Robust Block Selection for Watermarking Video Streams

Tamás Polyák, Gábor Fehér

Abstract— In this paper a robust video watermarking method is presented, which embeds data in the wavelet domain using edge detection. The algorithm uses the luminance values around the edges where changes are less noticeable for the human visual system. Watermark is embedded in an additive way using spread spectrum noise. The complexity of the method is low. The algorithms used for wavelet transformation and edge detection are fast and simple. Results show that the watermark signal is imperceptible, and the method is robust against low bitrate lossy compressions, like H.264 and XviD.

***Index Terms*—Edge detection, spread spectrum, video stream, wavelet transform, watermarking**

I. INTRODUCTION

DIGITAL video streaming is a more and more popular service among content providers. These streams can be displayed on various types of devices: computer, PDA etc. The content can be easily distributed through the internet, but digital contents carry a big security risk, i.e. the copying and reproduction of the content is very easy and effortless. Even users without special knowledge can easily share the downloaded content with other people.

To avoid illegitimate access, digital content is often encrypted, and travels in an encrypted form to the consumer. Although encryption secures the content on the way to the consumer, during playback it must be decrypted, and this is the point where it exposed to illegal copying. In these cases watermarks can be used to give some extra protection to the content, since watermarking can embed additional information in the digital content.

Watermarks are embedded in the digital content in an imperceptible way. Watermarks can carry some information about the content: owner of the content, metadata, QoS parameters etc. This protection does not eliminate the need for encryption, but is a supplemental technique to secure the content, or store additional data. These watermarks are called robust watermarks because they must survive transformations of the underlying content e.g. lossy compression.

Digital watermarking algorithms are generally divided into two groups: algorithms that hide data in the spatial domain. It means that information is embedded by modifying the pixel values directly [1]-[3], and algorithms that use a transform domain for data hiding. Discrete cosine transform (DCT) and discrete wavelet transform (DWT) are often used at transform domain watermarking [4]-[7]. These watermarking techniques modify coefficients of the given transform domain. Wavelet transform is commonly used because it has many advantages over DCT transform. It is closer to the human visual system (HVS), instead of processing 8 x 8 pixel blocks it processes the whole frame. It divides the frame into four parts with different frequencies (LL, LH, HL and HH) and directions.

Some watermarking techniques embed data in object borders, and other perceptually important areas. It has several advantages: the HVS is less sensitive to changes made in these components, and modern lossy compression algorithms leave them relatively untouched to maintain high video quality. These properties make these regions ideal for watermarking. The detection of suitable regions can be realized in the uncompressed domain. Pröfrock, Schlauweg and Müller use the Normed Centre of Gravity (NCG) algorithm to find the blocks that contain object borders or other significant changes in the video [8]. Ellinas presents an algorithm that embeds data in images using four level DWT and edge detection [7]. This algorithm also modifies the surroundings of the edges; it is accomplished using a morphological dilatation with a structuring element of 9x9.

The proposed algorithm is designed for watermarking low resolution (CIF and QCIF) video streams. During the design of a video watermarking algorithm complexity has to be taken into count. A trade-off has to be made between complexity, quality loss, and robustness. The algorithm uses the wavelet domain for data hiding. It modifies the wavelet coefficients that belong to object borders. Visible artifacts will not appear on the source video. The suitable coefficients are selected by an edge detection algorithm. Watermark is inserted in an additive way that a spread spectrum pseudorandom noise is added to the luminance pane of the middle frequency components.

II. THE WATERMARK EMBEDDING PROCESS

Fig. 1 shows the process of watermark embedding. First the input video signal ($X_{i,j}$) is transformed using a fast two level DWT transform, the Haar wavelet [9].

Manuscript received March 22, 2008.

T. Polyák is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok krt. 2., 1117 Budapest, Hungary (e-mail: tpolyak@tmit.bme.hu).

G. Fehér is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok krt. 2., 1117 Budapest, Hungary (e-mail: feher@tmit.bme.hu).

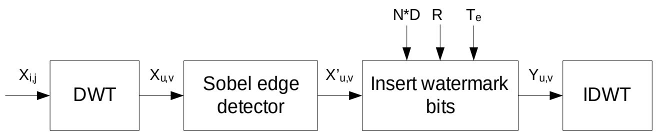


Fig.1. Block diagram of the watermark embedding process

The transformation produces a low frequency, low resolution approximation subband, and 6 detail subbands. Watermark is inserted into the middle frequency components (HL and LH) of each level. Fig. 2. shows the video frame after the wavelet decomposition

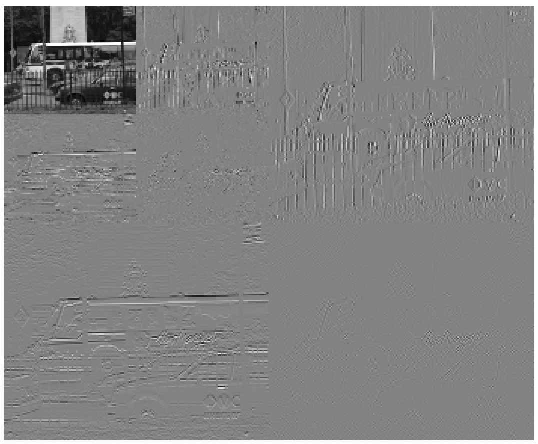


Fig.2. Video frame after wavelet transformation

The transformed video then gets through an edge detecting filter, where edge detection is applied on the middle frequency components. Edge detection is performed using the Sobel edge detector [10].

The Sobel operator performs a 2-D spatial gradient measurement on an image. It is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The Sobel edge detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (G_x) and the other estimating the gradient in the y-direction (G_y). A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The two kernels are shown in (1).

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \tag{1}$$

The magnitude of the gradient (G) is then calculated using (2)

$$|G| = \sqrt{G_x^2 + G_y^2}\,. \tag{2}$$

To improve performance an approximate magnitude is used (3)

$$|G| = |G_x| + |G_y|\,. \tag{3}$$

Fig. 3. shows the frame after edge detection in the middle frequencies.

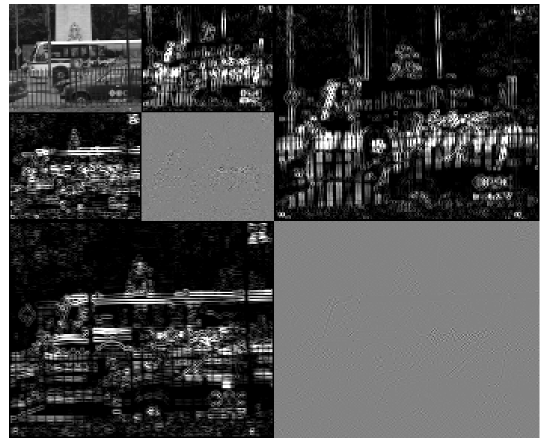


Fig.3. Video frame with Sobel edge detection in the middle frequency DWT coefficients

During watermark insertion the edges with value greater than a given threshold (T_e), and the pixels around them in a given radius (R) are selected. The value of the radius is different at the different levels of the transformed image. Radius of two pixels is used at the first level, and radius of one pixel is used at the second level considering that the higher level coefficients contain lower frequency components, which affect the quality of the video.

Data is embedded into the four selected middle frequency areas by adding a pseudo random spread spectrum noise.

First, data is extended with a chip rate *cr* to improve robustness.

$$d_i = D_x, \quad \text{where } i = x \cdot cr, ..., (x+1) \cdot cr - 1 \tag{4}$$

Insertion is done in an additive way using (5).

$$Y_{u,v} = X'_{u,v} + \alpha \cdot N_i \cdot d_i \tag{5}$$

where $Y_{u,v}$ are the modified wavelet coefficients, $X'_{u,v}$ are the selected wavelet coefficients for watermarking, α is the strength of the watermark (the value which the luminance values are modified by), N_i is the pseudorandom Gaussian noise consisting of -1 and 1, d_i is the data to embed. The data also consists of -1 and 1 according to 0 and 1.

The pseudorandom noise N_i is calculated from a seed, and from the *u* and *v* coordinates of the selected coefficients. This seed is used as the key for the watermarking method.

The same data are embedded into consequent frames to improve robustness.

III. The Watermark Detection Process

The process of the watermark detection is also made in the wavelet domain. After the two level DWT the middle frequency blocks get through an edge detection. Fig. 4. shows the block scheme of the watermark detecting process.

ISBN:978-988-98671-9-5

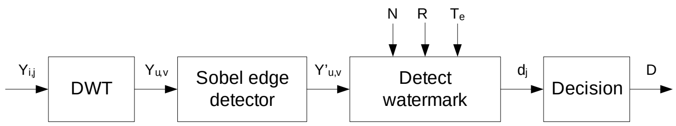


Fig.4. Block diagram of the watermark detection process

The embedded data are extracted by correlating the pseudorandom noise N_i with the area containing the detected edges and their surroundings. The detection is blind. Assuming that the value of $Y_{u,v}$ coefficients is almost equal,

$$d_j = \frac{1}{K} \sum_{u,v} Y_{u,v} \cdot N_i \, , \tag{6}$$

can be used, where K is the number of suitable coefficients and d_j is the extracted value. The embedded bit is calculated the following way:

$$\begin{aligned} \delta = 0, \quad if \ d < -T_w \, , \\ \delta = 1, \quad if \ d > T_w \end{aligned} \tag{7}$$

where T_w is a threshold used to decrease the number of false results.

Its value is:

$$T_w = \frac{\alpha}{2} \cdot \tag{8}$$

Because the embedded information changes only after 9 frames, simple voting is applied on the detected bits.

IV. Video Quality Measurement

A. *PSNR*

PSNR (Peak Signal-To-Noise Ratio) is a widely used method for measuring objective quality produced by image and video manipulating algorithms [11]. PSNR is expressed in terms of the logarithmic decibel scale. The typical values of the PSNR are between 30 and 50 dB, where higher means better quality. It is calculated using (9)

$$\mathrm{PSNR} = 10 \cdot \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right), \tag{9}$$

where *MSE* is the value of the Mean Square Error between the original and the modified frame. It is calculated using (10)

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[x(i,j) - y(i,j) \right]^2 , \tag{10}$$

where M is the height, N is the width of the frame, x and y are the original and the watermarked images. In some cases however PSNR does not correlate well with the HVS, although it is a widely used metric at measuring the quality loss caused by video manipulating and compressing algorithms.

B. *SSIM*

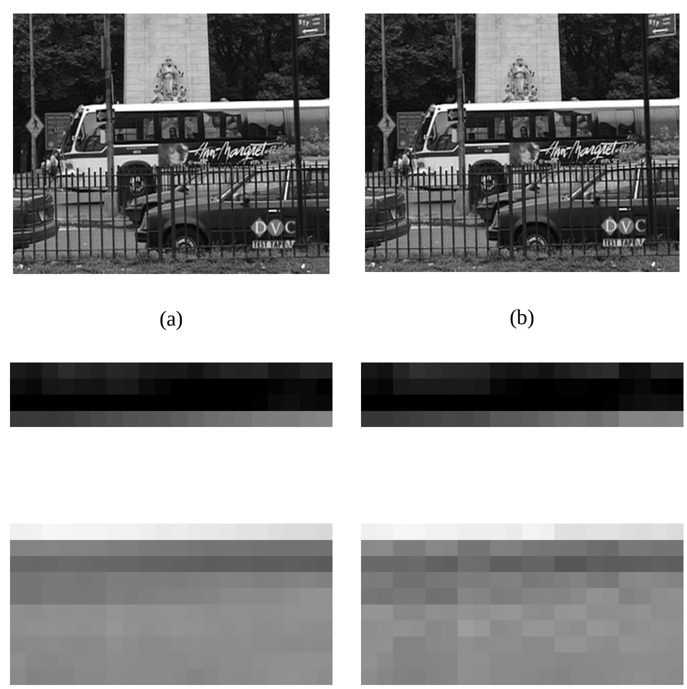
SSIM (Structured Similarity) is another widely used method for measuring the similarity between two images [12]. While it is more complex than the PSNR, the results are more close to the HVS. The similarity is calculated using 3 factors: luminance comparison, contrast comparison and structural comparison. Its value is between 0 and 1, where 1 means the perfect quality.

V. Experimental results

The watermarking algorithm was tested on four video streams: “Mobile”, “Bus” and “Flower”, which are highly detailed streams, and “Foreman”, which contains more smooth regions. The size of all video streams is 352x288 pixels.

A. *Quality Results*

Fig. 5. shows the original and the watermarked version of the same video frame. For better visibility it also contains cropped, 20x20 pixels blocks of the two frames, which contain significant edges (situated at the top of the bus on the right side of the frame). It can be seen that the noise caused by watermark signal is imperceptible.

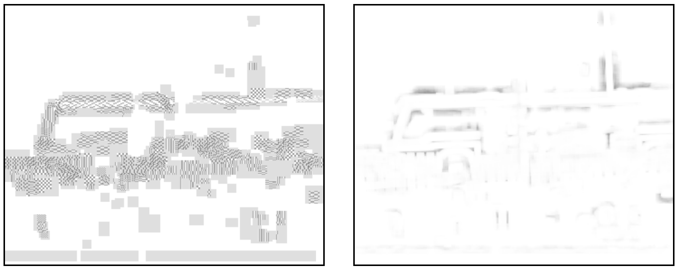


(a) (b)

(c) (d)

Fig.5. Original (a) and watermarked (b) video frame and blocks of 20x20 pixels from the original (c) and watermarked (d) video frame

Fig. 6. shows the PSNR and SSIM error maps between the original and the watermarked frame. The error maps are converted to be negative for better visibility. It can be seen that watermark is inserted around the edges. The PSNR error map shows that numerous pixels have been altered (PSNR=40,31 dB), but the SSIM error map shows a better quality (SSIM=0,9952).



(a) (b)

Fig.6. PSNR (a) and SSIM (b) error maps between original and watermarked frame (converted to negative for better visibility)

Table I presents the objective quality values of the tested videos.

TABLE I
AVERAGE PSNR AND SSIM VALUES OF WATERMARKED TEST SEQUENCES

Videos	Mobile	Foreman	Bus	Flower
PSNR[dB]	35,29	45,17	37,37	35,27
SSIM	0,9842	0,9969	0,9905	0,9905

It can be seen that the video quality depends on the content. If the content contains more edges or textured components, then more noise is added to it. Although it improves robustness of the watermark, the objective quality of the video becomes worse.

B. *Robustness of the algorithm*

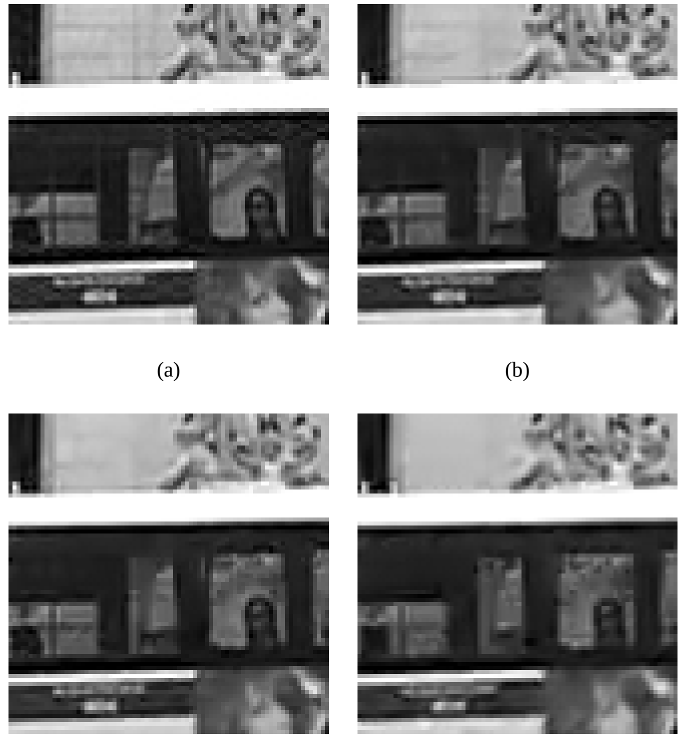
Robustness of the algorithm has been tested against lossy compression. Two codecs were used: the H.264/AVC with Baseline profile, and the XviD codec. These codecs are widely used for low bitrate video coding.

First, test data of 128 bits were embedded in the video streams; 8 bits were embedded at every frame using a watermarking strength (α) of 5. Then the watermarked videos have been compressed using the two codecs with different bitrates. Finally, the accordance was measured between the original and the extracted data produced by the detector.

Accordance of 100% means that every embedded bit can be correctly identified, while 50% means that watermark signal can not been retrieved, the embedded and the extracted data are uncorrelated.

1) *Results after H.264/AVC compression*

Fig. 7. presents blocks of 80x80 pixels cropped from a selected frame of the original, and from the compressed streams. It can be seen that the lower the bitrate of the compression is, the less details the video frame has.



(a) (b) (c) (d)

Fig.7. Blocks of 80x80 pixels of the uncompressed watermarked frame (a), and after H.264/AVC compression with bitrates 1008 kbps (b), 720 kbps (c), 384 kbps (d)

Table II presents the objective quality values of the different bitrates used at video compression.

TABLE II
AVERAGE PSNR AND SSIM VALUES OF H.264 COMPRESSION OF VIDEO "BUS"

Bitrates	1008 kbps	720 kbps	384 kbps
PSNR[dB]	33,08	31,70	28,96
SSIM	0,9364	0,9148	0,8573

Fig. 8. shows the test results of the H.264/AVC lossy compression.

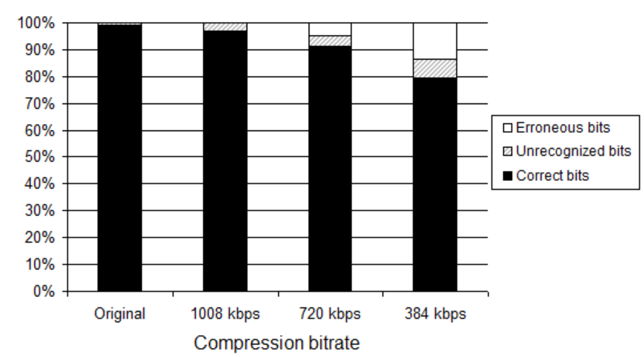


Fig. 8. Robustness results against H.264/AVC lossy compression

"Correct bits" means the number of bits that can be detected correctly.

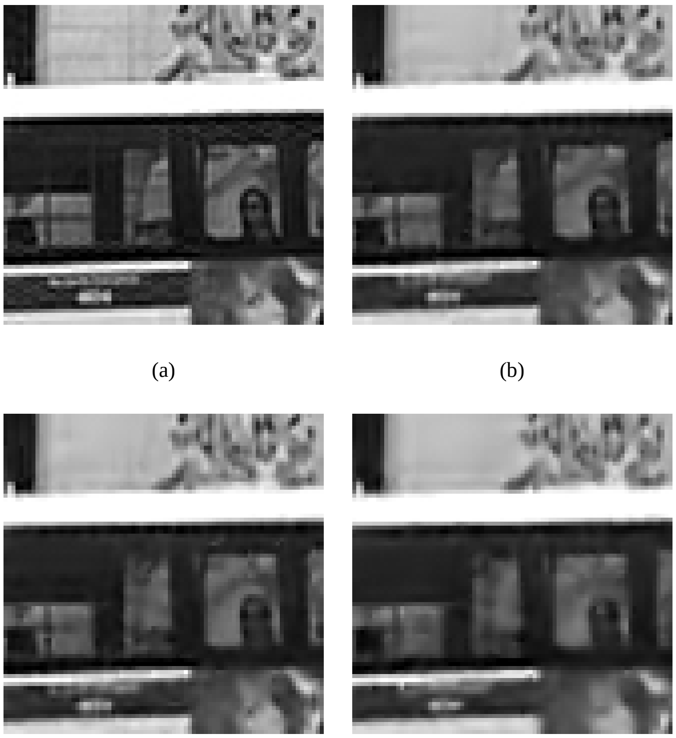
"Erroneous bits" is the number of false bits. The embedded bit was 0 and the detected was 1 and vice versa.

"Unrecognized bits" is the number of bits that the detecting algorithm can not decide whether the embedded bit was 0 or 1. Results show that the watermark can be extracted even from videos compressed using low bitrate H.264 compression.

2) Results after Xvid compression

Test results show that XviD codec produce worse image quality at the same bitrate than H.264/AVC codec.

Fig. 9. presents blocks of 80x80 pixels cropped from a selected frame of the original, and the compressed streams.



(a) (b)

(c) (d)

Fig.9. Blocks of 80x80 pixels from the uncompressed watermarked frame (a), and from the frames compressed with XviD with bitrates: 1008 kbps (b), 720 kbps (c) and 384 kbps (d)

TABLE III
AVERAGE PSNR AND SSIM VALUES OF XVID COMPRESSION OF VIDEO "BUS"

Bitrates	1008 kbps	720 kbps	384 kbps
PSNR[dB]	29,55	28,77	26,86
SSIM	0,8951	0,8709	0,8047

Fig. 10. shows the results of the tests. "Correct bits", "Erroneous bits" and "Unrecognized bits" mean the same.

Test results show that the watermark also survives XviD compression.

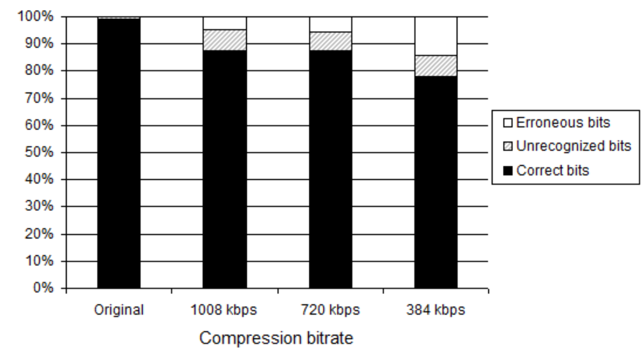


Fig. 10. Robustness results against XviD lossy compression

VI. CONCLUSION

In this paper a novel video watermarking algorithm is presented. Data embedding is made in the wavelet domain. It uses the Sobel edge detector for finding the appropriate areas for watermarking – these areas are the significant edges of the frames and their surroundings. The HVS is less sensitive to modifications on middle and high frequencies. Compression algorithms make only minor changes to these areas. The watermark itself is a pseudo random noise, which is calculated from the input data and a seed value. Watermark is detected by correlating the pixel values of the selected area with a pseudo random noise.

Test results show that the embedded watermark is imperceptible to the HVS, and the method is resistant to the H.264/AVC and the XviD lossy compression algorithms.

REFERENCES

[1] R. Schyndel, A. Tirkel and C. Osborne, "A digital watermark," IEEE Proc. Int. Conf. Image Processing, vol. 2, pp. 86–90, 1994.
[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35, no. 3-4, pp. 313-336, 1996.
[3] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in IEEE Proc. Int. Conf. Image Processing, vol. 3, pp. 219–222, 1996.
[4] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Trans. Image Processing, vol. 6, no. 12, pp. 1673–1687, 1997.
[5] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in IEEE Proc. Int. Conf. Image Processing, vol. 3, pp. 211–214, 1996.
[6] M. Barni, F. Bartolini,a and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," IEEE Trans. Image Processing, vol. 10, no. 5, pp. 783–791, 2001.
[7] J. N. Ellinas, "A robust wavelet-based watermarking algorithm using edge detection," International Journal of Computer Science vol. 2, no. 3, pp. 206–211, 2007.
[8] D. Pröfrock, M. Schlauweg, E. Müller, "A new uncompressed-domain video watermarking approach robust to H.264/AVC compression," Proc. of the IASTED Int. Conf. on Signal Processing, Pattern Recognition, and Applications, pp. 99–104, 2006.
[9] P. Schrder, W. Sweldens, M. Cohen, T. DeRose, D. Salesin, "Wavelets in Computer Graphics," SIGGRAPH 96 Course Notes.
[10] M. Nixon, A. S. Aguado, "Feature extraction & image processing," Academic Press, 2007.
[11] S. Winkler, "Digital Video Quality: Vision Models and Metrics," Wiley, 2005.
[12] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, 2004.