A Video Watermarking Scheme Resistant To Synchronization Attacks Based on Statistics and Shot Segmentation

Li-he ZHANG [1], Gang XU [1], Ji-jun ZHOU [2], Lin-jie SHEN [1]
[1] *Department of Electronic Engineering, Dalian University of Technology, China*
zhanglihe@yeah.net, dlut_xg@yahoo.com.cn
[2] *School of Electronics Engineering and Computer Science, Peking University, China*
zhoujj@pku.edu.cn

Abstract

One of the challenges of blind watermark detection is synchronization. In this paper, a new video watermarking procedure for resistant synchronization is proposed. We partition watermark into several segments, and embed every segment into different scenes of video sequence. Firstly, motion vectors are grouped according to their magnitude and each group is further partitioned into two subsets. We use element number ratio in two subsets of the same group to denote watermark bit 0 or bit 1. Thus, watermark is associated with motion vectors' statistical characteristics, and those motion vectors carrying the identical watermark bits are uniformly distributed to the whole video sequence, they are spatio–temporal indiscerptibly. It is shown that this kind of watermark is more resilient against temporal synchronization attacks. Experimental results from an implementation of the algorithm are presented.

1. Introduction

With the rapid development of stream media techniques, pirate and illegal replication of video contents have become more and more serious. A new technique of digital rights management called video watermarking has gradually turned into hot research topic. Some international organizations such as MPEG and SDMI have also appended watermarking technique to their standards to protect and manage digital contents' rights [1, 2].

Motion vectors are related to time continuity and smoothness of video frames. During the process of encoding and decoding, most of motion vectors take on small changes and we have conducted plenty of experiments to verify the conclusion. Motion vectors reflect motion shift between current coded macroblock of current frame and its optimal matching macroblock in referenced frame. Motion vectors themselves are independent of macroblock content, that's why the watermark that is embedded into motion vectors is immune from content attacks such as adding noise and filtering etc.

A motion-vector-based watermark algorithm has been proposed in 1997 by Kutter et al. in an MPEG-4 proposal [3]. In this work, the authors selected only one macroblock from every frame of picture as embedded location and watermark is embedded into one component of motion vector. Based on that, scholars have proposed some improved algorithms. Zhang has proposed that watermark should be only embedded in large magnitude and small phase difference component[4]. Liu first has defined a texture parameter of luminance component and then self-adaptive embedded watermark information in motion vectors according to the texture of frames[5]. Bodo has estimated the motion vectors computed by an exhaustive BMA on blocks of size 4×4 [6], it has used the average magnitude of motion vectors of neighboring four blocks as candidate embedding locations and then has inserted watermark according to human visual model. These methods above can enhance robustness and invisibility to some extent, but none of them is robust against synchronization attacks such as frame deleting and frame inserting.

In this paper, a new embedding technique has been introduced. First of all, video sequence should be partitioned into a series of shots. Motion vectors in each shot are grouped into some sets according to their magnitudes and every set includes two Regular Subsets. Then we embed watermark by modifying motion vectors' magnitude to make the ratio of motion vector number in two Regular Subsets of the same set satisfy specific conditions. In order to compensate image distortion due to motion vectors' change, the macroblock matching error is predetermined.

2. Video Shot Segmentation

P frame is a forward-prediction-coding picture. There are three kinds of macroblocks in a P frame, they are skipped macroblock, forward prediction macroblock and intra-coding macroblock. When there is very small change between current coded macroblock and its referenced macroblock, the current coded macroblock would be skipped, and would not be coded. When it is difficult to find appropriate matching macroblock in the referenced picture, the current coded macroblock would be coded in intra-coding style. Therefore, if there are many intra-coding macroblocks in a P frame, shot-cut very likely happens in that frame.

Suppose that $NI(i)$ and $NF(i)$ respectively are the number of intra-coding macroblocks and forward-prediction macroblocks in the i th P frame, where $N(i)$ is the total number of macro-blocks. Then $R(i)$ can be defined to detect shot cut as below.

$$R(i) = \frac{|NI(i) - NF(i)|}{N(i)}$$

0-7695-2976-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISDA.2007.151

If $R(i)$ approaches to 0, it shows that the difference between current picture and referenced picture is very small. If $R(i)$ approaches to 1, it shows there are many intra-coding macroblocks in the i th P frame and shot-cut much possibly would happen in this P frame. In this algorithm, $R(i)$ is compared with a fixed threshold to decide where shot-cut would take place. Fig.1 illustrates the experiment result of shot segmentation. Through the comparison of detection result with actual video shot-cut, the detection accuracy is higher than 90%.

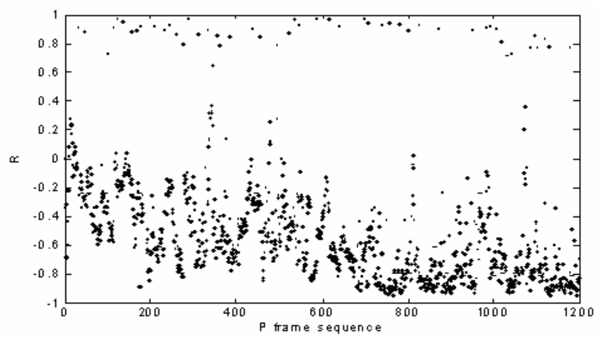

Fig.1. Video shot segmentation

3. Statistical Classification

Suppose that the motion vector of macroblock (*i,j*) at the time t is $V_{i,j,t} = \{V^h_{i,j,t}, V^v_{i,j,t}\}$, where $V^h_{i,j,t}$ and $V^v_{i,j,t}$ respectively are its horizontal and vertical components, and both of them always are integers because they are decoded from video code stream. The classification method is shown in Fig.2.

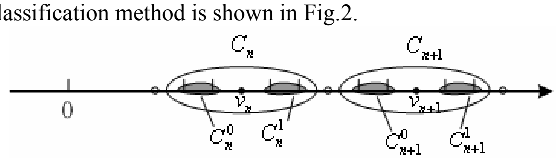

Fig.2. Motion vector's classification

The figure axis denotes the magnitude of motion vector component. Every five consecutive integral magnitudes are grouped into a set, they are rounded by a large ellipse as shown in Fig.2. Let $C_n = \{V_{i,j,t} \mid v_n - 2 \le V_{i,j,t} \le v_n + 2\}$ denote the *n*th set, where the suffix (*i, j*) is the current macroblock position in P frame, and the suffix *t* is the position of current P frame in the whole video sequence. There are two filled-in gray region including two consecutive integral magnitudes in every ellipse, they respectively denotes Regular Subsets $C^0_n = \{V_{i,j,t} \mid v_n - 1 \le V_{i,j,t} \le v_n - 2\}$ and $C^1_n = \{V_{i,j,t} \mid v_n + 1 \le V_{i,j,t} \le v_n + 2\}$. And there is an isolated element shown as solid dot between two subsets, it belongs to none of subsets. There is also an isolated element shown as hollow dot between two adjacent sets, similarly it belongs to none of sets. In the proposed method, we use the element number ratio of C^0_k and C^1_k to denote watermark bit, therefore isolated elements called Special Subset can decrease the possibility of element shift from one Regular Subset to another Regular Subset or from one set to another set when watermark is under attacks. The above is the basic classification method. If the elements of one set are too few, it will influence watermark algorithm's robustness and security. Therefore we should calculate motion vector histogram of P frame in advance, and then classify those motion vectors according to the statistical histogram guaranteeing that there are considerable elements in every set.

4. Algorithm Theory

Fig.3 illustrates the watermark embedding algorithm's block diagram. Firstly, video picture sequences are segmented into a series of shots by comparison between $R(i)$ of every P frame and a predefined threshold. Secondly, motion vectors of all of P frames are decoded from video code stream, and those in the same shot are classified according to their histograms. Thirdly, watermark information is embedded into classified motion vectors under the control of pseudorandom sequence. Motion vectors in different shots are processed separately. Finally, the modified motion vectors and other information are recoded to form the watermarked video.

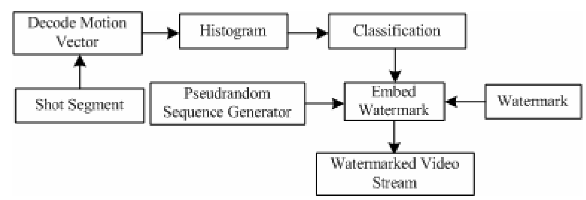

Fig.3. Watermark embedding block diagram

4.1. Watermark Embedding

In order to maintain the quality of video, the motion vector sets with small magnitude are not used to embed watermark. Suppose C_k denotes the *k*th candidate set, it consists of two Regular Subsets C^0_k and C^1_k. n^0_k and n^1_k respectively are the element numbers of C^0_k and C^1_k. Thus the proportion of n^0_k / n^1_k is used to denote watermark information (markbit). Table.1 illustrates the mapping relation.

Table 1. Watermark information coding mapping relation

markbit	01	10	11	00
Meaning	$n^0_k \approx n^1_k$	$n^0_k / n^1_k \approx 2$ or $n^0_k / n^1_k \approx 1/2$	$n^0_k / n^1_k \approx 3$ or $n^0_k / n^1_k \approx 1/3$	$n^0_k << n^1_k$ or $n^0_k >> n^1_k$

Watermark is embedded through adjusting element magnitude of a set to make n^0_k / n^1_k satisfy above relations. The changed elements' locations are controlled by a pseudorandom sequence in order to keep all of elements having equal chances to be changed. It is helpful to extract watermark and to enhance watermark's robustness against concentrated frame deleting attack.

The methods of adjusting that the elements are shifted between Regular Subset and Special Subset through adding 1 operation or subtracting 1 operation are illustrated in Fig.4. They are marked by symbols of ①, ②, ③ and ④. The functions of ① and ③ are respectively to decrease n^0_k and n^1_k, the functions of ② and ④ are respectively to increase n_k^0 and n_k^1. The concrete watermark embedding method is as follows.

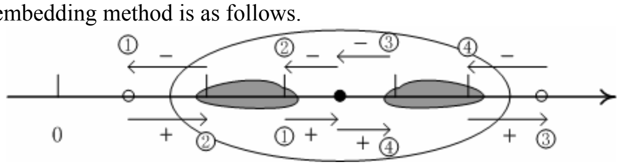

Fig.4. Methods of adjusting n_k^0 and n_k^1

if markbit==00
  if $n_k^0 < n_k^1$
  most elements' magnitudes of C_k^0 increase with 1, make $n_k^0 << n_k^1$
  else if $n_k^0 > n_k^1$
  most elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 >> n_k^1$
  else
  most elements' magnitudes of C_k^0 increase with 1, make $n_k^0 << n_k^1$. Or most elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 >> n_k^1$
else if markbit==01
  if $n_k^0 < n_k^1$
  partial elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 == n_k^1$
  else if $n_k^0 > n_k^1$
  partial elements' magnitudes of C_k^0 increase with 1, make $n_k^0 == n_k^1$
  else
  Don't adjust
else if markbit==10
  if $n_k^0 / n_k^1 < 1/2$
    partial elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 / n_k^1 == 1/2$
  else if $5/4 > n_k^0 / n_k^1 > 1/2$
    partial elements' magnitudes of C_k^0 increase with 1, make $n_k^0 / n_k^1 == 1/2$
  else if $2 > n_k^0 / n_k^1 > 5/4$
    partial elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 / n_k^1 == 2$
  else
    partial elements' magnitudes of C_k^0 increase with 1, make $n_k^0 / n_k^1 == 2$
else
  if $n_k^0 / n_k^1 < 1/3$
    partial elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 / n_k^1 == 1/3$
  else if $5/3 > n_k^0 / n_k^1 > 1/3$
    partial elements' magnitudes of C_k^0 increase with 1, make $n_k^0 / n_k^1 == 1/3$
  else if $3 > n_k^0 / n_k^1 > 5/3$
    partial elements' magnitudes of C_k^1 decrease with 1, make $n_k^0 / n_k^1 == 3$
  else
    partial elements' magnitudes of C_k^0 increase with 1, make $n_k^0 / n_k^1 == 3$

In this algorithm, we use the element numbers of two Regular Subsets in the same set to denote watermark information. Thus if attackers want to destroy watermark by frame deleting, it must change the proportion of n_k^0 / n_k^1. This would be very difficult when the value of n_k^0 / n_k^1 is bigger. Because frame-deleting synchronously changes n_k^0 and n_k^1, the proportion would be basically invariable. On the other hand, motion vectors are distributed to the whole video sequence, therefore deleting several frames will not change n_k^0 / n_k^1 greatly. Apparently this algorithm can also resist video cutting attack, namely watermark information can be extracted intactly from partial video data.

4.2. Watermark Extracting

Watermark extracting is the contrary process of watermark embedding. First of all, video shot is segmented using the same method as that is used in watermark embedding, then motion vectors are decoded and their histograms are computed. Secondly, motion components of P frame in every video shot are classified, and compare n_k^0 with n_k^1 to get pseudrandom modulated watermark bit sequence. Finally demodulate the bit sequence into watermark.

if $n_k^0 \approx\approx n_k^1$
  markbit=01
else if $n_k^0 / n_k^1 \approx\approx 2$ || $n_k^0 / n_k^1 \approx\approx 1/2$
  markbit=10
else if $n_k^0 / n_k^1 \approx\approx 3$ || $n_k^0 / n_k^1 \approx\approx 1/3$
  markbit=11
else
  markbit=00

4.3. Error Compensation

Motion vector denotes the optimal matching macroblock' motion relative to the referenced macroblock. According to MPEG-2 video coding standard, the code mode of motion vectors is difference coding. Only the difference between interframe coding macroblocks' motion vectors and those of referenced macroblocks is saved during the encoding process. Error compensation must be performed to degrade error accumulation resulting from watermark embedding.

Suppose that MVe1 is current coded motion vector, MVc is its referenced motion vector, then EMV1 = MVe1 - MVc. If embedding watermark makes MVc increase 1, MVe1 will also increase 1 if coding difference EMV1 is invariable. If MVe1 also increases 1 due to watermark embedding, the motion vector which use MVe1 as referenced vector will possibly increases 2. Analogically it will possibly bring back vectors larger offset. In order to avoid error cumulating, we must subtract the change brought by watermark embedding in advance if watermark is directly inserted into the difference of current motion vectors, otherwise watermark should be inserted into decoded motion vectors.

5. Experiments and Performance Analysis

The sequences *Table Tennis*(*TT*), *Mobile Calendar*(*MC*) and *Flower Garden*(*FG*) are experimented. Table 2 shows the thresholds of extracting watermark in these experiments.

Table 2. Thresholds for extracting watermark

Threshold	$0.6 < n_k^0 / n_k^1 \leq 1.5$	$1.5 < n_k^0 / n_k^1 \leq 2.5$ or $0.4 < n_k^0 / n_k^1 \leq 0.6$	$2.5 < n_k^0 / n_k^1 \leq 3.5$ or $0.2 < n_k^0 / n_k^1 \leq 0.4$	$n_k^0 / n_k^1 > 3.5$ or $n_k^0 / n_k^1 \leq 0.2$
markbit	01	10	11	00

5.1. Stability Test of P Picture Motion Vectors

Two thousand motion vectors are randomly extracted from P frames of four different bit rate sequences respectively. The mean of magnitude difference is 0.0255, variance is 0.0382. It is obvious that motion vectors' changes are very small during the video recoding process. This is helpful to extract watermarking. Fig.5 and Fig.6 are the experiment results of *Flower Garden*.

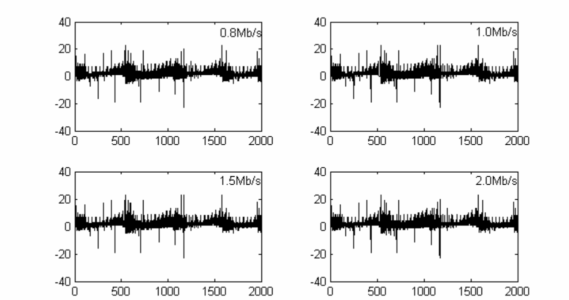

Fig.5. Motion vectors' magnitudes of different code rates (x axis denotes randomly extracted motion vector from P frames, y axis denotes magnitude)

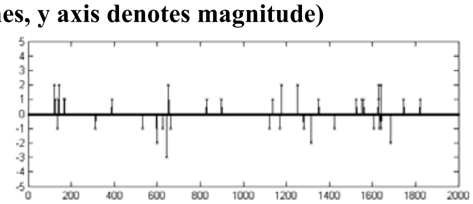

Fig.6. Magnitude difference of P frame motion vectors (x axis denotes randomly extracted motion vector from P frames, y axis denotes magnitude difference of the same location macroblock's motion vector with code ratio 0.8Mb/s and 1.5Mb/s)

5.2. Watermark Invisibility

The peak signal-to-noise ratio (PSNR) is taken to measure the video quality. The PSNR for each frame is separately calculated, and then their mean is used to measure the watermark invisibility. The results of three video sequences respectively with different bit rates are shown in Table 3.

Table 3. Average PSNR of watermarked frames

Bitrate (Mbits/s)		1.5	4.0	6.0	8.0
PSNR	*TT*	45.4	46.3	47.2	48.0
	MC	44.5	45.4	46.9	47.2
	FG	46.4	46.3	48.6	49.5

5.3. Watermark Robustness

In the experiment, attack methods contain frame deleting, frame inserting, video cutting and noise adding. The experiment results of frame deleting and frame inserting attack are shown in Table 4.

Table 4. Loss frame ratio vs. Watermark accuracy

Loss frame ratio (%)		8	12	16	20	22
Watermark accuracy (%)	*TT*	100	98.1	91.0	72.5	61.2
	MC	100	98.0	91.3	71.4	58.6
	FG	100	98.4	89.4	70.0	56.5

Table 5. Frame inserting ratio vs. Watermark accuracy

Frame inserting ratio (%)		8	12	16	20	22
Watermark accuracy (%)	*TT*	100	98.9	94.2	74.8	63.4
	MC	100	99.2	95.6	75.4	61.1
	FG	100	98.5	93.1	72.0	58.1

From the result of above experiment, it can be said that the algorithm is able to effectively resist frame deleting and frame inserting attacks. Watermark accuracy is higher than 80% when the attack intensity is less than 18%. Watermark accuracy is the percentage of bits that are correctly decoded.

Table 6. Frame cutting ratio vs. Watermark accuracy

Frame cutting ratio (%)		75	80	85	90
Watermark accuracy (%)	*TT*	100	90.1	76.7	58.0
	MC	100	91.6	75.4	59.3
	FG	100	89.9	72.8	55.4

Table 7. PSNR vs. Watermark accuracy

PSNR(dB)		44.3	41.6	38.9	35.8
Watermark accuracy (%)	*TT*	100	91.5	78.1	60.2
	MC	100	90.2	75.5	59.7
	FG	100	90.6	74.8	58.0

The experiment result of frame cutting is shown in Table 6. When frame cutting ratio is lower than 80%, watermark accuracy will be about 90%. Namely watermarks can be extracted at accuracy of 90% from 90 frames sequence.

If only PSNR is not lower than 38dB, watermark accuracy is always higher than 75%, it is shown in Table 7.

5.4. Algorithm Performance Analysis

In order to indicate advantages of the proposed algorithm, a comparative experiment with reference [7] is made. The experiments use the same video sequences and the same watermark sequences. Fig.7 and Fig.8 show that when attack intensity is less than 15%, the two algorithms' performances are very close. But if we continue to enhance attack intensity, our proposed algorithm exhibits better performances than reference [7].

Noise attack has more great influence on DCT coefficients of video images than that on motion vectors. Because noise simultaneously influences current frame and reference frame, it will generate very small additional offset between current coded macroblock and optimal matching macroblock in reference frame. Fig.9 also validates the conclusion that under the comparative intensity noise attack, watermark accuracy of the proposed algorithm is higher than that of reference [7].

Reference [7] algorithm embeds the same watermark information into all the video frames of a scene. Video clip very possibly deleting all the video frames of some scene will destroy the watermark information. Therefore, the algorithm isn't robust against frame cutting attack. The result is shown in Fig.10.

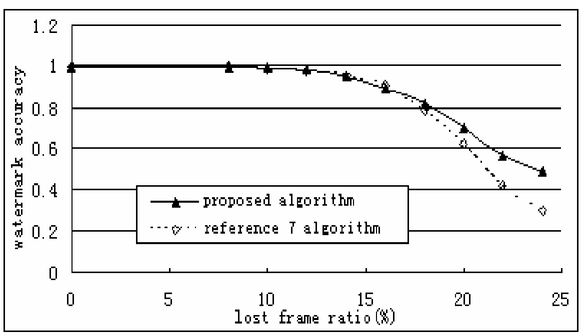

Fig.7 Algorithm performance of frame dropping

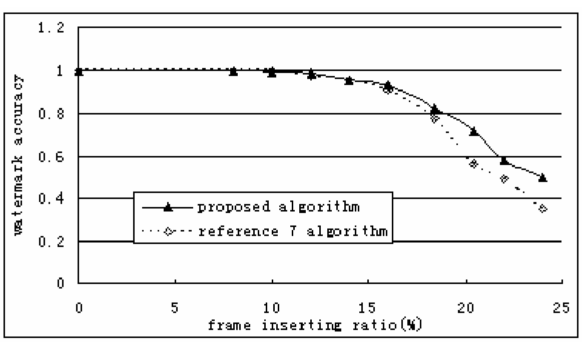

Fig.8 Algorithm performance of frame inserting

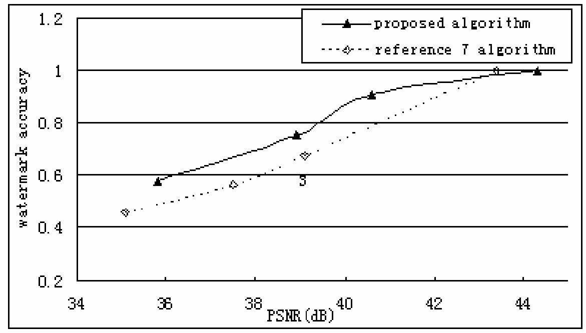

Fig.9 Algorithm performance of noise attack

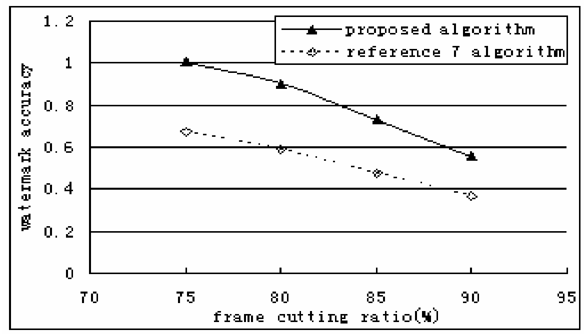

Fig.10 Algorithm performance of frame cutting

6. Conclusion

In this paper, a robust watermark algorithm based on characteristic classification of motion vector is proposed. It uses motion vector's statistical characteristic to denote watermark binary information and motion vectors of a given set are to disperse over the whole video sequence. As there is no strictly physical partition, watermark synchronization problem is not being. If only the characteristics are kept no changing after attacks, watermark can be extracted correctly. Moreover, the algorithm has good payload due to embedding different information into different shots. The experiment results validate its robustness and invisibility.

7. References

[1] L.J. Camp, "First principles of copyright for DRM design," *IEEE Internet Computing,* 2003, Vol. 7, pp. 59-65.

[2] F. Hartung, F. Ramme, "Digital rights management and watermarking of multimedia content for m-commerce applications," *IEEE Communications Magazine*, 2000, Vol. 38, No. 11, pp. 78-84.

[3] M. Kutter, F. Jordan, T. Ebrahimi, "Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video," Technical report M2281, ISO/IEC document, JTC1/SC29/WG11, 1997.

[4] J. Zhang, J.G. Li, L. Zhang, "Video watermark technique in motion vector," *Proc. of XIV Brazilian Symposium on Computer Graphics and Image Processing*, 2001, pp. 179-182.

[5] Z.N. Liu, H.Q. Liang, X.X. Niu, Y.X. Yang, "A robust video watermarking in motion vectors," *Proc. of 7th International Conference on Signal Processing*, 2004, Vol. 3, pp. 2358-2361.
[6] Y. Bodo, N. Laurent, J.L. Dugelay, "Watermarking video, hierarchical embedding in motion vectors," *Proc. of International Conference on Image Processing*, 2003, Vol. 2, pp. 739-742.
[7] S. Biswas, S.R. Das, E.M. Petriu, "An adaptive compressed MPEG-2 video watermarking scheme," *IEEE Transactions on Instrumentation and Measurement*, 2005, Vol. 54, No. 5, pp. 1853-1861.