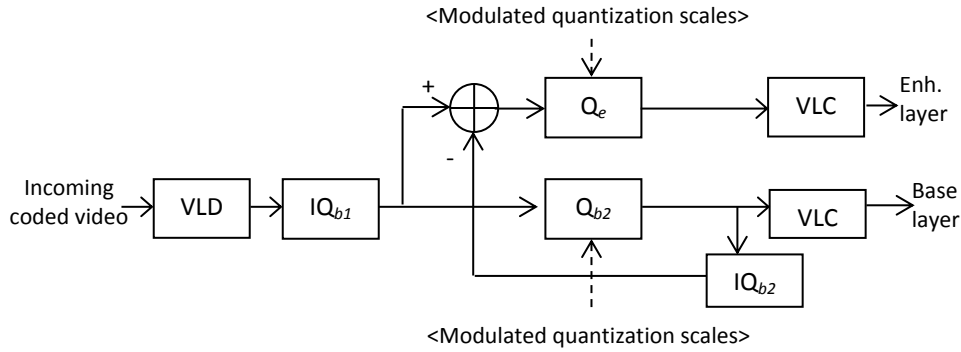


Figure 2. Proposed two layer video transcoder for message embedding.

Note that both of the encoder and the transcoder of Figures 1 and 2 are compatible with MPEG-2 and H.264/AVC annex G.

In this work, we propose to encode the video in two layers such that the total bitrate does not exceed that of single layer coding. For instance, in Figure 2, the incoming video is transcoded into two video layers with a total bitrate equal to that of the incoming video. More specifically, the incoming video is first Variable Length Decoded (VLD) and Inverse Quantized (IQ_{b1}) using the quantization scales of the incoming bitstream, hence the subscript $b1$. The dequantized DCT coefficients are then re-quantized according to the bitrate of the outgoing base layer. When message embedding is used, these values shall be modulated first using matrix encoding. Consequently, the dequantized coefficients (the output of the IQ_{b2} block) are subtracted from the incoming dequantized coefficients (the output of the IQ_{b1} block). The result of the subtraction is due to the requantization error, this error is further encoded in the enhancement layer (using the quantizer of the enhancement layer Q_e) taking into account the modulation of the quantization scales.

To embed a message using the proposed SNR multilayer encoding or transcoding, the following 3 tier procedure is followed. In the first tier, if the original raw video is not available, then the coded video is first transcoded into two layers such that the total bitrate does not exceed that of the incoming single layer video. Whilst transcoding, macroblock level information is gathered and stored into separate files. This includes MB types, motion information and the new quantization scales. Clearly, as a result of video transcoding into 2 layers, the quantization scales will differ and some additional MBs might become skipped. This information is stored for both video layers.

In the second tier, the new quantization scales are modulated according to the message bits to embed. The procedure for message embedding is explained in details in the next section. For accurate modulation, skipped macroblocks shall be identified because they do not carry quantization scales, hence, the input to the modulator is both the new quantization scales and the macroblocks types.

In the third and last tier, the incoming single layer video is transcoded again into two layers, but this time, the transcoder uses the modulated quantization scales. Hence the output of the transcoder is two video layers with embedded message bits. It is important to mention that in this process, the transcoder should use the same macroblock types and motion information that was gathered in tier 1. Otherwise, because of the modulated quantization scales, the percentage of skipped macroblocks might be different than that of tier 1. As a consequence, the indices of the coded macroblocks used for message embedding might differ from those used at the decoder to extract the embedded message. Hence the message extraction process losses synchronization with the embedding process, leading to inaccurate extraction of message bits. The proposed message embedding solution is

further illustrated in Figure 3. In the figure, MB refers to a macroblock and mquant refers to a quantization scale. The figure also refers to the variable k which is the matrix encoding parameter to be explained in the next section.

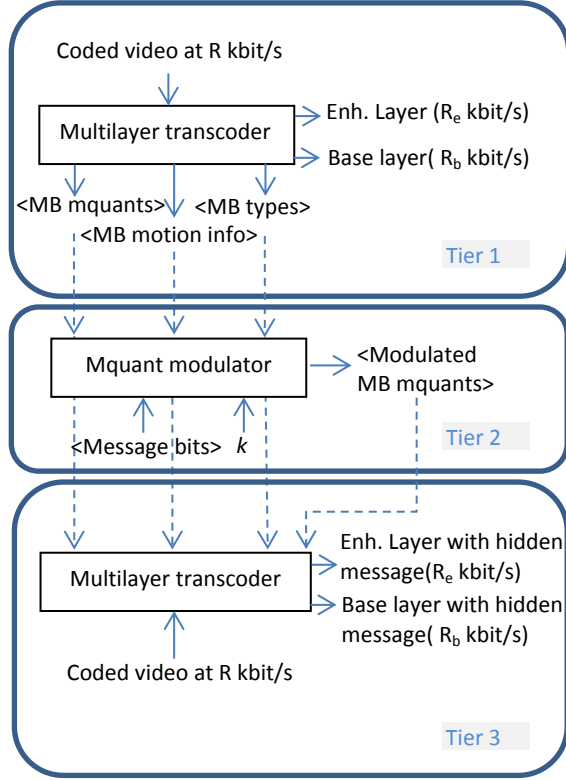


Figure 3. Message embedding using the proposed multilayer transcoding solution.

On the other hand, if the original raw video is available then multilayer video encoding can be used instead of video transcoding. The proposed 3 tier solution of Figure 3 applies provided that the transcoding process in tiers 1 and 3 are replaced by multilayer encoding.

The payload of embedded messages and the effect of the proposed solution on video bitrate and quality are elaborated upon in the experimental results section.

3. Matrix encoding:

An efficient technique for message embedding was introduced in [24] which is known as matrix encoding. The technique reduces the number of changes required to embed a given message. The first implementation of matrix encoding was reported in [28] where the least significant bits of DCT coefficients in JPEG image compression are used to embed message bits. Matrix encoding was also used for modulating the quantization scales of single layer video as reported in [22]. In this work, we

use a similar technique for message embedding but we propose a new a solution for modifying the quantization scales.

More specifically, message bits are divided into k bit long segments. The k parameter is also known as the matrix encoding parameter. Typical values of k used in this work for data hiding are 2,3 and 4. Each message segment is then embedded in the video bitstream using 2^k-1 quantization scales. For a particular example of $k = 3$, the dependencies between the message bits $m_i ; i=1..k$ and the quantization scales $q_j ; j=1..2^k-1$ are shown in Table 1. The purpose of these dependencies is explained with an example in Table 3.

Table 1. Dependencies between message bits and quantization scales

	q_1	q_2	q_3	q_4	q_5	q_6	q_7
m_1	√		√		√		√
m_2		√	√			√	√
m_3				√	√	√	√

To embed a message segment, a message bit value is compared against the parity of the corresponding quantization scales of Table 1. If the parity is different than the message bit value, then one of the quantization scales is modified accordingly. For example, if $m_1 \neq q_1 \oplus q_3 \oplus q_5 \oplus q_7$ (i.e. $m_1 \neq \text{mod}(q_1 + q_3 + q_5 + q_7, 2)$) then q_1 is either increased or decreased in value. Note that for embedding any message segment, a maximum of one quantization scale shall be modified. With the case of $k=3$ and according to the dependencies in Table 1, let p_1, p_2 and p_3 denote the quantization scale parities. That is, let $p_1 = q_1 \oplus q_3 \oplus q_5 \oplus q_7$, $p_2 = q_2 \oplus q_3 \oplus q_6 \oplus q_7$ and $p_3 = q_4 \oplus q_5 \oplus q_6 \oplus q_7$. Consequently, the embedding of a 3-bit message segment shall require one of the actions listed in Table 2. The action is determined according to the values of message bits as illustrated in the ‘parity tests’ column of the same table.

Table 2. Matrix encoding example with ($k=3$)

Parity tests			Action
$m_1=p_1$	$m_2=p_2$	$m_3=p_3$	No change
$m_1 \neq p_1$	$m_2 \neq p_2$	$m_3 \neq p_3$	Modify q_7
$m_1 \neq p_1$	$m_2 \neq p_2$	$m_3 = p_3$	Modify q_3
$m_1 \neq p_1$	$m_2 = p_2$	$m_3 \neq p_3$	Modify q_5
$m_1 = p_1$	$m_2 \neq p_2$	$m_3 \neq p_3$	Modify q_6
$m_1 \neq p_1$	$m_2 = p_2$	$m_3 = p_3$	Modify q_1
$m_1 = p_1$	$m_2 \neq p_2$	$m_3 = p_3$	Modify q_2
$m_1 = p_1$	$m_2 = p_2$	$m_3 \neq p_3$	Modify q_4

In general, the embedding rate using matrix encoding is defined as:

$$R(k) = \frac{k}{n} = \frac{k}{(2^k-1)} \quad (1)$$

In video coding, the maximum number of message bits to embed is equal to the maximum number of quantization scales available in the video bitstream. If the video is coded at a spatial resolution of 352x288 and a frame rate of 30Hz, then the maximum message payload is (352/16)x(288/16)x30Hz which is around 11.9 Kbit/s. When matrix encoding is used, the maximum message payload becomes $11.9(\text{Kbits}) * \frac{k}{(2^k-1)}$. If $k=3$ then the payload is around 5 Kbit/s. It is clear that such a payload might not be reached because some macroblocks are skipped in the coding process. Additionally, if a macroblock uses the same quantization scale used by its successor then such a quantization scale is not encoded in the bitstream. Hence, reducing the number of quantization scales available for modulation.

In this work, we propose to increase the message payload by the use of multilayer encoding/transcoding as mentioned previously. With two layer encoding/transcoding, the message payload can be double. Table 3 shows the message rates for $k=1..4$ using the above-mentioned video coding parameters. The table compares the theoretical maximum message payload of the proposed solution against that reported in [22].

Table 3. Maximum message payload of the proposed solution.

k	Maximum message payload (Kbit/s)	
	Reviewed solution[22]	Proposed Solution
1	11.9	23.8
2	7.9	15.8
3	5	10
4	3.17	6.34

3.1 Sub-optimal histogram preserving:

Following the work reported in [22], this section proposes a sub-optimal histogram preserving algorithm for the modification of the quantization scales.

The proposed algorithm is described as follows. A weight vector is created and initialized to zeros. The values in the vector represent the difference between the histograms of the quantization scales before and after the modulation. These values are denoted by w_j ; $j=1..\text{max quantization scale value}$. The algorithm reads one group of quantization scales at a time denoted by q_i ; $i=1..2^k-1$. If for example, a quantization scale at index $i=x$ (i.e. q_x) is modified because of modulation to $q_x+\Delta$ where

$\Delta = -1$ or $+1$, then the weight vector is modified by decrementing the weight that corresponds to the original quantization scale value, i.e. $w_{q_x} \leftarrow w_{q_x} - 1$. The weight vector is also modified by incrementing the weight that corresponds to the new quantization scale value i.e.

$w_{q_{x+\Delta}} \leftarrow w_{q_{x+\Delta}} + 1$. Prior to making a choice on whether to modify q_x to $q_x + 1$ or $q_x - 1$, the corresponding values in the weight vector are examined (i.e. w_{q_x-1} and w_{q_x+1}). Preference is given to new quantization scale that corresponds to a smaller weight. Ties are broken randomly. As such, an attempt is made to keep the values in the weight vector as close to zero as possible, hence preserving the frequencies of the original quantization scale values prior to modulation. The algorithm is further illustrated in Figure 4.

Following the work in [22], we quantify the distance between the histograms of the original and the modulated quantization scales using the following L_1 -norm:

$$f(x) = \frac{1000}{c} \sum_{j=1}^{31} |h_j - h'_j| \quad (2)$$

Where h_j and h'_j represent the histogram frequencies at the j^{th} bin of the original and modulated quantization scales respectively, c is the total number of coded (non-skipped) macroblocks in the video sequence. The total number of histogram bins is 31 which is equal to the allowable range of quantization scales in the employed MPEG video coder. The smaller the histogram distance the higher is the evidence that the two histograms come from the same distribution.

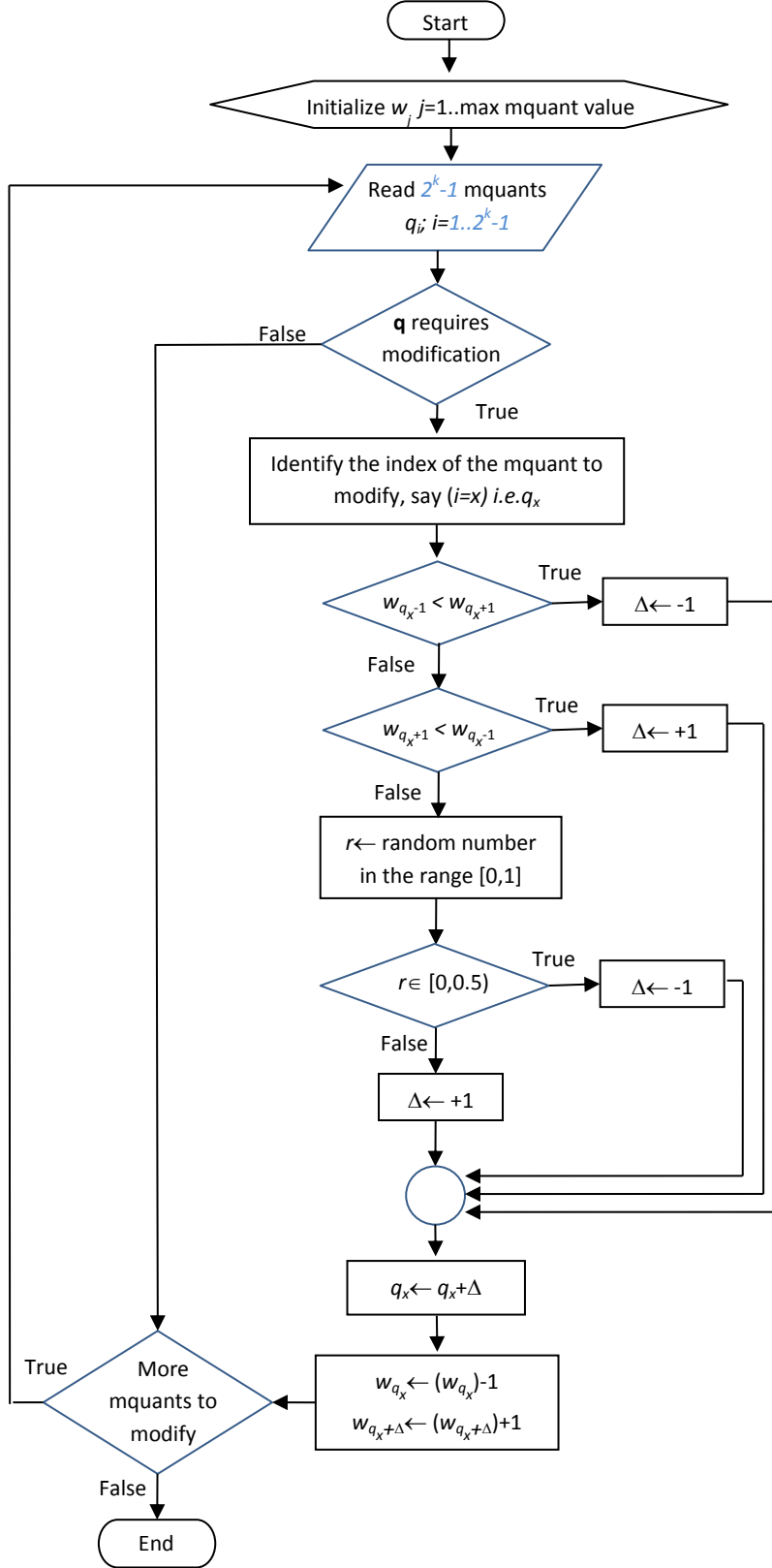


Figure 4. Flowchart of the proposed quantization scale modification algorithm.

4. Motion vector modulation:

The proposed solution of quantization scale modulation using matrix encoding can be extended to motion vector modulation. In this case, the x and y components of motion vectors can be modified to embed message bits. In a similar arrangement to quantization scale modulation, there are two approaches to embedding messages depending on whether or not the original video images are available. The block diagram of the proposed solution is shown in Figure 5.

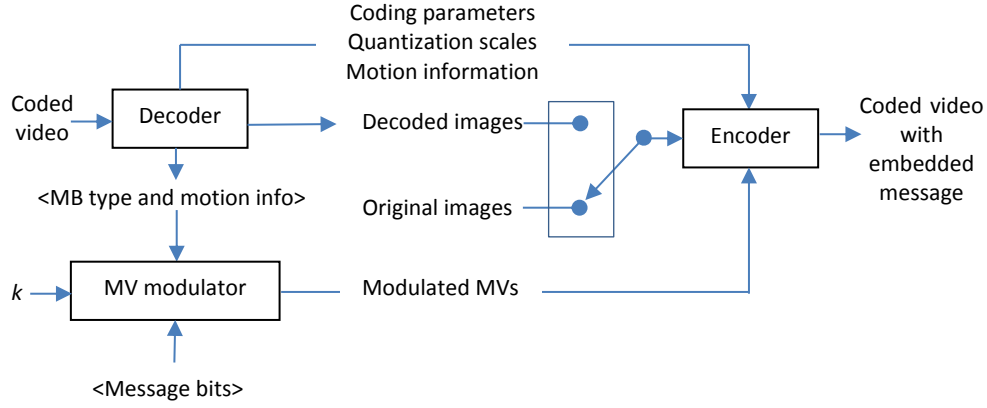


Figure 5. Block diagram of the proposed motion vector modulation solution.

As shown in the figure, if the original images are not available then a transcoding approach is followed. The input video is first decoded and the parsed coding parameters, motion information and quantization scales are passed alongside the decoded images to an encoder for reencoding. The motion vectors on the other hand, are passed to the modulation block where matrix encoding is used to embed the message as explained in Section 3 above. Note that the macroblock types are also needed at this point because skipped macroblocks shall be identified and excluded from modulation as they do not carry motion vectors. Once the motion vectors are modified, new macroblock prediction errors have to be computed. This is needed because changing the value of a motion vector results in changing the location of the prediction source as well. Therefore, reencoding using the modified motion vectors is required as shown in Figure 5.

On the other hand, if the original images are available, then the same modulation procedure is followed except that the input to the encoder will be the original instead of the decoded images. In this case, the quality of the output video is expected to be higher. We will refer to this approach as 2-pass encoding.

In terms of message payload, the maximum number of message bits to embed is equal to the maximum number of motion vector components available in the video bitstream. If the video is coded at a spatial resolution of 352x288 and a frame rate of 30Hz with 2 I frames, 8 P frames and 20 B frames per second, then the maximum message payload can be computed for both P and B frames as follows. Assuming one motion vector per macroblock in P frames (with two components x and y), the message payload per second is $(352/16) \times (288/16) \times 2 \times 8\text{Hz}$ which is around 6.3 Kbit/s. In B frames on the other hand, if we assume two motion vectors per macroblock, then the message payload is $(352/16) \times (288/16) \times 4 \times 20\text{Hz}$ which is around 31.7 Kbit/s. When matrix encoding is used, the maximum message payload becomes $(6.3+31.7)(\text{Kbit/s}) * \frac{k}{(2^k-1)}$. If $k=2$ then the maximum payload is around 25.3 Kbit/s. If $k=3$ then the maximum payload is 16.3 Kbit/s. Again, such a payload might not be reached because some macroblocks are skipped in the video coding process.

We compare the aforementioned solution to a recently reported solution for motion vector modulation in [20]. It was proposed to adaptively modulate motion vectors based on the associated Peak Signal to Noise Ratio (PSNR) of individual macroblocks. The idea is to compute a PSNR threshold per video frame. Motion vectors of macroblocks with PSNRs below the threshold are selected for modulation. This is performed in an attempt to reduce the overall distortion caused by embedding a given message. The modulation is performed in a simple procedure where the message bits replace the least significant bits of selected motion vectors. Consequently, the frame-based threshold is sent in a separate channel or hidden in the I-frames of each Group of Pictures (GoP). As such, enabling the decoder to identify the modulated motion vectors and therefore correctly extract the message.

As with all data hiding solutions, each technique has its benefits and drawbacks. A potential drawback of the solution reported in [20] is the iterative algorithm in which the PSNR threshold is computed. At each iteration, the selected set of motion vectors is used for motion compensation to compute the prediction residuals. In each iteration, every inter-coded video frame is compressed, decompressed and the PSNR of individual macroblocks is computed. The algorithm starts with an initial PSNR threshold that is decremented by 1 (dB) until all of the selected motion vectors result in reconstruction quality that are less than the PSNR threshold. In the worst case scenario, the algorithm stops when a predefined minimum PSNR threshold is reached. Hence in the worst case scenario, the total number of iterations per video frame is equal to the difference between the initial and minimum PSNR thresholds. Additionally, once the thresholds are computed, they are

potentially hidden in the I-frames of each GoP as mentioned previously. This might result in increasing the total bitrate or decreasing the video quality.

In contrast, our proposed solution of motion vector modulation using matrix encoding, is a non-iterative solution which requires one pass of reencoding only. Additionally, there are no extra thresholds or metadata to be transmitted to facilitate correct extraction of the message bits. The experimental results section compares the two solutions further by means of message payload, resultant distortion and quality degradation.

5. Experimental results:

This section presents the experimental results using the proposed quantization scale and motion vector modulation. The proposed solutions are also compared against the existing work of [20] and [22].

5.1 Quantization scale modulation:

In this section, the proposed quantization scale modulation solution using multilayer encoding is assessed in terms of its effect on video quality and excessive bitrate. It is also assessed in terms of message payload and quantization scale histogram differences. The proposed solution is compared against the work reported in [22]. Thus, we use similar video encoding parameters. Namely, MPEG-2 video is used to code six different test sequences at a constant bitrate of 1.5 Mbit/s. For the proposed solution of multilayer encoding, the base layer is coded at 1Mbit/s and the enhancement layer is coded at 0.5Mbit/s. The test sequences are listed in Table 4.

Table 4. Video test sequences

ID	Sequence Name	#MBs /frame	Frames /sec
V1	Coastguard	396	30
V2	Container	396	30
V3	Flowergarden	330	30
V4	Foreman	396	30
V5	Mobile	396	30
V6	Hall monitor	396	30

All messages to be embedded are generated randomly with a uniform distribution. As mentioned previously, the message bits are divided into k bit segments, each segment is embedded using $2^k - 1$ quantization scales. In this section, we experiment with three values of the matrix encoding parameter, $k=2..4$. Hence the 3, 7 and 15 quantization scales are required for message embedding respectively.

The message payload per video sequence is reported in Table 5. The payload is measure in terms of Kbit/s. The reported values are less than or equal to their theoretical counterparts as reported in Table 3 above. This is so because skipped macroblocks do not carry quantization scales, therefore the higher the percentage of skipped macroblocks, the lower is the message payload. The values reported in Table 5 indicate that message payload is noticeably higher when the proposed solution is used. This conclusion is expected because two video layers are used, hence, the total number of quantization scales is roughly doubled. It is also shown in the table that when $k=2$, the message payload is higher. However this comes at the expense of excessive bitrate as more quantization scales are modulated.

Table 5. Message payload in Kbit/s

	$k=2$		$k=3$		$k=4$	
Seq. ID	Proposed	Reviewed	Proposed	Reviewed	Proposed	Reviewed
V1	15	7.5	9.6	4.8	6.0	3.0
V2	14.3	7.2	9.1	4.6	5.7	2.9
V3	12.3	6.2	7.8	3.9	4.9	2.5
V4	15.4	7.7	9.8	4.9	6.2	3.1
V5	14.8	7.4	9.4	4.7	5.9	2.9
V6	15.4	7.8	9.8	5	6.2	3.1
Avg	14.5	7.3	9.3	4.6	5.8	2.9

The effect of message embedding on the total bitrate of coded video is reported in Table 6. The excessive bitrate is reported as a percentage of the original video size without message embedding. Again, the results are reported for $k=2..4$. Part (a) of the Table reports the results of the proposed solution. The label “Encoding” means that the original raw video is available for re-encoding using a multilayer encoder. The label “Transcoding” means that the original video is not available, therefore a multilayer transcoder is used as previously introduced in Figure 3 of Section 2. Part (b) of the Table reports the reviewed results of [22]. The label “Encoding” means that the original video is available for re-encoding using a single layer encoder, and the label “Scaling” means that the original raw video is not available hence the DCT coefficients are rescaled according to the new values of the quantization scales as reported in [22].

The results in the table indicate that the percentage increase in the video bitrate is higher when $k=2$. The results also indicate that by using the proposed multilayer transcoding solution, the average total bitrate has actually decreased by -0.1% when $k=3$ and -1.3% when $k=4$. On the other hand, when $k=3$, the average increase in bitrate of the reviewed solution is 1.3% when the original raw video is

available and 1.6% otherwise. Notice that the change in bitrate is mainly caused by the sub-optimal histogram preserving algorithm used in the modulation of the quantization scales. The quantization scales are incremented or decremented accordingly without taking the resultant bitrate into account.

Table 6. Percentage increase in bitrate due to message embedding.

	$k=2$		$k=3$		$k=4$	
Seq. ID	Encoding	Transcoding	Encoding	Transcoding	Encoding	Transcoding
V1	4.1	-1.5	2.9	-2.7	2.3	-3.4
V2	11.6	9.2	7.0	4.1	4.4	1.3
V3	3.3	-4.3	2.5	-4.9	2.2	-5.6
V4	4.7	1.2	3.4	-0.1	2.6	-0.8
V5	3.1	-0.1	2.5	-0.7	2.0	-1.0
V6	9.7	7.4	5.9	3.5	3.7	1.5
Avg	6.1	2.0	4.0	-0.1	2.9	-1.3

(a) Results of the proposed solution

	$k=2$		$k=3$		$k=4$	
Seq. ID	Encoding	Scaling	Encoding	Scaling	Encoding	Scaling
V1	1.3	1.4	0.7	0.8	1.2	1.2
V2	2.7	3.6	1.5	2.4	0.9	0.9
V3	0.7	0.8	0.6	0.7	1.1	1.2
V4	2.5	2.8	1.3	1.6	1.5	1.5
V5	0.5	0.5	0.3	0.3	1.2	1.3
V6	7.5	7.9	3.8	3.9	1.5	1.6
Avg	2.5	2.8	1.3	1.6	1.2	1.3

(b) Results of the reviewed solution

In part (a) of the table, it is shown that the percentage increase in bitrate caused by the multilayer encoding solution is higher than that of the proposed transcoding solution. The reason behind this increase is that quantization scales can be reduced during the modulation process. When applied to the multilayer encoding solution, a smaller quantization scale results in higher coding quality at the expense of higher bitrate. On the other hand, when applied to the transcoding solution, taking into account that the input is a pre-encoded video, the coding quality cannot be increased because of the existence of quantization error. In other words, a smaller quantization scale in this case can result in a different coefficient value but does not result in decreasing the number of zero DCT coefficients, hence, the bitrate is unlikely to increase.

As mentioned previously, the histogram of the quantization scales is affected because of message embedding. The proposed sub-optimal histogram preserving algorithm illustrated in Figure 4 is

intended to minimize the difference between the histograms of the original and modified quantization scales. We have applied both the proposed and the reviewed [22] sub-optimal histogram preserving algorithms on the proposed multilayer encoding solution with $k=3$ and $k=4$. The histogram distance is quantified using Equation 2 above. The results are shown in Table 7. It is shown that despite the simplicity of the proposed algorithm, it results in histogram differences that are less than the reviewed algorithm for all of the test sequences. Additionally, the histogram differences become smaller as k increases. This is clear because as k increases, the message payload decreases, thus the percentage of modified quantization scales decreases as well.

Table 7. Histogram differences due to message embedding.

	$k=3$		$k=4$	
Seq. ID	Proposed	Reviewed	Proposed	Reviewed
V1	11.6	15.2	11.5	14.5
V2	39.0	42.9	25.4	25.7
V3	18.8	24.5	16.1	20.8
V4	6.6	14.4	9.0	13.0
V5	15.4	24.9	13.5	18.3
V6	55.8	58.1	34.0	34.6
Avg	24.5	30.0	18.2	21.1

Lastly, the effect of message embedding on the video's quality is assessed in terms of PSNR. The PSNRs of the videos that embed a message are compared against the PSNRs of message free videos in Table 8. Part (a) reports the PSNR results of the proposed solution with $k=2..4$. Part (b) reports the PSNR results of the reviewed solution with $k=2..4$ as well. It is observed from the tables that the PSNRs of the original video in Part (a) has a slightly higher quality than its counterparts in Part (b). This is so because multilayer encoding is used in Part (a). More specifically, in SNR multilayer encoding of MPEG-2, the dequantized coefficients of the enhancement layer are fed into the motion compensation loop of the first layer forming a higher quality of image perdition sources, this is well understood in the multilayer video encoding literature. It was also noticed that the PSNR results are very similar for the cases when $k=2..4$. This is an acceptable result given that the difference between $k=2..4$ was mainly manifested in the bitrate of the modified videos as reported in Table 6.

Table 8. Effect of message embedding on video's PSNR in dB.

Seq. ID	Original	Encoding	Transcoding
V1	34.3	34.3	33.9
V2	40.0	40.1	39.5
V3	27.2	27.2	26.7
V4	37.0	37.1	36.7
V5	28.9	28.9	28.5
V6	39.4	39.5	39.1
Avg	34.5	34.5	34.1

(a) Proposed solution with $k=2,3$ and 4.

Seq. ID	Original	Encoding	Scaling
V1	34.2	34.2	34.0
V2	39.4	39.4	39.4
V3	27.3	27.3	27.2
V4	36.7	36.7	36.5
V5	28.5	28.5	28.4
V6	38.4	38.4	38.3
Avg	34.1	34.1	34.0

(b) Reviewed solution with $k=2,3$ and 4

The tables also show that the proposed solution using multilayer encoding did not result in noticeable PSNR degradations. This observation is also true for both of the reviewed solutions. The results also show that that the proposed solution of multilayer transcoding results in slight drop of 0.4 dB on average. However, it is understood that PSNR differences below 0.5 dB are subjectively negligible.

5.2 Motion Vector Modulation:

For a fair comparison between the proposed motion vector modulation solution and existing work, we use the same video sequences and coding parameters as reported in [20]. The following sequences are used at a frame rate of 25Hz and coded using MPEG2 VBR coding: Car-phone (176x144 pixels, 270 frames, average bitrate of 424Kbit/s), Foreman (352x288 pixels, 135 frames, average bitrate of 1150Kbit/s), Football (352x240 pixels, 117 frames, average bitrate of 1600 Kbit/s), Coastguard (352x288 pixels, 270 frames, average bitrate of 1660Kbit/s), Mobile (352x288 pixels, 270 frames, average bitrate of 1830 Kbit/s) and Flower-Garden (352x288 pixels, 250 frames, average bitrate of 1670 Kbit/s). The GoP structure used is N=9 and M=3 (total of 9 pictures per

GoP with 2 B-frames between reference frames). Full motion estimation range is used in both the reviewed and the proposed work.

The message payloads of the proposed and the reviewed work are reported in Table 9. The payload of the reviewed work varies from one sequence to the other because different adaptive thresholds are computed for different sequences as explained in Section 4. The payload of the proposed work on the other hand, has lower variance and depends mainly on the value of the matrix encoding parameter k and the percentage of skipped macroblocks. Notice that the payloads of the two approaches of the proposed solution (i.e. transcoding and 2-pass encoding) are similar because both approaches reuse the motion vectors and coding parameters of the decoded input video. In both approaches, at $k=2$, the average message payload is 17.1 Kbit/s. At $k=3$ the payload drops to 10.9 Kbit/s. Lastly, at $k=4$ the payload drops to 5.7 Kbit/s. However at a higher value of k , the distortion caused by message embedding is reduced as evident in Tables 10 and 11.

Table 9. Message payload in Kbit/s

		$k=2$	$k=3$	$k=4$
Sequence	Reviewed	Proposed		
Carphone	3.3	5.0	3.1	1.9
Foreman	8.4	19.4	12.4	1.7
Football	16.2	20.0	12.7	7.9
Coastguard	13.4	19.3	12.3	7.6
Mobile	14.7	19.4	12.4	7.6
Flowergarden	15.7	19.4	12.4	7.6
Avg	11.9	17.1	10.9	5.7

The excessive bitrate caused by message embedding is reported in Table 10. We report the results of the reviewed work, the proposed 2-pass encoding and the proposed transcoding solutions with $k=2..4$. The excessive bitrate is reported as a percentage of the original video size without message embedding. It is shown in the table that the excessive bitrate is reduced when k increases from 2 to 4. This is expected because the message payload is reduced with higher values of k . It is also shown in the table, that the average excessive bitrate when $k=3$ and $k=4$ is noticeably lower than that of the reviewed work.

Table 10. Percentage increase in bitrate due to message embedding.

		$k=2$	$k=3$	$k=4$	$k=2$	$k=3$	$k=4$
Sequence	Reviewed	2-pass encoding			Transcoding		
Carphone	3.29	22.2	11.2	5.0	20.9	10.8	5.0
Foreman	8.44	18.9	9.6	4.5	15.2	7.5	3.5
Football	16.17	4.1	2.0	1.0	3.0	1.4	0.7
Coastguard	13.35	16.9	8.4	4.0	13.7	6.6	3.2
Mobile	14.7	21.9	11.2	5.3	17.2	8.4	3.9
Flowergarden	15.7	23.2	11.6	5.3	18.8	9.0	4.1
Avg	11.94	17.9	9.0	4.2	14.8	7.3	3.4

The effect of message embedding on the average drop in PSNR is reported in Table 11. Noteworthy are the results of the 2-pass encoding with $k=2..4$. In all cases the average drop in PSNR is constantly less than 0.5 dB. In the transcoding approach on the other hand, it is expected that the average drop in PSNR to be higher as shown in the table. This is so because there are two factors affecting the drop of PSNR in this case. The first factor is the modulation of the motion vectors and the second factor is the transcoding of pre-encoded video. Such pre-encoded video contains quantization errors from the original encoder and result in further quality loss. It is interesting to observe that as the value of k increases, less motion vectors are modulated and the drop in PSNR becomes mainly due to the transcoding of pre-encoded videos.

Table 11. Drop in average PSNR due to message embedding.

		$k=2$	$k=3$	$k=4$	$k=2$	$k=3$	$k=4$
Sequence	Reviewed	2-pass encoding			Transcoding		
Carphone	0.4	0.26	0.14	0.1	1.21	0.72	0.4
Foreman	0.22	0.30	0.17	0.1	1.05	0.61	0.3
Football	0.29	0.11	0.05	0.0	0.57	0.30	0.2
Coastguard	0.27	0.19	0.10	0.1	1.05	0.60	0.3
Mobile	0.6	0.32	0.17	0.1	1.33	0.80	0.4
Flowergarden	0.55	0.26	0.14	0.1	1.47	0.88	0.5
Avg	0.39	0.24	0.13	0.07	1.11	0.65	0.36

To summarize, the proposed motion vector modulation solution exhibited enhanced results over the reviewed work of [20] in terms of reduced excessive bitrate and to a lesser extent, in terms of message payload. The difference in terms of average drop in PSNR is not significant though. On the

other hand, as mentioned previously, the proposed solution has an advantage of being a non-iterative solution and does not require sending additional information to the decoder to facilitate message extraction.

6. Conclusion:

The paper proposed the use of multilayer video coding and transcoding for message embedding using matrix encoding. In comparison to single layer coding, it was shown that such a solution roughly doubles the message payload whilst attenuating the video distortion and excessive bitrate. It was also shown in the paper that matrix encoding can be used for modulating motion vectors. The message payload in this case is high because of the abundance of motion vectors in a coded video bitstream. However, In comparison to quantization scale modulation, modifying the values of motion vectors is likely to result in less efficient motion compensation leading to either excessive bitrate or lower coding quality.

References:

- [1] I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker, "Digital Watermarking and steganography," 2nd Edition, Morgan Kaufmann, November, 2007.
- [2] D. Xu, R. Wang, J. Wang, "A novel watermarking scheme for H.264/AVC video authentication, Signal Processing: Image Communication, 26(6), pp. 267-279, July, 2011.
- [3] L. Tian, N. Zheng, J. Xue, C. Li, X. Wang, "An integrated visual saliency-based watermarking approach for synchronous image authentication and copyright protection," Signal Processing: Image Communication, 26(8-9), pp. 427-437, October, 2011.
- [4] P.C. Su, C.-S. Wu, I.-F. Chen, C.-Y. Wu, Y.-C. Wu, "A practical design of digital video watermarking in H.264/AVC for content authentication," Signal Processing: Image Communication, 26(8-9), pp. 413-426, October, 2011.
- [5] S. Emmanuel, A. Vinod, D. Rajan, C.K Heng, "An Authentication Watermarking Scheme with Transaction Tracking Enabled," In Proc. Digital EcoSystems and Technologies Conference, 2007. Inaugural, 21-23 February, 2007.
- [6] F.C. Chang, H.C. Huang and H.M. Hang, "Layered access control schemes on watermarked scalable media," Journal of VLSI Signal Processing, 49(2007), pp. 443-455, 2007.
- [7] A. Yilmaz, A. Alatan, "Error detection and concealment for video transmission using information hiding," Signal Processing: Image Communication, 23(4), pp. 298-312, April, 2008.

- [8] P. Campisi, D. Kundur, D. Hatzinakos, and A. Neri, "Compressive data hiding: An unconventional approach for improved color image coding," *EURASIP Journal of Applied Signal Processing*, 2002(2), pp. 152–163, 2002.
- [9] M. Chaumont and W. Puech, "A DCT-based data-hiding method to embed the color information in a JPEG grey level image," in *Proceedings of EUSIPCO*, Italy, September, 2006.
- [10] J.-M. Thiesse, J. Jung, M. Antonini, "Rate Distortion Data Hiding of Motion Vector Competition Information in Chroma and Luma Samples for Video Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, 21(6), pp.729-741, June, 2011.
- [11] Recommendation ITU-T H.264 | ISO/IEC 14496-10:2009, "Advanced Video Coding for generic audio-visual services," March 2009.
- [12] Y. Hu, C. Zhang and Y. Su, "Information Hiding Based on Intra Prediction Modes H.264/AVC," *IEEE International Conference on Multimedia and Expo, ICME 2007*, pp.1231-1234, July, 2007.
- [13] G. Yang, J. Li, Y. He and Z. Kang, "An information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream," *International Journal of Electronics and Communications*, (65)4, pp. 331-337, April, 2011.
- [14] S. M. Kim, S. B. Kim, Y. Hong, and C. S. Won, "Data hiding on H.264/AVC compressed video," in *Proc. ICIAR*, Montreal, Canada, pp. 698–707, August, 2007
- [15] T. Shanableh, "Data hiding in MPEG video files using Multivariate Regression and Flexible Macroblock Ordering," *IEEE Transactions on Information Forensics and Security*, 7(2), November, 2011.
- [16] Y. Li, H.-X. Chen, Y. Zhao, "A new method of data hiding based on H.264 encoded video sequences," *IEEE International Conference on Signal Processing, ICSP 2010*, pp.1833-1836, October, 2010.
- [17] F. Jordan, M. Kutter, and T. Ebrahimi, "Watermarking Technique for Hiding/ Retrieving Data in Compressed and Decompressed Video," document ISO/IEC JTC1/SC29/WG11, MPEG97, July, 1997.
- [18] D.-Y. Fang, L.-W. Chang, "Data hiding for digital video with phase of motion vector," *IEEE International Symposium on Circuits and Systems, ISCAS 2006*, September 2006.
- [19] C. Xu, X. Ping, and T. Zhang, "Steganography in compressed video stream," *International Conference on Innovative Computing, Information and Control, ICICIC'06*, vol. II, pp. 803–806, 2006.

- [20] H. Aly, "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error," IEEE Transactions on Information Forensics and Security, 6(1), pp.14-18, March, 2011.
- [21] K. Wong, K. Tanaka, K. Takagi and Y. Nakajima, "Complete Video Quality-Preserving Data Hiding," IEEE Transactions on circuits and systems for video technology, 19(10), October 2009.
- [22] K. Wong and K. Tanaka, "A Data Hiding Method Using Mquant in MPEG Domain", Journal of the Institute of Image Electronics Engineers of Japan, 37(3), pp.256-267, 2008.
- [23] F. Battisti, M. Carli, A. Neri, "Video error concealment based on data hiding in the 3D wavelet domain," Proc. of 2nd European Workshop on Visual Information Processing (EUVIP), Rome, Italy, July, 2010.
- [24] Ron Crandall, "Some Notes on Steganography," 1998. Available online <http://whitepapers.hackerjournals.com/wp-content/uploads/2010/03/Some-Notes-on-Steganography1.pdf>
- [25] H. Schwarz, D. Marpe and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard", IEEE Trans. on Circuits and Systems for Video Technology., 17(9), pp. 1103-1120, 2007.
- [26] T. Shanableh and M. Ghanbari, "Multilayer Transcoding with format portability for multicasting of single-layered video," IEEE transactions on multimedia, 7(1), 1-15, February, 2005
- [27] J. De Cock, S. Notebaert, P. Lambert, R. Van de Walle, "Architectures for Fast Transcoding of H.264/AVC to Quality-Scalable SVC Streams," IEEE Transactions on Multimedia, 11(7), pp.1209-1224, November, 2009.
- [28] A. Westfeld. "F5-A Steganographic Algorithm: High Capacity Despite Better Steganalysis," 4th International Workshop on Information Hiding, Pittsburgh, USA, April, 2001.

Biography:



Tamer Shanableh earned his Ph.D. in Electronic Systems Engineering in 2002 from the University of Essex, UK. From 1998 to 2001, he was a senior research officer at the University of Essex, during which, he collaborated with BTextact on inventing video transcoders. He joined Motorola UK Research Labs in 2001. During

his affiliation with Motorola, he contributed to establishing a new profile within the ISO/IEC MPEG-4 known as the Error Resilient Simple Scalable Profile. He joined the American University of Sharjah in 2002 and is currently an associate professor of computer science. Dr. Shanableh spent the summers of 2003, 2004, 2006, 2007 and 2008 as a visiting professor at Motorola multimedia Labs. His research interests include digital video processing and pattern recognition.