Authentication of Topographic EEG: Employing Transform Based Watermarking

Rafi Ullah and Hani Ali Alquhayz
Department of Compute Science and Information, College of Science at Al-Zulfi, Majmaah University, Saudi Arabia

Abstract
With the use of advanced communication systems, attackers can easily tamper with data without being detected. In this paper, the wavelet-based watermarking method has been used to ensure the secure transformation of topographic EEG data. Unlike other more conventional approaches, a blind semi-fragile watermark is embedded in the estimated motions of topographic EEG. Instead of using conventional block-based approaches, this approach concisely determines the altered areas. An efficient trade-off has been achieved in the three most common conflicting properties of watermarking: payload, distortion rate, and robustness. The proposed approach is able to survive legitimate manipulations such as JPEG compression. The simulation results demonstrate the performance of our algorithm.

Keywords
EEG topo-maps, brain, motion vectors, block-based motion estimation (BMA), watermark.

1. Introduction

Due to advanced communication technology, attackers today can easily access digital content such as images, videos, graphics, text and even bio-medical data, without leaving any traces. Thus, protection of digital content is a key issue for researchers. In the last two decades, data hiding techniques have been used to ensure the secure transformation of digital content on any communication channel, either wired or wireless. Digital watermarking is one of the most commonly used data hiding techniques. It is used for several purposes including authentication, copyright protection, fingerprinting, broadcast monitoring, ownership assertion, and transaction tracking. The watermark can be used either blind or informed. In the blind watermark process, there is no need for there to be the original or any part of the original content on the verification side [1]. Alternatively, in informed watermarking, the original content is required on the verification side. This paper has used the blind watermark for authenticating biomedical data, i.e. EEG data. The significance of the watermarking properties depends upon the requirements of any particular application, i.e. biomedical data, in this case. These properties are perceptual similarity, robustness, data payload and embedding effectiveness. These properties are conflicting and the authors have had to make a trade-off between these properties according to the particular application. Effectiveness is the probability of detecting the watermark. The ultimate goal is to achieve 100%, but it is often not possible because of conflict with perceptual similarity. Sometimes, the fidelity of the system can be sacrificed for better performance with respect to other characteristics, such as higher robustness. Similarly, robustness depends upon the application and usually, we need robustness in copyright protection where perceptual similarity can be sacrificed. Watermark (wm) strength with respect to two other contradicting parameters is illustrated in Figure 1. The curves shown are theoretical and the increase in one parameter will affect the other. These parameters are inversely proportional and we have to make a trade-off between these parameters according to the application.

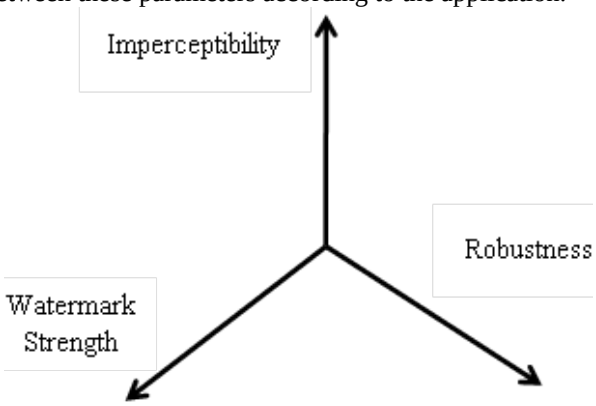


Figure 1. The three intrinsic and contradicting parameters of a watermarking system: imperceptibility, robustness, and watermark strength

There are different watermarking techniques that can be categorized by the following conditions:

- Watermark embedding locations
- Watermark type (fragile, robust or semi-fragile: as will be discussed later in this section)
- Robustness against different attacks
- Domains for watermark embedding
- Watermark is additive or coefficients' modification will represent the presence of the watermark
- Watermark is blind or informed
- Strength of the watermark
- Behaviour of tamper detection and localization

The three commonly used watermarks in the literature are robust, fragile and semi-fragile watermarks and they are

Manuscript received March 5, 2017
Manuscript revised March 20, 2017

application dependent. A robust watermark is able to survive attacks, even those that are illegitimate, and is generally used for copyright protection. A fragile watermark is not able to survive a legitimate attack such as JPEG compression. It cannot survive even a file format change, i.e. converting TIFF to BMP. It is normally used for authentication purposes. The semi-fragile watermark is robust against legitimate attacks such as JPEG compression and fragile against illegitimate attacks such as geometric transformations. In the last few years, researchers have been using a reversible watermarking technique, which is able to extract and retrieve both the embedded information and the original content.

2. Literature Survey

2.1 Existing techniques

In [2, 3], mid wavelet frequencies were used to protect the copyright of the content. In [4], the authors used a robust watermark that was based on the pattern recognition using radon transform. This was used to protect the copyright of digital images. Fragile watermarks are usually used for authenticating digital content. In the previous literature, many fragile watermarking techniques based on wavelet transform, quantization index modulation (QIM), hashing, non-deterministic block wise dependence, and image structure [5-9] were used to authenticate the digital content. Since there are very common friendly attacks, such as JPEG compression, which is especially important for communication channels, it is necessary to secure the content in such a way that it can survive this type of friendly manipulation. In the literature, many semi-fragile watermarking techniques [10-15] have been used, which are able to tolerate the legitimate attacks according to the requirements of the application. The flexible parameters are set according to the attack strength, i.e. JPEG compression strength. Wavelet and DCT-based semi-fragile watermarking techniques were used in [13, 16] and were able to survive JPEG lossy compression up to the defined extent. In [17], the authors used histogram-based reversible watermarking. The authors in [18, 19] used reversible watermarking to authenticate the images for 3D cameras.

In [34], the authors used a web-based watermarking method to protect the EEG signals. The spatial information was used for data hiding but without recovery of the data after distortion. They were unable to survive lossy compression and distinguish the legitimate/illegitimate attacks. In [35], the authors used wavelet-based watermarking to authenticate the EEG data. The imperceptibility of the watermark was much better, but the algorithm was unable to survive against compression. They were unable to classify attacks, or recover data after tampering. The comparison of the proposed approach with [34] and [35] is given in Table 3.

In this paper, we used a twenty-nine topo-maps sequence that collectively represented a five-second video. In normal videos, there is almost no inter-frame correlation whenever there is a scene change. The two consecutive frames where the scene changes in a video are completely different from each other, i.e. these frames are not comparable. On the other hand, the topo-maps (frames sequence) are correlated in some way and this correlation is either strong or weak, i.e. every frame is comparable in some way to its successive frame. We were interested in watermark embedding in the motion vectors of the block-based motion estimation of these frames. The motion vectors were created according to the movements in the topo-maps sequence. Each EEG topo-map was watermarked by embedding a wavelet-based Huffman coded watermark. The watermark is semi-fragile and compressed using the lossless Huffman coding. By compressing the watermark, its strength becomes low and consequently it increases the imperceptibility. Although, in biomedical data transmission, compression is discouraged as it is very sensitive data and it should be transferred in its original form without losing any information. However, we have used a semi-fragile watermark that made our system robust against the JPEG compression to some extent, i.e. the architecture of our system is open for friendly manipulations. The sender can compress the EEG topo-maps before transmission according to his requirements. The general architecture of the watermarking system is given in Figure 2.

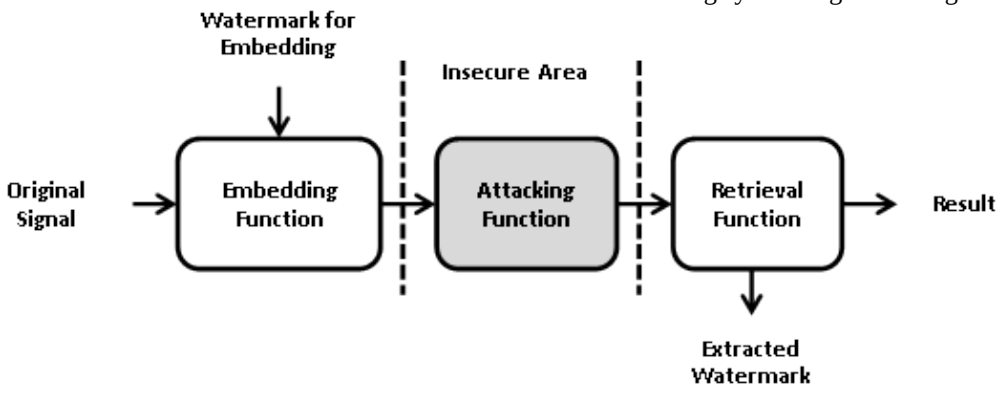


Figure 2. Block diagram of digital watermarking system

2.2 Block-based motion estimation techniques:

Normally, motion vectors are used to compress the video by eliminating several frames based on spatio-temporal correlation. We were not interested in eliminating any frame from the frames sequence. We created motion vectors, which demonstrated the movements in these frames and then embedded the watermarks in the motion vectors' amplitudes. Several BMAs can be used to create the motion vectors to estimate the motion. In our experiments, we used only full search BMA for its better performance compared to others such as the three step search (TSS), new three step search (NTSS), four step search (FSS), and diamond search (DS) [20].

2.3 Measurements:

Peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) were used to measure the performance of watermark strength and its embedding. PSNR and SSIM are the ratio in maximum power of original frame and the watermarked/degraded frame. There is no precise rule to select either PSNR or SSIM measurement for evaluating the quality of the watermarked image/frame. PSNR is the ratio in maximum power of frame and the degraded (by embedding the watermark) frame, i.e. noise induced in signal. On the other hand, SSIM illustrate and investigate the structural similarity between the original and the watermarked frame. The authors in [21] briefly compared PSNR and SSIM for common degradations, such as Gaussian noise, JPEG compression. Some studies discovered the advantages and disadvantages of PSNR and SSIM [22-24]. The value of SSIM for same images is one e.g. $SSIM(im1, im1) = 1$. Equation 1 and Equation 2 are used to define PSNR and SSIM, respectively.

$$PSNR = 20\log_{10}(255^2 / MSE) \quad (1)$$

where MSE is the mean square error and can be defined as,

$MSE = 1/N^2 \sum_{i=0}^{N}\sum_{J=0}^{N}(C_{ij} - R_{ij})^2$ and 255 is the highest value in the frame(s).

where N is the size of block. C_{ij} and R_{ij} are the pixels that are being compared in original and watermarked frames, respectively.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c} \quad (2)$$

Where x and y are two windows of size $N \times N$. μ_x and μ_y are the averages of x and y respectively. σ_x^2 and σ_y^2 are the variances of x and y respectively. $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L is the maximum value i.e. $2^{\#bites}$, and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

The remainder of this paper is organized as follows: in Section 3, we explain the proposed algorithm. In Section 4, we discuss the recovery of the watermark and Section 5 presents the results and discussions. In Section 6, we conclude the paper and present some future directions.

3. Proposed Algorithm

Several watermarking methods have been proposed for hiding data in video, where each single frame in the video is considered as a still image [25, 26]. In [27], the author used a phase angle of the motion vectors of the blocks in the inter-frame. This can be applied to either compressed or uncompressed videos. An effective watermarking scheme is proposed in this paper, where the watermarks are embedded in inter-frame motion vectors instead of using the frames themselves. We extended our watermarking approach from frames in normal videos to topographic EEG (topo-maps sequence). The frames correlation is complex in topographic EEG as compared to the correlation of frames in normal videos. The block diagram for the proposed algorithm (watermark generation and embedding) is given in Figure 4.

3.1 Watermark generation

Instead of using the very common watermarks such as Logo or Signature, we generated the watermarks from the frames of the sequence. The frames were highly compressed and then embedded in their respective motion vectors. For compression, Huffman coding was applied to make the frame strength very low and recoverable. Thus, the proposed approach is a self-recovery approach and it is possible to retrieve the original frame after alteration (attack). The following procedure was used for generating the planned watermarks according to the frames whose motion vectors were used for watermark embedding. Actually, the watermark to be embedded is a highly compressed version of the original frame. The same procedure was then repeated for every watermark to be embedded in the respective inter-frame motion vectors.

- Integer wavelet transform (IWT) was applied on the original frame.

- Integer discrete cosine transform (IDCT) was then applied on the approximation ($LL1$ sub-band) of the frame. The other wavelet sub-bands, $LH1$, $HL1$ and $DD1$ were horizontal, vertical, and diagonal details of the frame, respectively.

Further, Huffman coding was used to reduce the strength of the IDCT transformed coefficients. The Huffman coder reduction ratio of input to output was about $4.5:1$. For example, in one of our experiments, the input bits were $256*256*8 = 524288$ and the resultant Huffman coder was 120000. The compression ratio varied according to the input image (topo-frame in this case). More textured regions will reduce the compression ratio and vice versa. The compression ratio of every topo-map is given in Figure 3.

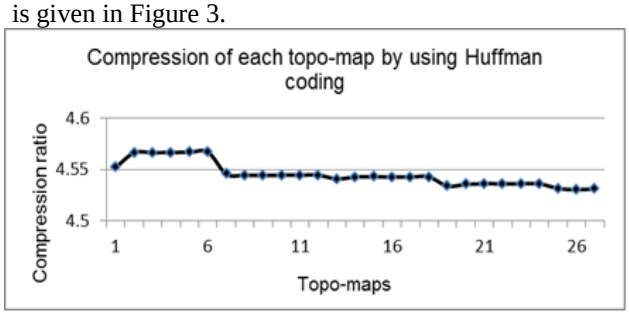


Figure 3. Compression ratio of each topo-map of the input sequence

- The Huffman coded coefficients were then passed through a BCH encoder. This encoder added extra bits to the Huffman coded coefficients that were used for recovering the watermarked bits when they were affected by some legitimate/illegitimate manipulations. Although the BCH encoder increases the watermark strength and consequently decreases the perceptual similarity, the BCH encoded bits can help us to retrieve the original frame even after malicious manipulations. An example of one of the BCH coding pairs is $(31, 21, 2)$, where 21 are actual bits, 31 are physical bits and 2 bits can be corrected for every 21 bits [28]. When the ratio of the actual and physical bits is increased, the watermark payload will increase and ultimately, the imperceptibility will be decreased, and vice versa. Thus, use of a BCH pair is flexible for making a trade-off in robustness and imperceptibility. Table 1 shows the BCH encoder and decoder. The proposed approach survived JPEG lossy compression to some extent. In the case of recovering the watermark after the frame is attacked, it could not be recovered beyond the 10% loss after JPEG compression for the BCH layer (31, 16, 3) used in our experiment.

Table 1. Generator of primitive BCH codes. The symbol n indicates actual bits, k indicates physical bits and t are the bits that can be recovered after compression

n	k	t	n	k	t
7	4	1	255	171	11
15	11	1		163	12
	7	2		155	13
	5	3		147	14
31	26	1		139	15
	21	2		131	18
	16	3			
	11	5		123	19
	6	7			
63	57	1		115	21
	51	2			
	45	3		107	22
	39	4			
	36	5		99	23
	30	6			
	24	7		91	25
	18	10			
	16	11		87	26
	10	13			
	7	15		79	27

- The obtained coefficients were then $XORed$ with the random sequence and then permuted by using secret keys. This step was used to make the watermark secure by itself. These keys should be available on the receiving side.
- The result was the secure Huffman coded watermark that was ready to be embedded in the suitable locations of the respective inter-frame motion vectors. Embedding in suitable locations makes the watermark secure, imperceptible and robust against legitimate attacks. The embedding space also depends on the size of blocks. The frames are divided into blocks in BMAs, where each block represents a motion vector. If the size of the block is small, then there are a large number of motion vectors in the frame (topo-map) and vice versa. However, the computational cost varies according to the block size, i.e. a large number of computations is required while dividing the frame into small blocks and vice versa.
- This semi-fragile watermark is also referred to as a recovery watermark, because it can be used to recover the original frame/topo-map after alteration. Actually, the watermark is a highly compressed version of the original frame. The recovery of the original image will be discussed in Section 4.

As we used the BCH encoder to add some extra bits to make the watermark robust against manipulations, we had to reduce the watermark strength, which had a very high payload because of the BCH encoded extra bits. This issue can be resolved by using one watermark for two inter-frame motion vectors. Based on the secret key, half of the frames are selected randomly from the entire frames sequence, and the watermarks are generated from each of the selected frames using a watermark generation procedure. Every

watermark is then embedded in two inter-frame motion vectors. This will reduce the watermark strength by 50%. The random selection is key-based and this key should be available at the receiving end. This key, along with other two keys, which are used in watermark generation, can be considered as a shortcoming of this algorithm.

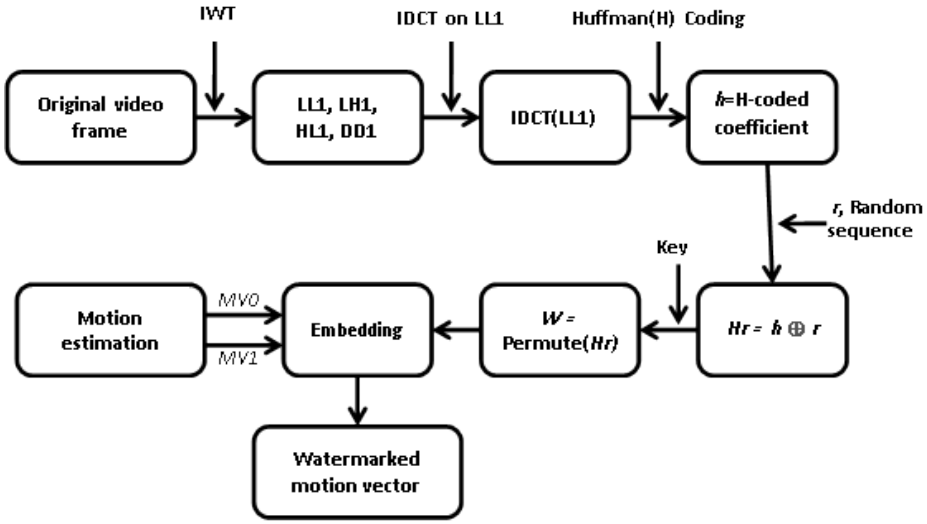
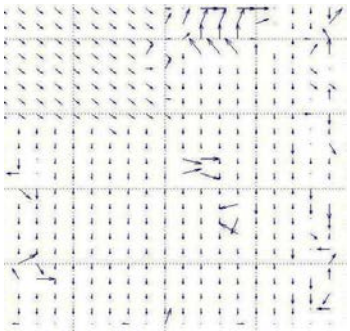


Figure 4. Block diagram for generating and embedding watermark

3.2 Watermark embedding and extraction

The planned watermark was embedded in the motion vectors in such a way that it should survive a legitimate alteration such as JPEG compression, although we were not interested in compressing our data, as EEG data is very sensitive and compression may lose some important information. Unlike the traditional use of digital image watermarking, we used motion vectors of the topographic EEG to embed the watermarks, which would not affect the quality of the topo-maps as much as in common grayscale/colour images. Embedding in the motion vectors preserved the quality of individual topo-maps, as we embedded the watermarks in the temporal information of the EEG topographic sequence, i.e. the embedding distortion was low.

Unlike the frame sequence in common video, the EEG topo-maps sequence has a complex combination of expansion/shrinking along with slight scaling and shifting. Since these frames have a large number of motion vectors, when compared to normal video, and mostly represent the area expansion and shrinking instead of object movements in the traditional video. Thus, motion vectors produced from EEG topo-maps have the capability to hide a large number of watermark bits, i.e. we had a large amount of space in the form of motion vectors to hide extensive watermark bits. Consequently, the imperceptibility and robustness of the watermark was increased in the topographic EEG instead of normal videos. Figure 5 shows the samples from the normal video and topo-maps sequences, and their respective motion vectors. The motion vectors for topo-maps have a large number of motion vectors and thus it has the capability of storing the watermarks with high payload.



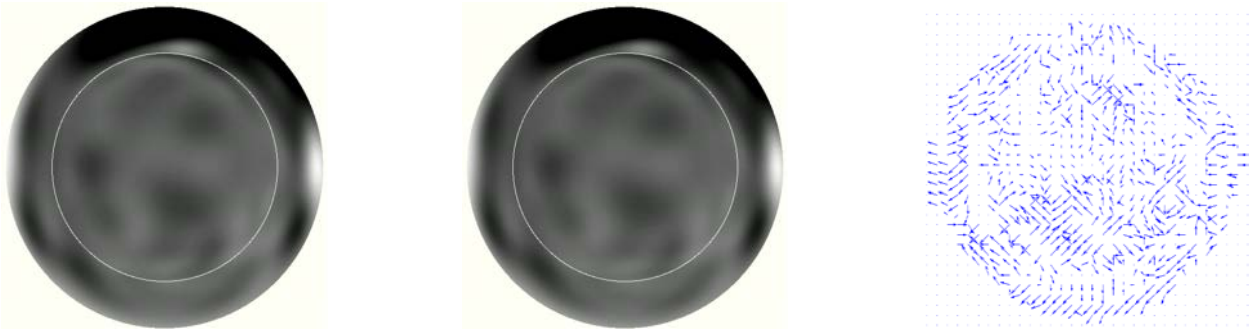


Figure 5. Two consecutive frames are selected from a normal video and a topographic EEG in the first and second rows, respectively. Both have their related motion vectors.

One of the BMAs was used to create the motion vectors. In this paper, the ES/FS BMA has been used to estimate the motion and generate the motion vectors. In ES/FS, each block in the search window is compared with the current block based on one of the comparison criterion, such as mean square error (MSE), mean absolute difference (MAD), displayed frame difference (DFD), and the best match is obtained. In this paper, we used MSE to calculate the horizontal/vertical differences between the current block with the entire blocks of the search window. The final motion vector was then obtained using the Pythagorean law given in Equation 3.

$$c = \sqrt{a^2 + b^2} \qquad (3)$$

where a and b are the horizontal and vertical distances, respectively.

ES/FS is better in terms of quality and algorithm simplicity, but it requires a large number of computations [29]. Although other BMAs are computationally more efficient than ES/FS, they do not give as good quality as ES/FS. In this work, we were not interested in reducing the computational cost.

The candidate motion vectors (CMV) were selected for embedding the watermark bits under a certain threshold. Selection of CMV is the main issue in several existing approaches. In [30, 31], the CMV is selected based on the magnitude. On the other hand, in [27, 32], the CMV selection is based on the magnitude and phase between the consecutive motion vectors, i.e. error. The block error associated with the large CMV is expected to be large and thus, this idea has been extended for watermark embedding in the topo-maps. The CMV is based on vector magnitude (mv) and associated error (E), not mv alone [33].

After producing the pair (mv,E), the watermark bits to be embedded were embedded and the resulting pair became ($\widetilde{mv}$,$\widetilde{E}$). The CMV was selected for watermark embedding, if the associated block prediction error measured in PSNR was below the given threshold value $\mathcal{T}$. The significant bits were modified and we kept the embedded watermark robust against legitimate attacks such as JPEG compression and fragile against illegitimate attacks. Embedding in the least significant bits made the watermark fragile. Hence, we selected the suitable coefficient for watermark embedding that was able to survive the legitimate attacks. The generation and embedding of the watermark is given in the following algorithm.

The high strength watermark was generated from the image sequence. Unlike the normal videos, the capacity was large enough, thus the strength did not affect the perceptual shape of the video. Because of self-embedding, the frames can be recovered up to some extent, even after malicious attacks.

3.3 Watermark embedding

We embedded the watermark as follows:

Extract the motion vectors from the video. In this research, we used the EEG topo-maps sequence. The vectors are large in number as compared to normal videos. In our experiment, we used the five-second video which had 29 frames, with a frame size $of\ 544 \times 538$. Around six frames run per second. The macro block size was 16×16 and the frames distance was three. We kept the ratio fast, as it did not affect the performance of our algorithm. For calculating the motion vectors, we fixed the macroblock size according to the application. In EEG video (topo-maps sequence), we kept the macroblock size high as the movement was significantly high as compared to the normal videos. Due to the high capacity in EEG data, we were able to increase or decrease the macroblock size easily. We divided the coordinates into two parts, as shown in Figure 6. If the watermark bit was one, we used coordinate sectors 1 and 2, otherwise, we used coordinate sectors 3 and 4.

Figure 6. The coordinate sectors. Sectors 1 and 2 for bit $"1"$ and sectors 2 and 3 for bit $"0"$.

The most suitable motion vectors were selected for watermark embedding. This selection was based on phase angle and vector magnitude.

The watermark was embedded in the magnitude of the selected motion vectors. The higher magnitude reflects a large movement in the video and vice versa. Phase angles were calculated as given in Equation 4.

$$\begin{aligned} &\text{if wm} = 0; 0 < |\theta_i - \theta_{i+1}| \leq 180, \text{Embed in sectors } 1, 2 \\ &\text{if wm} = 1; 180 < |\theta_i - \theta_{i+1}| \leq 360, \text{Embed in sectors } 3, 4 \end{aligned} \quad (4)$$

where θ is phase angle of motion vector.

If Equation 5 is satisfied, then embed, otherwise go for the next suitable motion vectors.

Calculate the magnitudes of the motion vectors.

$$|\text{Mag}_V(i)| = \sqrt{(h_i^2 + v_i^2)}, 0 < i \leq MB, \quad (5)$$

where MB is macroblock, h and v are horizontal and vertical values of vector, respectively.

Set a threshold τ

Calculate phase angle for embedding the watermark bits

$$\theta(j) = \tan^{-1}\left(\frac{v_j}{h_j}\right), \text{where } j \in E(i) \text{and } E(i) \neq 0 \quad (6)$$

$$\text{if } \theta(j) < 90$$

$$\text{if } (H(j) \times q + \tau) \bmod 2 \neq \text{watermark}(k) \quad (7)$$

$$H_w(j) = H_w(j) + d \text{ otherwise } H_w(j) = H_w(j)$$

$$\text{if } \theta(j) \geq 90 \;\&\&\; \theta(j) < 180$$

$$V_w(j) = V_w(j) + d \text{ otherwise } V_w(j) = V_w(j) \quad (8)$$

$$\text{if } \theta(j) = 45$$

$$H_w(j) = H_w(j) + d \text{ otherwise } H_w(j) = H_w(j)$$

$$V_w(j) = V_w(j) + d \text{ otherwise } V_w(j) = V_w(j) \quad (9)$$

3.4 Watermark extraction and verification

We extracted the watermark as follows:

Obtain motion vectors for the image to be checked and calculate the magnitudes of obtained motion vectors by using Equation 10.

Decode τ

$$E_w(i) = P_w(i) \times F_w(i) \quad (10)$$

$$\text{where } P(i) = \begin{cases} 1 & |\text{Mag}_V(i)| > \tau \\ 0 & |\text{Mag}_V(i)| \leq \tau \end{cases}$$

Calculate phase angle

$$\theta_w = \tan^{-1}\left(\frac{v_{wj}}{h_{wj}}\right) \quad (11)$$

$$\text{If } \theta(j) < 90$$

$$\text{watermark}(k) = (H(j) \times q + \tau) \bmod 2 \quad (12)$$

$$\text{If } \theta(j) > 90 \;\&\&\text{If } \theta(j) < 180$$

$$\text{watermark}(k) = (V(j) \times q + \tau) \bmod 2 \quad (13)$$

$$\text{If } \theta(j) = 45$$

$$\text{watermark}(k) = (H(j) \times q + \tau) \bmod 2$$

$$\textbf{OR } \text{watermark}(k) = (V(j) \times q + \tau) \bmod 2 \quad (14)$$

If the frames or video to be checked are not tampered with then the extracted watermark will match the original watermark, otherwise the extracted watermark will be different.

Algorithm 1. Generating and embedding the watermark

Generation of watermark (w)

Repeat for each frame of the sequence

1. $[LL1, HL1, LH1, HH1] = \text{iwt}(\text{frame})$ // IWT is applied on the frame.
2. $f_d \leftarrow \text{idct2}(LL1)$ // Full-Frame IDCT is applied to $LL1$ of the frame
3. $f_h \leftarrow \text{huffmanco}(f_d)$ // Frame is compressed by using Huffman coding
4. $f_{rand} \leftarrow f_h \text{ XOR } P_{rand}$ // Taking XOR of f_h with key based random sequence P_{rand}

5. $f_{bch} \leftarrow bchenco(f_{rand})$ // BCH encoding of f_{rand} with different pairs (31, 16, 3), (31, 11, 5), (255, 187, 9), (255, 179, 10)
6. $f_{final} \leftarrow Permute$ // Permutation function is then applied to the BCH encoded values for distributing the malicious attack throughout the watermark on verification side

Watermark Embedding

1. Watermark Embedding // The detailed embedding procedure is given in Section 3.3

4. Recovery

The proposed algorithm was able to recover the frame(s) that had been affected by some manipulations, even malicious manipulation such as cut/copy/paste, geometric transformation i.e. rotation, shifting, and scaling. The embedded watermark, which was a highly compressed version of the original topo-map, was decompressed to obtain the original frame back. Note that only the approximation ($LL1$) of the topo-map could be obtained as we compressed $LL1$ of the topo-maps during the watermark generation. The decompression is the inverse procedure of compressing the topo-map and is detailed in the following steps.

- The locations where the watermark bits are embedded are identified, and permuted inversely by using the same key. The key(s) should be available on the verification side.
- The inverse BCH decoder is applied by using the same pair of actual and physical bits.
- The resultant bits are then decoded by the Huffman decoder.
- The resultant coefficients are then passed through the inverse IDCT to obtain the approximation ($LL1$) of the topo-map.

As we used the Huffman coding to highly compress each topo-map, which is highly fragile to JPEG compression, if one Huffman coded bit is affected by JPEG compression or other attacks, the Huffman decoder will be unable to recover the original coefficients. This is because Huffman coding is lossless compression and its decoder requires the correct bits for successful decompression. To avoid this issue, the BCH decoder first corrects the erroneous bits before passing through the Huffman decoder. The correction ratio depends upon the BCH pair used on the BCH encoding side.

5. Results and discussions

5.1 Data collection:

This was an open-eye activity in which the subject pressed the enter button on the keyboard and looked at the blue box on the screen while recording the data. The recorded signals were then converted to topographic format, i.e. EEG topo-maps. Thus, for the experimental work, we obtained the five-second video composed of 29 topo-maps/frames. Some of the motion vectors are shown in Figure 7, where the reference frames are at a distance of three from the current frame, i.e. there are two frames between the current and the reference frame. Although we could have used the immediate frames, the purpose of keeping a distance of three between the current and reference frames was to make the size of the motion vectors large enough. The frame size and the block size within the topo-map are 544×538 and 16×16 respectively, and the search parameter is 7. The watermarked vectors may show a perceptual difference from the original. This is because the block size is 16×16. The vector size varies according to the block size. The search parameter indicates the size of the search space and allows the block to move in all directions within the search space with respect to the block of the previous frame.

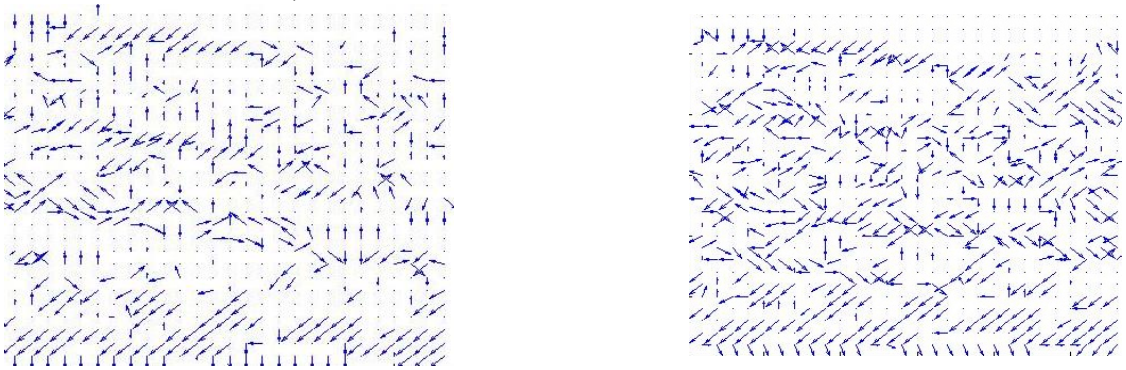


Figure 7. Motion vectors for frame number 1, 2

5.2 Implementation and degradation measurement:

The algorithm has been implemented in the MATLAB – 2011b environment. The watermark was embedded in each frame by using the motion vectors. Figure 8 shows some of the watermarked motion vectors. The watermark size is much bigger because of BCH encoding for error correction, but, as discussed earlier, the embedding capacity is large enough in comparison to normal video. Therefore, it is easy to utilize each frame for embedding.

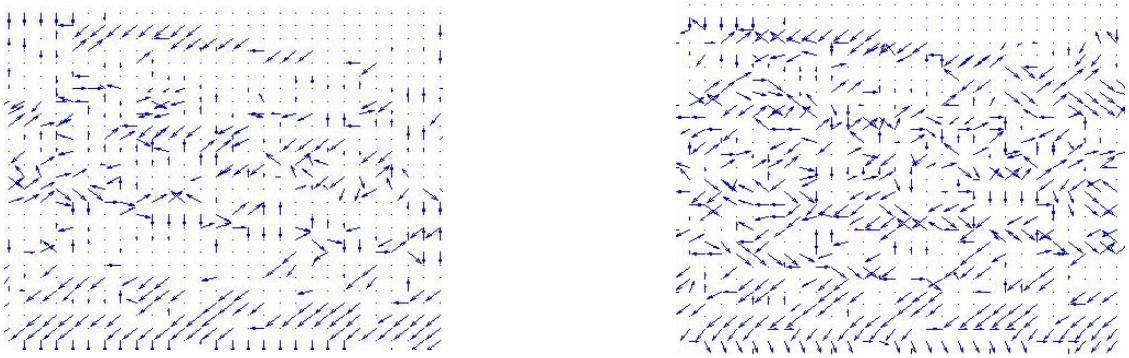


Figure 7. Watermarked motion vectors for the frames given in Figure 6.

The results have been compared with the previous approaches based on PSNR and SSIM. Some of the PSNR and SSIM for the frames and the corresponding watermarked frames are given in Table 2. PSNRs show that the imperceptibility of watermarked frames is much better. In the last column, the values of SSIM are near to one. This shows that the original frames and their corresponding watermarked frames are structurally similar to each other.

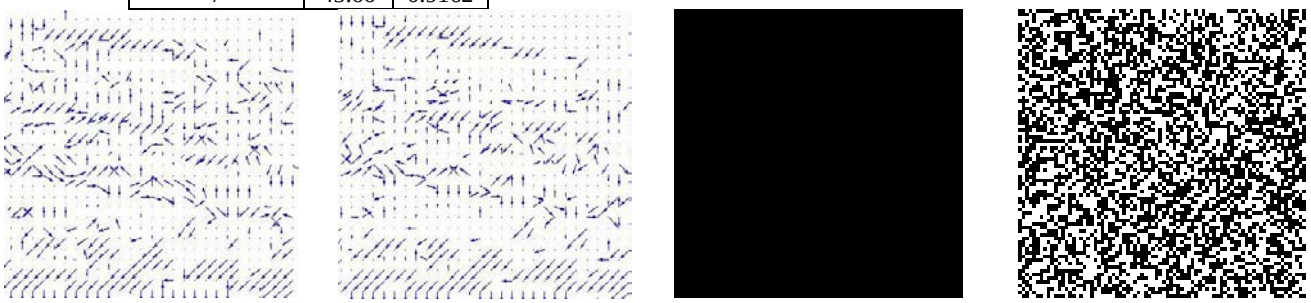
Table 2. PSNR and SSIM for seven frames

Frame Number	PSNR	SSIM
1	45.43	0.9424
2	45.12	0.9534
3	45.76	0.9123
4	45.33	0.8907
5	45.85	0.9632
6	45.21	0.9421
7	45.66	0.9162

The PSNRs show that the degradation is not high even after embedding the high strength watermark, and the SSIM column shows the structural similarity, which is quite acceptable in medical applications.

5.3 Authentication analysis of host data:

This algorithm shows the capability of authenticity, which is not provided by the previous approaches. The first column in Figure 8 shows some of the frames taken from the original data. In the second column, the watermarked frames are displayed. The third and the last columns indicate the difference between the original and extracted watermarks, one without tampering and the other with tampering, respectively. The non-zero regions indicate the non-authenticity of frames.



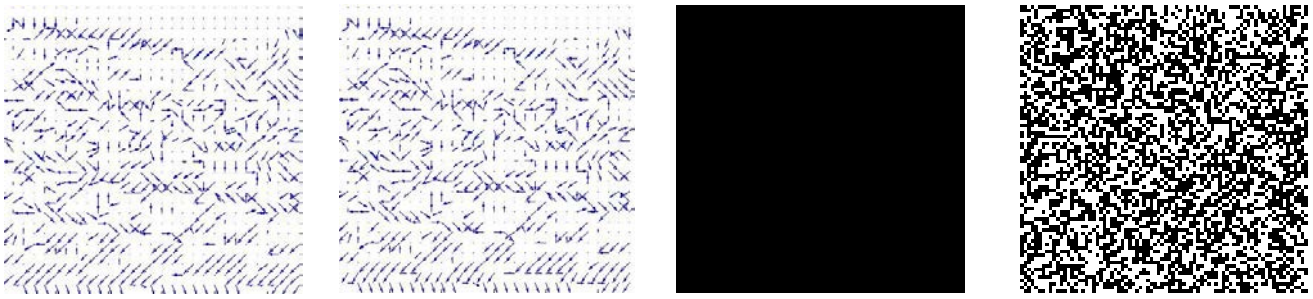


Figure 8. In the first and second columns, original and watermarked images are displayed, respectively. In the third and last column, the extracted watermarks are displayed with the correct and wrong key, respectively.

5.4 Temporal analysis:

By temporal degradation, the extracted watermark will be completely different from the original watermark and show the non-authenticity of the host data. In Figure 8, it is apparent that the last column shows the difference between the original and extracted watermarks. Unlike traditional video, the EEG frames are complex and the temporal changes make a large degradation. In a practical, temporal attack, each area or region that is capable of change may be exposed to this possibility.

5.5 Security analysis:

This algorithm ensures the secure transmission of host data by recovering the tampered regions. The BCH encoder and decoder have the ability to recover erroneous pixels from the tampered watermarked frames. The detail for BCH encoding has been discussed in Section 3.1. The secret key plays an important role in the security of digital content. Hence, in the proposed approach, with knowledge of the key, the watermarks can be extracted; otherwise, the extracted watermark will not be comprehensible. The extracted bits will be similar to the last column of Figure 8.

5.6 Performance comparison with existing approaches:

Generally, the watermarking approaches are comparable in terms of security, imperceptibility, robustness, capacity, etc. The proposed approach is to embed watermarks in the motion vectors of EEG topo-maps. Several algorithms exist that embed watermark(s) in the motion vectors of normal videos. The capacity, behavior, and structure of EEG topo-maps/frames are different from normal video frames. Normal videos are simple, have moving behavior, while the EEG topo-maps are complex, and have expanding/shrinking behavior. Normal videos are not capable of hiding the high strength watermark imperceptibly. On the other hand, in EEG topo-maps, we were able to hide the BCH encoded watermark, which had almost double the strength of the original watermark. Thus, it is difficult to compare the EEG watermarking approaches with normal video watermarking, although, in the literature, we found some watermarking methods for EEG. These used spatial information for the embedding process, rather than motion vector based watermarking. Table 3 shows some of the performance comparisons of the proposed approach with the existing approaches.

Table 3. Performance comparison of the proposed approach with [34] and [35]

Features	**Ref. [34]**	**Ref. [35]**	**Our Approach**
Watermark Payload	Low	Low	Low
Watermark Security	Fragile	No	Highly Secure
Tamper Detection	Yes	No	Good
Imperceptibility (PSNR)	Imperceptible	Better	Good
Imperceptibility (SSIM)	Not calculated	Not calculated	Yes
Survival against Compression	No	No	Yes (User Control)

Attack Classification: Legitimate/Illegitimate	No	No	Yes
Image/Data Recovery	No	No	Yes
Embedding in Spatial/Temporal Information	Spatial	Spatial	Temporal

6. Conclusions and future directions

In this paper, we have authenticated the EEG data using blind semi-fragile watermarks. The EEG data is in the topographic format and the topographic maps collectively represent the video sequence. The inter-frame mv are generated by applying the motion estimation techniques, i.e. BMA. These mv are used for watermark(s) embedding instead of conventional approaches, where the frames (images) are used for watermark embedding. The embedding distortion in the conventional approaches is high, if the watermark is semi-fragile or robust. The watermarking technique is semi-fragile and has the capability to survive legitimate attacks, such as JPEG compression. Some slight illegitimate modification can also be survived by correcting the erroneous bits by using the BCH decoder. Instead of embedding the watermark bits directly in the maps, we embedded the inter-frame difference and this made the embedding distortion very low. The embedding space varied according to the block size and thus we had to make a trade-off while dividing the frame into blocks. In this paper, the trade-off was made between the three conflicting properties of the watermarking system, i.e. imperceptibility, robustness, and capacity.

Future directions: In future, we will use reversible watermarking to retrieve the original data after extracting the watermark and verifying the EEG data. In reversible watermarking, we will be able to resolve the embedding distortion. The embedding distortion is an attack by itself and it is very important to overcome this problem, especially in sensitive applications such as medical and military applications.

Acknowledgment:

This work was supported by the Deanship of Scientific Research, Majmaah University, Saudi Arabia

References

[1] Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2008) Digital Watermarking and Steganography, 2nd ed: Morgan Kaufmann, USA.

[2] Hsieh M. S, Tseng D. C, Huang Y. H (2001) Hiding digital watermarks using multiresolution wavelet transform, Industrial Electronics. IEEE Transactions 48(5): 875-88.

[3] Young-Ho S, Dong-Wook K (20014) A digital watermarking algorithm using correlation of the tree structure of DWT coefficients, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 87(6): 1347-1354.

[4] Hyung-Shin K, Baek Y, Heung-Kyu L,Young-Ho S (2003) Robust image watermark using Radon transform and bispectrum invariants, 5th International Workshop on Information Hiding, Springer, pp. 145-159.

[5] Kundur D, Hatzinakos D (1999) Digital watermarking for telltale tamper proofing and authentication, Proceedings of the IEEE, 87(7): 1167-1180.

[6] Yu D, Sattar F, Barkat B (2006) Multiresolution fragile watermarking using complex chirp signals for content authentication, Pattern Recognition 39(5): 935-952.

[7] Lin p. L, Huang P. W, Peng A. W (20014) A fragile watermarking scheme for image authentication with localization and recovery, Symposium on Multimedia Software Engineering, International, 146-153.

[8] Fridrich J, Goljan M (1999) Images with self-correcting capabilities, 3: 792-796.

[9] Chen C. C, Fan K. C, Wang S. W (2003) A wavelet-based public key image authentication watermarking, International Carnahan Conference on Security Technology, Proceedings. IEEE, 321-324.

[10] Ho C. K, Li C-T 2004) Semi-fragile watermarking scheme for authentication of JPEG images, Proceedings of the International Conference on Information Technology: Coding and Computing 01: 7-11.

[11] Yang S. Y, Lu Z. D. Zou F. H (2004) A novel semi-fragile watermarking technique for image authentication, Proceedings. 7th IEEE International Conference on Signal Processing.

[12] Chamlawi R, Khan A, Idris A (2007) Wavelet based image authentication and recovery, Journal of Computer Science and Technology 22(6): 795-804.

[13] Chamlawi R, Khan A (2010) Digital image authentication and recovery: Employing integer transform based information embedding and extraction, Information Sciences 180(24): 4909-4928.

[14] Campisi P, Kundur D,Hatzinakos D, Neri A (2002) Compressive data hiding: An unconventional approach for improved color image coding, EURASIP Journal on Applied Signal Processing, 2002(2) :, 152-163.

[15] Chamlawi R, Khan A, Usman I (2010) Authentication and recovery of images using multiple watermarks, Computers & Electrical Engineering, 36(3): 578-584.

[16] Lin C. y, Chang S. F (2000) Semi-fragile watermarking for authenticating JPEG visual content, Proc. SPIE 3971, Security and Watermarking of Multimedia Contents II, 3971: 140-151.

[17] Kim K. S, Lee M. J, Lee H. Y, Lee H. K (2009) Reversible data hiding exploiting spatial correlation between sub-sampled images, Pattern Recognition 42(11): 3083-3096.

[18] Khan A, Malik S. A, Ali A, Chamlawi R, et. al. (2012) Intelligent Reversible Watermarking and Authentication: Hiding Depth Map Information for 3D Cameras, Information Sciences 216: 155-175.

[19] Malik S. A, Khan A, Hussain M. et al. (2012) Authentication of Images for 3D Cameras: Reversibly Embedding Information Using Intelligent Approaches, Journal of Systems and Software 85(11): 11, 2665-2673.
[20] Barjatya A (2004) Block matching algorithms for motion estimation, IEEE Transactions Evolution Computation 8(3): 225-239.
[21] Horé A, Ziou (2010) Image quality metrics: PSNR vs. SSIM, Proceedings of the 2010 20th International Conference on Pattern Recognition, 2366-2369.
[22] Teo P. C, Heeger D. J (19940 Perceptual image distortion, First International Conference on Image Processing 2: 982-986.
[23] Farrell J. E (1999) Image quality evaluation, in: Colour imaging: vision and technology, WILEY, Derby, UK, 315-337.
[24] Eskicioglu A. M, Fisher P. S (1995) Image quality measures and their performance, IEEE Transactions on Communications 43(12): 2959-2965.
[25] Hartung F, Girod B (1997) Digital watermarking of MPEG-2 coded video in the bitstream domain, Conference on Acoustics, Speech, and Signal Processing 4: 2621-2624.
[26] Song J, Liu K (1999) A data embedding scheme for H. 263 compatible video coding, Proceedings of the 1999 IEEE International Symposium on Circuits and Systems 4: 390-393.
[27] Fang D. Y, Chang L. W (2006) Data hiding for digital video with phase of motion vector, Proceedings of IEEE International Symposium on Circuits and System 1422-1426.
[28] Wade G (1994) Signal coding and processing, 2nd edition, Cambridge University Press, UK.
[29] Turaga D, Alkanhal M (1998) Search algorithms for block-matching in motion estimation, Mid-term Project.
[30] Zhang J,Li J, Zhang L (2001) Video watermark technique in motion vector, Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing, 179-182.
[31] Xu C, Ping X, Zhang T (2006) Steganography in compressed video stream, Proceedings of the First International Conference on Innovative Computing, Information and Control 1:269-272.
[32] He X, Luo Z (2008) A novel steganographic algorithm based on the motion vector phase, IEEE International Conference on Computer Science and Software Engineering, 822-825.
[33] Aly H (2011) Data hiding in motion vectors of compressed video based on their associated prediction error, IEEE Transactions on Information Forensics and Security 6(1): 14-18.
[34] Mukherjee A, Dey G, Dey M, Dey N (2014) Web-Based intelligent EEG signal authentication and tamper detection system for secure telemonitoring, in: Brain-Computer Interfaces, Intelligent Systems Reference Library, SPRINGER, 295-312.
[35] Pham T. D, Tran D, Ma W (2015) A Proposed Blind DWT-SVD Watermarking Scheme for EEG Data, Neural Information Processing, Lecture Notes in Computer Science, SPRINGER 9492: 69-76.

Rafi Ullah was born in KP Pakistan in 1977. He received the MS degree in computer system engineering in 2006 from GIK institute Pakistan and Ph.D. degree in computer and information sciences in 2010 from PIEAS Pakistan. In addition, he worked as postdoctoral fellow in 2012 at UTP Malaysia. From 2012 to 2015, he was an assistant professor of computer science in COMSATS Islamabad Pakistan. Currently, he is working as assistant professor in college of science, Majmaah University, Saudi Arabia. His research areas are image processing, digital watermarking, and medical/neuro imaging. He publishes several research articles in the area of image watermarking.

Hani Ali Alquhayz received the B.S degree in computer science in 2006. He received master degree in information system management in 2009 form De Montfort University in the UK. He received PhD degree in computer science from De Montfort University in the UK in 2015. From 2006 to 2008, he worked as system engineer in Pan Nesma Co. Ltd. Currently; he is working as assistant professor in college of science, Majmaah University, Saudi Arabia. His research area is security management systems for networks and image processing.