

Video Watermarking Algorithms Using the SVD Transform

Lama Rajab

*Department of Computer Information System
University of Jordan, Amman, Jordan
E-mail: lama.Rajab@ju.edu.jo*

Tahani Al-Khatib

*Department of Computer Information System
University of Jordan, Amman, Jordan
E-mail: tahani.khatib@ju.edu.jo*

Ali Al-Haj

*Department of Computer Engineering, Princess Sumaya University
Amman, Jordan
E-mail: ali.alhaj@ju.edu.jo*

Abstract

Video watermarking is relatively a new technology that has been proposed to solve the problem of illegal manipulation and distribution of digital video. It is the process of embedding copyright information in video bit streams. In this paper, we propose two effective, robust and imperceptible video watermarking algorithms. The two algorithms are based on the algebraic transform of Singular Value Decomposition (SVD). In the first algorithm, watermark bit information are embedded in the SVD-transformed video in a diagonal-wise fashion, and in the second algorithm bits are embedded in a blocks-wise fashion. The performance of the two proposed algorithms was evaluated with respect to imperceptibility, robustness and data payload. Both algorithms showed similar but high level of imperceptibility, however their performance varied with respect to robustness and payload. The diagonal-wise based algorithm achieved better robustness results, while the block-wise algorithm gave higher data payload rate.

Keywords: Digital Video Watermarking, Singular Value Decomposition (SVD), Robustness, Imperceptibility, Data Payload.

1. Introduction

As digital video-based application technologies grow, such as Internet video, wireless video, videophones, and video conferencing, the problem of unauthorized copying and distribution of digital video rises more and more, thus creating copyright dilemma for the multimedia industry in general, and to the audio-video industry in particular. Many researches and technologies were proposed to provide methods to solve the problem of illegal copying and manipulations of digital video. An attractive method that has been proposed a decade ago to implement copyright information in multimedia documents is digital watermarking [1,2,3].

Digital watermarking refers to embedding watermarks in a multimedia documents and files in order to protect them from illegal copying and identifying manipulations [4,5]. This promising technology received a considerable attention for embedding copyright information in a broad range of multimedia applications. In particular, video proposed watermarking techniques embed small copyright information, called a watermark, in the digital video such that the watermark is imperceptible and robust against attempts to degrade it or remove it from the digital object. Video watermarking research received less attention than image watermarking due to its inherent difficulty, however, many algorithms have already been proposed [6,7,8,9].

Video watermarking approaches can be classified into two main categories based on the method of hiding watermark information bits in the host video. The two categories are: Spatial domain watermarking, and transform-domain watermarking [10,11]. In spatial-domain watermarking techniques, embedding and detection are performed on spatial pixels values (luminance, chrominance, color space) or on the overall video frame. Spatial-domain techniques are easy to implement, however they are not robust against common digital signal processing operations such as video compression. Transform-domain techniques, on the other hand, alter spatial pixel values of the host video according to a pre-determined transform. Commonly used transforms are the Discrete Cosine Transform (DCT), the Fast Fourier Transform (FFT), the Discrete Wavelet Transform (DWT), and the Singular Value Decomposition (SVD). Transform-domain watermarking techniques proved to be more robust and imperceptible compared to spatial domain techniques since disperse the watermark in the special domain of video frame, making it very difficult to remove the embedded watermark [12,13,14,15].

In this paper, we propose two blind, imperceptible and robust video watermarking algorithms based on Singular Value Decomposition (SVD) [16]. Each algorithm embeds the watermark in the transform-domain YCbCr space thus spreading the watermark in each frame of the video. The first algorithm suggests hiding watermark information in a diagonal-wise manner in one of three SVD matrices; U , S and V . On the other hand, the second algorithm hides the watermark information in a block-wise manner in either the U or V matrices. The rest of the paper is organized as follows. The SVD transform is outlined in section 2. The two proposed video SVD-based watermarking algorithms are described in sections 3 and 4, respectively. Performance evaluation of the two algorithms with respect to imperceptibility, robustness and payload are described in section 5. Concluding remarks are given in the last section.

2. Singular Value Decomposition

Singular Value Decomposition (SVD) is a numerical technique for diagonalizing matrices in which the transformed domain consists of basis states that is optimal in some sense. The SVD of an $N \times N$ matrix A is defined by the operation:

$$A = U S V^T \quad (1)$$

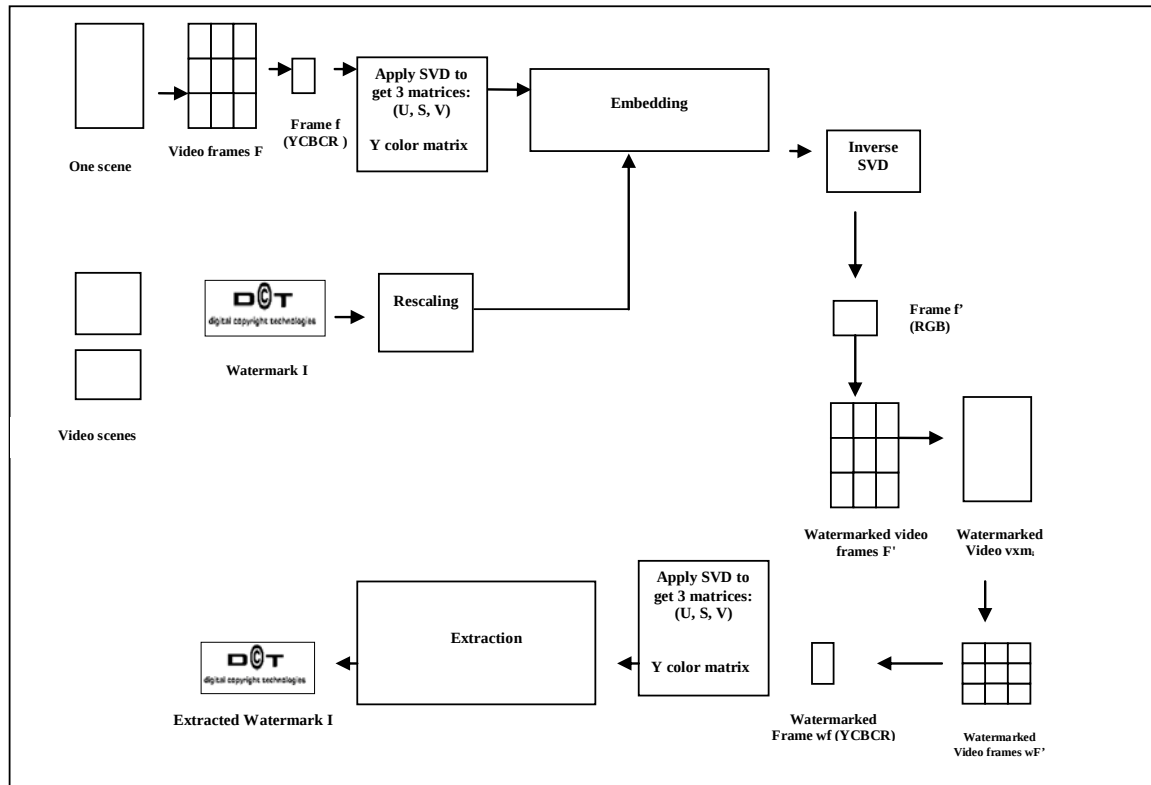
Where U and $V \in \mathbf{R}^{N \times N}$ are unitary, and $S \in \mathbf{R}^{N \times N}$ is a diagonal matrix. The diagonal entries of S are called the singular values of A and are assumed to be arranged in decreasing order $\sigma_i > \sigma_{i+1}$. The columns of the U matrix are called the left singular vectors while the columns of the V matrix are called the right singular vectors of A . Each singular value σ_i specifies the luminance of an image layer while the corresponding pair of singular vectors specifies the geometry of the image layer [17, 18, 19]. In SVD-based watermarking, a frame image is treated as a matrix decomposed into the three matrices; S , U and V^T , as shown below in Figure 1.

Figure 1: The SVD operation $SVD(A) = U S V^T$

$$SVD(A) = \begin{bmatrix} U_{1,1} & . & . & U_{1,n} \\ U_{2,1} & . & . & U_{2,n} \\ . & . & . & . \\ U_{n,1} & . & . & U_{n,n} \end{bmatrix} \begin{bmatrix} \sigma_{11} & 0 & 0 & 0 \\ 0 & \sigma_{22} & 0 & 0 \\ . & . & . & . \\ 0 & 0 & 0 & \sigma_m \end{bmatrix} \begin{bmatrix} V_{1,1} & . & . & V_{1,n} \\ V_{2,1} & . & . & V_{2,n} \\ . & . & . & . \\ V_{n,1} & . & . & V_{n,n} \end{bmatrix}^T$$

3. First Proposed SVD-Based Video Watermarking Algorithm

The first algorithm is based on transforming the host video using the SVD operator and then embedding the watermark information in the S, U, or V matrices diagonal-wise. The proposed algorithm consists of two procedures, the first embeds the watermark into the original video clip, while the other extracts it from the watermarked version of the video clip. We embed only the foreground pixels in the watermark. A block diagram showing the embedding and extraction procedure of the algorithm is shown in Figure 2.

Figure 2: Block diagram for the first algorithm's procedures.

3.1. Watermark Embedding Procedure

The embedding procedure of the first algorithm, with its three possible variations, is described in details in the following steps:

Step 1: Divide the video clip into video scenes Vs_i

Step 2: Process the frames of each video scene using SVD described in steps 3 ~ 9 below.

Step 3: Convert every video frame F from RGB to YCBCR color matrix format.

Step 4: Compute the SVD for the Y matrix in each frame F . This operation generates 3 Matrices (U,S,V) such as:

$$Y = U_Y S_Y V_Y \quad (2)$$

Step 5: Rescale the watermark image so that the size, of the watermark will match the size of the matrix which will be used for embedding either U, V or S.

Step 6: Embedding can be done in one of the three SVD matrices: U, V, or S, as follows:

Embedding in Matrix U Diagonal-wise

- inverse each diagonal value ($u_{i,i}$) in The U matrix, such as $x = 1/u_{i,i}$
- Embed the binary bits of the watermark W_{Vsi} into the integer part of x by substituting the watermark bit W_i with the 7th bit of x.
- Apply the inverse to each x, to get the modified values of U matrix, such as $u_{i,i}' = 1/x'$.
- Apply inverse SVD on the modified coefficient matrix U' . such as:

$$Y' = U_Y' S_Y V_Y^T \quad (3)$$

Embedding in Matrix V Diagonal-wise

- Inverse each diagonal value ($v_{i,i}$) in The V matrix , such as $x = 1/v_{i,i}$
- Embed the binary bits of the watermark W_{Vsi} into the integer part of x by substituting the watermark bit W_i with the 7th bit of x.
- Apply the inverse to each x, to get the modified values of V matrix, such as $v_{i,i}' = 1/x'$.
- Apply inverse SVD on the modified coefficient matrix V' . Such as:

$$Y' = U_Y S_Y V_Y'^T \quad (4)$$

Embedding in Matrix S Diagonal-wise

- Embed the binary bits of the watermark W_{Vsi} into the integer part of each diagonal value of the S matrix, $s_{i,i}$ by substituting the watermark bit W_i with the 7th bit of $s_{i,i}$
- Apply inverse SVD on the modified coefficient matrix S' . such as:

$$Y' = U_Y S_Y' V_Y^T \quad (5)$$

Where Y' is the updated luminance in the YCBCR color representation. This operation produces the final watermarked Video frame F' .

Step 7: Convert the video frames F' from YCBCR to RGB color matrix.

Step 8: Reconstruct frames into the final watermarked Video scene V_{si}' .

Step 9: Reconstruct watermarked scenes to get the final watermarked Video clip.

3.2. Watermark Extraction Procedure

This proposed algorithm is blind in the sense that it does not need the original video in the extraction process. Therefore, we can extract the watermark image from the watermarked video frames directly as described in details in the following steps:

Step 1: Divide the watermarked Video clip V' into watermarked scenes V_{si}' .

Step 2: Process the watermarked frames of each watermarked video scene using SVD as described in steps 3 ~ 7.

Step 3: Convert the video frame F' from RGB color matrix to YCBCR.

Step 4: Compute the SVD for the Y matrix in frame F' . , this operation generates 3 Matrices (U, S, V).

Step 5: Extraction is done in one of the three SVD matrices: U, V, or S, as follows:

Extraction from Matrix U

- Inverse each diagonal value ($u_{i,i}$) in The U matrix , such as $x = 1/u_{i,i}$
- Extract the embedded watermark from the integer part of x as follows:

$$W_{Vsi}(i) = 7^{\text{th}} \text{LSB}(\text{fix}(x)) \quad (6)$$

Extraction from Matrix

- Inverse each diagonal value ($v_{i,i}$) in The V matrix , such as $x = 1/v_{i,i}$.

- b. Extract the embedded watermark from the integer part of x using:

$$W_{Vsi}(i) = 7^{\text{th}} \text{LSB}(\text{fix}(x)) \quad (7)$$

Extraction from Matrix S

Extract the embedded watermark from the integer part of each diagonal value of the S matrix, $s_{i,i}$ as follows:

$$W_{Vsi}(i) = 7^{\text{th}} \text{LSB}(\text{fix}(s_{i,i})) \quad (8)$$

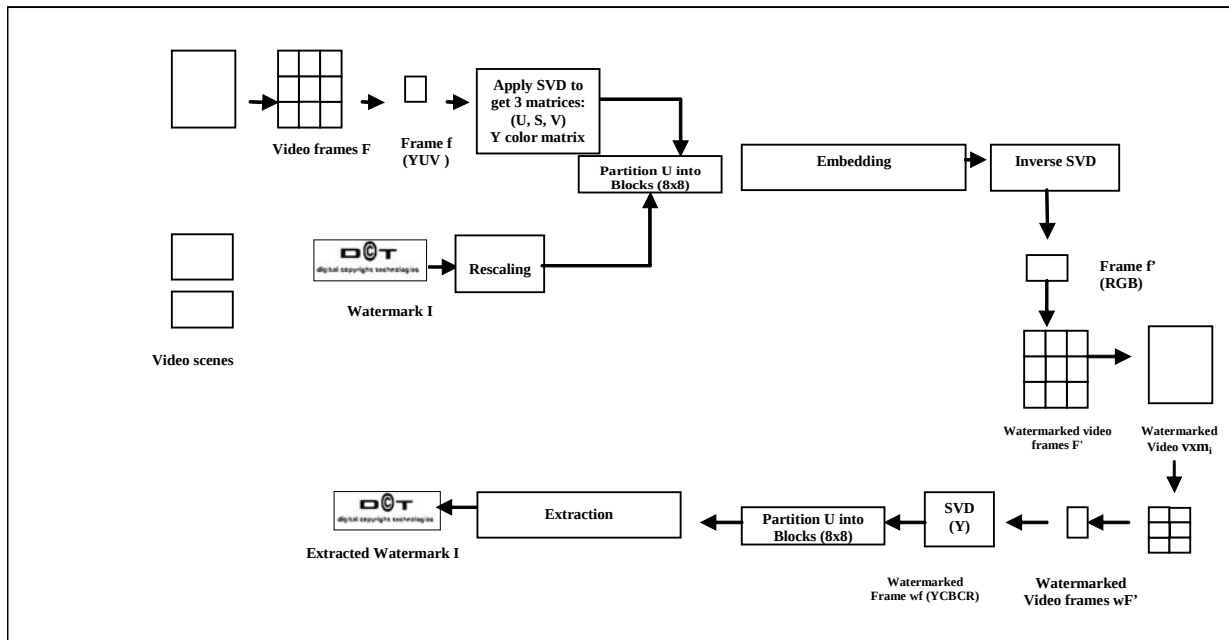
Step 6: Construct the image watermark W_{Vsi} by cascading all watermark bits extracted from all frames.

Step 7: Repeat the same procedure for all video scenes.

4. Second Proposed SVD-Based Video Watermarking Algorithm

The second algorithm is based on transforming the host video using the SVD operator and then embedding the watermark information in the U, or V matrices in a block-wise fashion. The proposed algorithm consists of two procedures, the first embeds the watermark into the original video clip, while the other extracts it from the watermarked version of the video clip. We embed only the foreground pixels in the watermark. A block diagram showing the embedding and extraction procedure of the second algorithm is shown in Figure 3.

Figure 3: Block diagram for the second algorithm's procedures



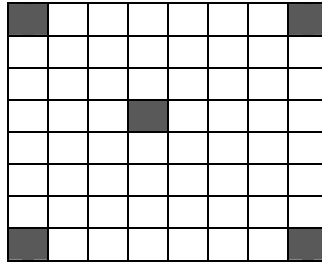
4.1. Watermark Embedding Procedure

The embedding procedure of the second algorithm, with its three possible variations, is described in details in the following steps:

- Step 1:** Divide the video clip into video scenes Vs_i
- Step 2:** Process the frames of each video scene using SVD described in steps 3 ~ 8 below.
- Step 3:** Convert every video frame F from RGB to YCBGR color matrix format.
- Step 4:** Compute the SVD for the Y matrix in each frame F . This operation generates 3 Matrices (U, S, V).
- Step 5:** Rescale the watermark image so that the size, of the watermark will match the size of the matrix which will be used for embedding U or V.

Embedding in Matrix U Block-wise

- Partition the U matrix into blocks with size 8x8.
- Inverse the pixel value for 5 predetermined pixels in each odd block shown in figure 4 in the U matrix such as $x = 1 / \text{pixel value}$.

Figure 4: Positions used for embedding watermark bits in each 8x8 block.

- Embed the binary bits of the watermark W_{vsi} into the integer part of x by substituting the watermark bit W_i with the 7th bit of x .
- Apply the inverse to each x , to get the modified values of U matrix.
- Apply inverse SVD on the modified coefficient matrix U' . Such as:

$$Y' = U_Y' S_Y V_Y^T \quad (9)$$

Embedding in Matrix V Block-wise

- Partition the V matrix into blocks with size 8x8.
- inverse the pixel value for 5 predetermined pixels in each odd block shown in figure 4 in each odd block in The V matrix such as $x = 1 / \text{pixel value}$.
- Embed the binary bits of the watermark W_{vsi} into the integer part of x by substituting the watermark bit W_i with the 7th bit of x
- Apply the inverse to each x , to get the modified values of V matrix.
- Apply inverse SVD on the modified coefficient matrix V' . such as:

$$Y' = U_Y' S_Y V_Y^T \quad (10)$$

Where Y' is the updated luminance in the YCBCR color representation.

Step 6: Convert the video frames F' from YCBCR to RGB color matrix.

Step 7: Reconstruct frames into the final watermarked Video scene V_{si}' .

Step 8: Reconstruct watermarked scenes to get the final watermarked Video

4.2. Watermark Extraction Procedure

The proposed algorithm is also blind in the sense that it does not need the original video in the extraction process. Therefore, we can extract the watermark image from the watermarked video frames directly as described in details in the following steps:

Step 1: Divide the watermarked Video clip V' into watermarked scenes V_{si}'

Step 2: Process the watermarked frames of each watermarked video scene using SVD as described in steps 3 ~ 8.

Step 3: Convert the video frame F' from RGB color matrix to YCBCR.

Step 4: Compute the SVD for the Y matrix in frame F' . , this operation generates 3 Matrices (U, S, V).

Step 5: inverse the pixel value for 5 pixels in each odd block used in the embedding process in The U matrix if it is used in the embedding, or in the V matrix , such as $x = 1 / \text{pixel value}$.

Step 6: Extract the embedded watermark from the integer part of x .

- Step 7:** Construct the image watermark W_{vsi} by cascading all watermark bits extracted from all frames.
- Step 8:** Repeat the same procedure for all video scenes.

5. Experimental Results and Performance Evaluation

We evaluated the performance of the two proposed video watermarking algorithms, with their different variations, using two colored host video clips. Each of the two video clips was partitioned into three scenes with having different number of frames. The first clip was partitioned into three scenes with 70, 101 and 85 frames, and was used to evaluate the performance of the first algorithm. The second video clip, was partitioned into three scenes with 60, 43 and 56 frames, and was used to evaluate performance of the second algorithm. The watermark used in our experiments was a binary image. Snapshots from the video and the watermark are shown in Figure 5.

Figure 5: Snapshots from the video clips and the watermark.



As stated earlier, performances of the proposed algorithms are evaluated with respect to three metrics: imperceptibility, robustness and payload [20].

5.1. Imperceptibility Performance

Imperceptibility means that the perceived quality of the video clip should not be distorted by the presence of the watermark. As a measure of the quality of a watermarked video, the peak signal to noise ratio (PSNR) is typically used [21]. In our work, the watermark was embedded in the video according to the two algorithms described sections three and four. The average PSNR for the all frames of the three watermarked scenes was 48.1308. This high PSNR value proves imperceptibility of the proposed algorithms.

5.2. Robustness Performance

Robustness of a watermarking algorithm is a measure of the immunity or resistance of the watermark against attempts to remove or degrade it from the video frames by different types of digital signal processing attacks [22]. The similarity between the original watermark and the extracted watermark from the attacked watermarked video frames was measured by using the correlation factor ρ , which is computed using the following Equation:

$$\rho(w, \hat{w}) = \frac{\sum_{i=1}^N w_i \hat{w}_i}{\sqrt{\sum_{i=1}^N w_i^2} \sqrt{\sum_{i=1}^N \hat{w}_i^2}} \quad (11)$$

Where N is the number of pixels in watermark, w and $\hat{w}$ are the original and extracted watermarks respectively. The correlation factor ρ may take values between 0 and 1.

We evaluated robustness of the two algorithms against the following video attacks: *JPEG compression, video frame rotation, noise attacks (Gaussian, and salt and pepper noise), frame dropping and frame swapping & averaging.*

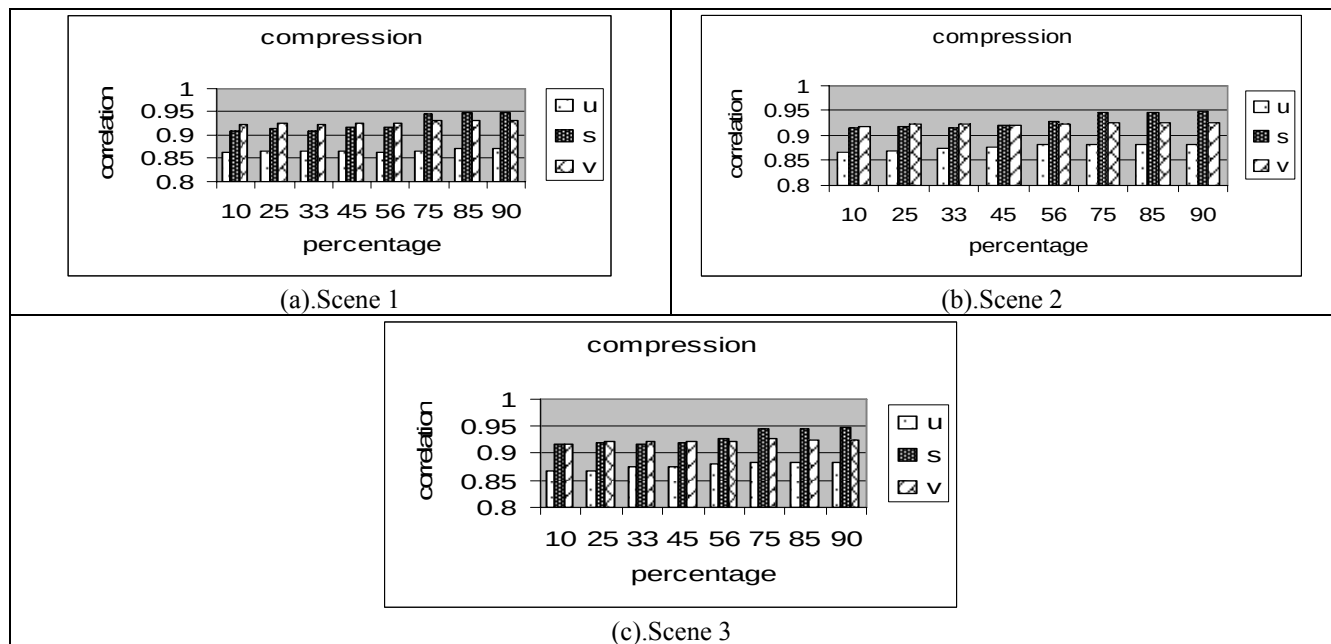
5.2.1 Robustness of the First Algorithm

As we have mentioned later, embedding in the first algorithm is done diagonal-wise in one of the matrices produced by the SVD operator: the U matrix, the V matrix, or the S matrix. In what follows, we report on the robustness performance of the algorithms for the three possible embedding matrices.

(i). JPEG Compression

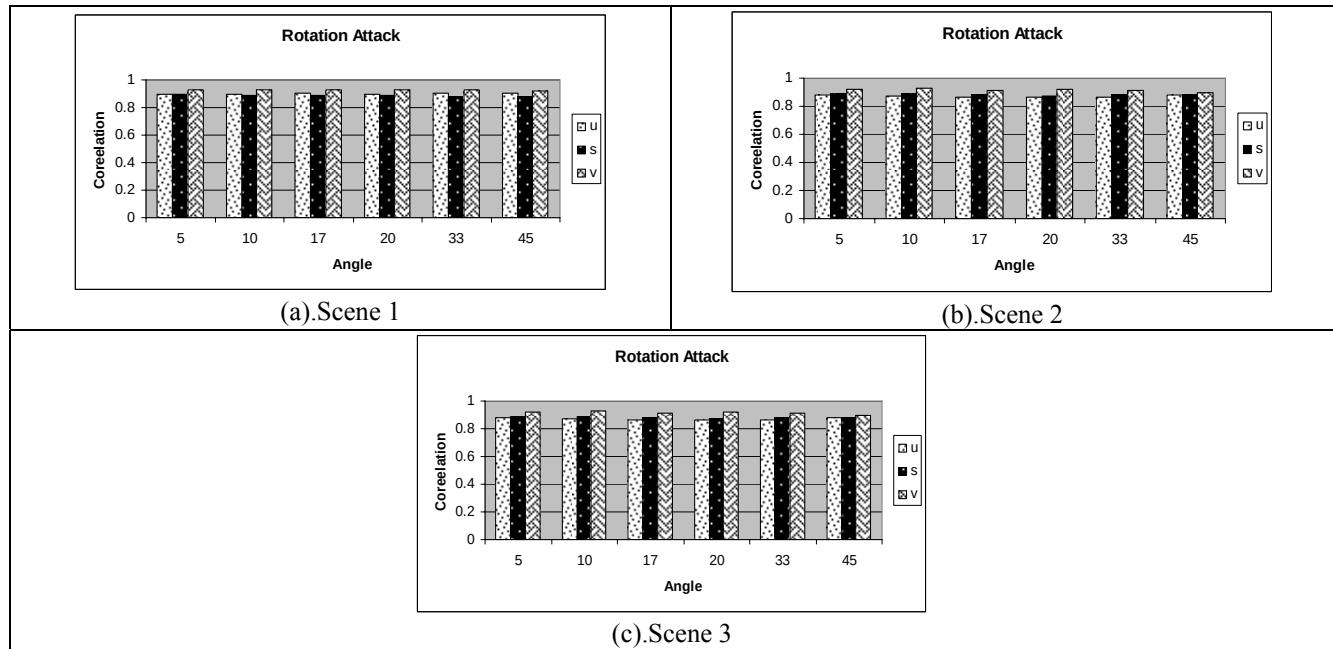
The watermarked video frames were compressed with different quality factors. As shown in Figure 6 the correlation values indicate clearly the high robustness of the proposed algorithm across all matrices. However, its clearly seen that embedding in the S matrix gives the highest robustness results in the three scenes.

Figure 6: First algorithm's correlation values due to different JPEG compression rates.



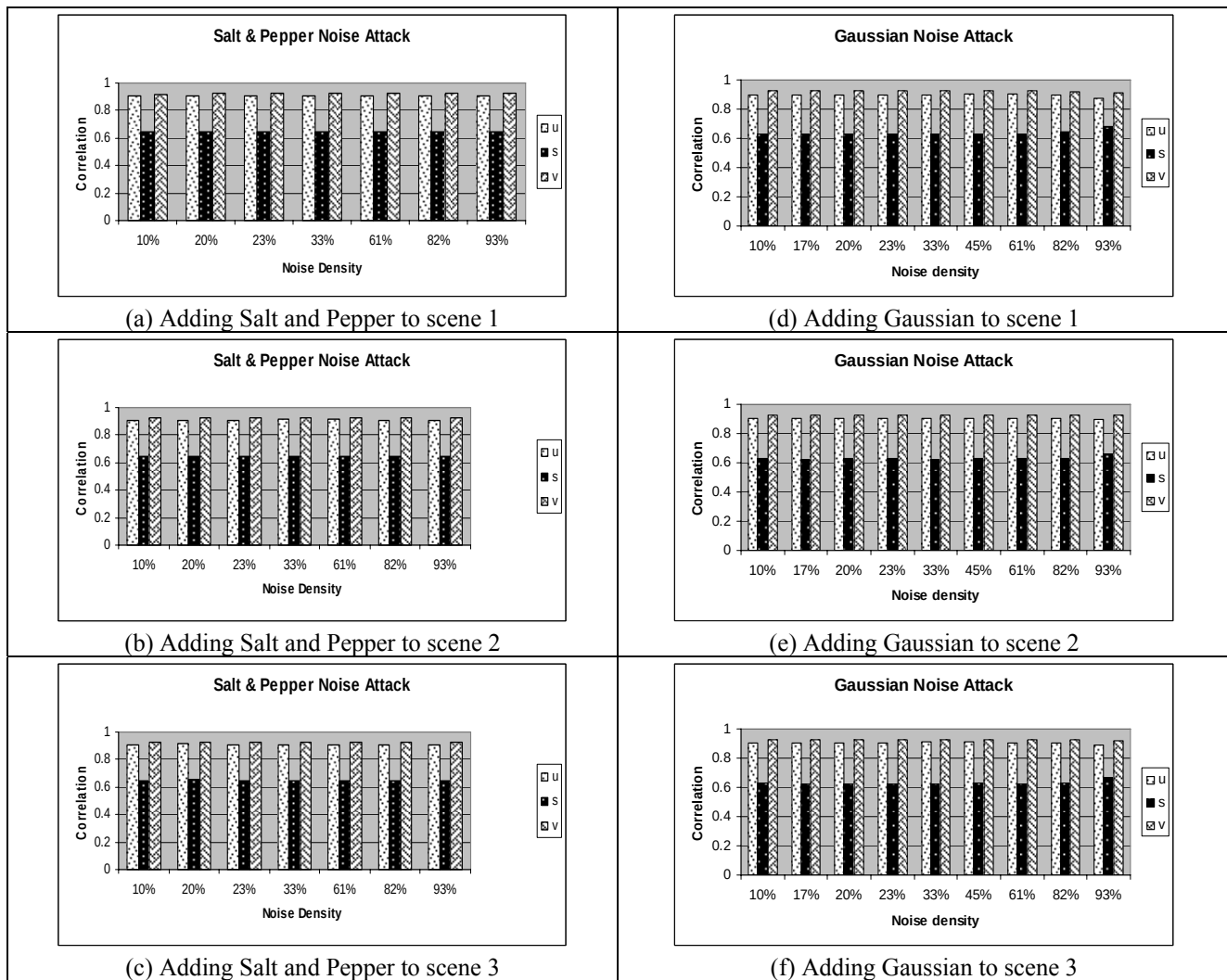
(ii). Video Angular Rotation

The watermarked video frames for the three scenes were rotated with different angles. As shown in Figure 7, the correlation values generally indicate robustness of the first algorithm against the video frames rotation. However, embedding the watermark diagonal-wise in the V matrix resulted in slightly higher robustness compared to S or U matrices for all angles.

Figure 7: First algorithm's Correlation values due to different angular rotation values.**(iii). Gaussian and Salt and Pepper Noises**

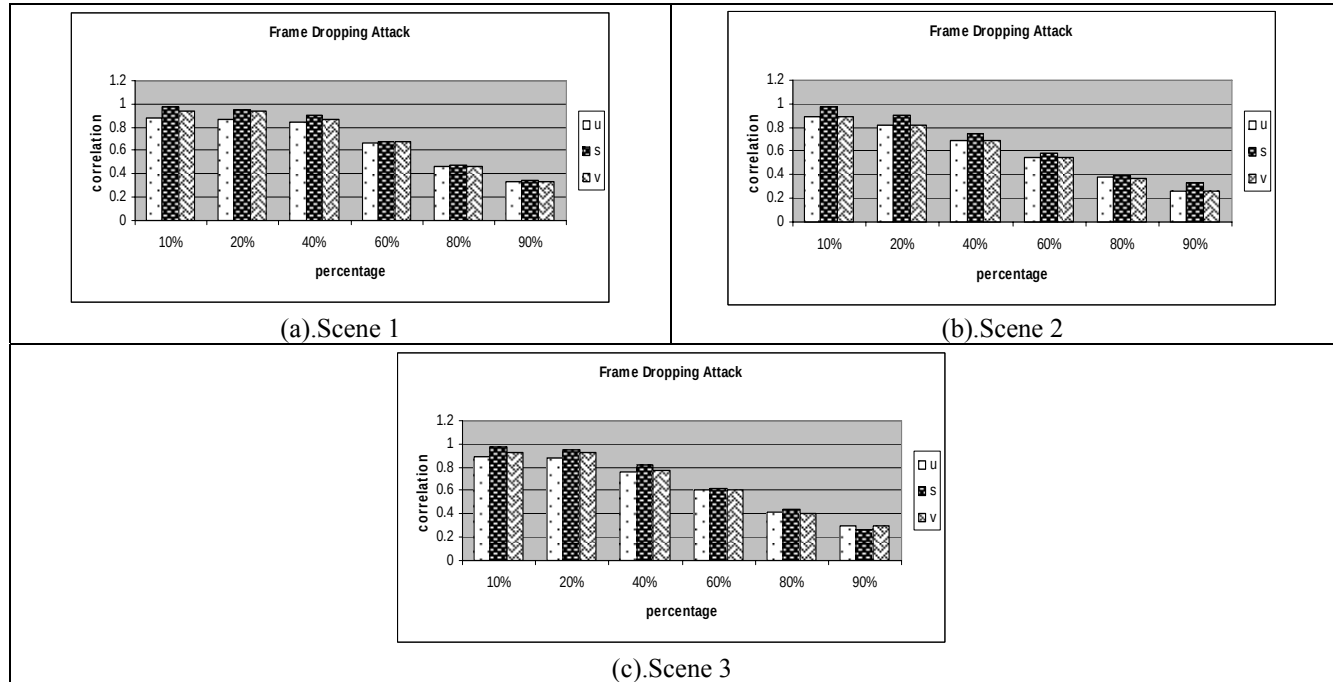
Two kinds of common additive noise were added with varying intensities to the watermarked video frames for the three scenes; Gaussian and Salt and pepper. Each noise was tested separately, but showed relatively similar results. As shown in Figure 8, results generally indicate robustness of the first algorithm against addition of Gaussian and salt and pepper noise. As shown in the figures below, embedding the watermark in diagonal of S matrix resulted in lower robustness compared to V or U matrices for all densities.

Figure 8: First algorithm's Correlation values due to the addition of two types of noise; salt and pepper (a-c) and Gaussian noise (d-f) at different densities of noise for the three scenes.

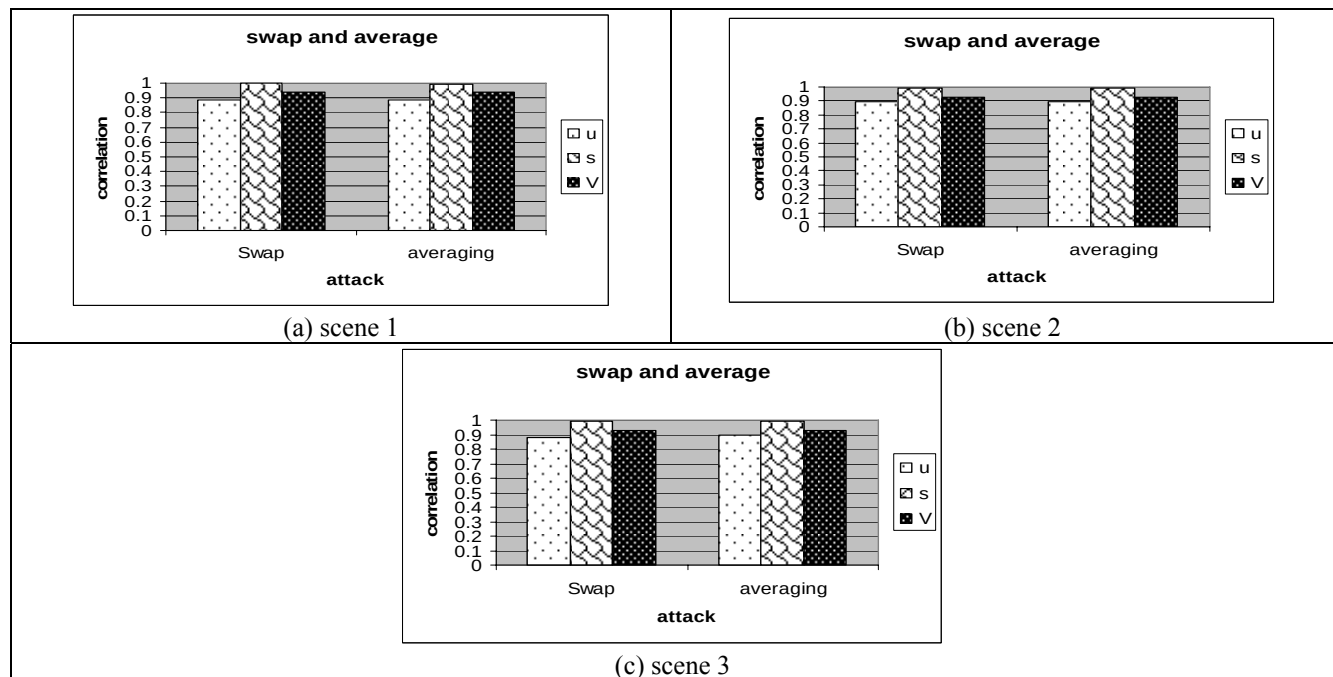


(iv). Frame Dropping

The attackers hope by performing frame dropping attack that the embedded watermark will be degraded or removed without hindering the original video. This is due to the fact that large amount of redundancy exist between video frames, and therefore video dropping should leave the integrity of the original video intact. The results are shown in Figure 9. As seen, even if the attacker drops 60% of the frames, the watermark can still be extracted with an acceptable correlation value. Results also show that embedding the watermark in S matrix achieved relatively better correlation after frame-dropping compared with the other two matrices.

Figure 9: First algorithm's correlation values due to Frame dropping.**(v). Frame Swapping and Averaging**

Other than the frame dropping attack, we evaluated robustness due to video frame averaging and swapping. Frame swapping was performed by taking two random frames from the video and swapping them, then trying to extract the watermark. While averaging is performed by taking two random frames pixels, then taking their average, and replacing the original pixels with the averaged ones. Results show that embedding the watermark in S matrix achieved better correlation for both attacks than U and V matrices in the three scenes. Furthermore embedding in V showed slightly better results embedding in U .

Figure 10: First algorithm's Correlation values due to Frame Swapping and frame averaging.

5.2.2 Robustness of the Second Algorithm

As described in section 4, embedding in the second algorithm was done block-wise in either the U or V matrices of each SVD-transformed video frame. The algorithm embeds the watermark pixels into odd-numbered blocks of the selected matrix rather than embedding into the diagonal pixels of the matrix. Five locations were chosen in the selected odd-numbered blocks to embed watermark pixels. Therefore, five watermark pixels were embedded in each block in the U or V matrices. We performed the same robustness experiments which we used for the first algorithm. The results obtained showed a high degree of similarity with the robustness results obtained for the first algorithm. The results are not shown here to avoid redundancy.

5.3. Payload

Data payload or watermarking capacity for a given host video clip is defined as the number of watermark pixels that can be embedded in the host video without causing any visual distortion in the video. To compare the two algorithms in terms of payload, suppose that we have a scene with 56 frames and an SVD matrix (U , S , V) with a dimension 240×240 . The payload of the first algorithm can be found by multiplying number of frames by number of diagonal elements in the matrix. This results in 13440 pixels. On the other hand, the payload of the second algorithm is found by multiplying number of frames by number of blocks per frame by number of pixel per frame. This results in 252000 pixels. This shows clearly that the payload of the second block-wise algorithm is much larger than the first algorithm. Therefore, the second algorithm could be easily recommended to be adopted when large watermarks are needed in video watermarking.

6. Conclusions

In this paper, two SVD-based digital video watermarking algorithms were proposed. In the first algorithm, watermark information was embedded in the diagonal elements of S , U , or V matrices. On the other hand, embedding watermark information in second algorithm was done block-wise in the U or V matrices of each SVD-transformed frame. The two algorithms were evaluated in terms of three metrics; imperceptibility, robustness, and data payload. Both algorithms performed equally well with respect to imperceptibility, however their performance slightly varied with respect to robustness and payload. Embedding watermark information in the diagonal elements of matrix U or matrix V showed more robustness against noise than embedding in matrix S . To be specific, embedding in matrix V gave slightly better results than embedding in matrix U . However, embedding in the diagonal elements of matrix S produced higher robustness values against attacks such as frame compression, rotation, frame dropping, frame swapping and frame averaging attacks. The block-wise based embedding in the second algorithm allows larger watermark to be hidden in the host video, compared with the diagonal-wise embedding of the first algorithm. This gives a clear superiority of the second algorithm with respect to data payload.

References

- [1] Langelaar, G., I. Setyawan, and R. Lagendijk, 2000. "Watermarking Digital Image and Video Data: A State-of-Art Overview", *IEEE Signal Processing Magazine* 17, pp. 20-46.
- [2] V. Potdar, S. Han, and E. Chang, 2005. "A Survey of Digital Image Watermarking Techniques", in *Proceedings of the 2005 IEEE International Conference on Industrial Informatics*, pp. 709-716.
- [3] M. Ramkumar and A. Akansu, 2004. "A Robust Protocol for Proving Ownership of Multimedia Content", *IEEE Trans. Multimedia* 6, pp. 496-478.
- [4] L. Qiao and K. Nahrstedt, 1998. "Watermarking Schemes and Protocols For Protecting Rightful Ownership and Customer's Rights", *Journal of Visual Commun. and Image Represent* 9, pp.194–210.
- [5] M. Arnold, M. Schumucker, and S. Wolthusen, 2003. "Techniques and Applications of Digital Watermarking and Content Protection". Artech House.
- [6] Doerr, G., and J. Dugelay, 2003. "A Guided Tour to Video Watermarking", *Signal Processing: Image Communication* 18, pp. 263-282.
- [7] Hartung, H., and B. Girod, 1998. "Watermarking of Compressed and Un-Compressed Video", *Signal Processing* 66, pp. 283-301.
- [8] P. Chan and M. Lyu, 2003. "A DWT-Based Digital Video Watermarking Scheme with Error Correcting Code", in *Proceedings of the 5th International Conference on Information and Communications Security*, Springer Berlin/Heidelberg 2836, pp. 202-213.
- [9] Gao, X., and X. Tang, 2002. "Unsupervised Video-Shot Segmentation and Model-Free Anchorperson Detection for News Video Story Parsing", *IEEE Trans. Circuits and Systems for Video Technology* 12, pp.765-776.
- [10] D. Mukherjee, S. Maitra, and S. Acton, 2004. "Spatial Domain Digital Watermarking of Multimedia Objects for Buyer Authentication", *IEEE Trans. Multimedia* 6, pp. 1-15.
- [11] R. Shah, A. Argawal, and S. Ganesan, 2005. "Frequency Domain Real Time Digital Watermarking", in *Proc. of the IEEE 2005 Int. Conf. on Elector Info. Tech*, pp. 1-6.
- [12] S. Mitra, 1998. "*Digital Signal Processing*", McGraw–Hill, USA.
- [13] M. Hernandez, M. Miyatake, and H. Meana, 2005. "Analysis of a DFT-based watermarking algorithm", in *Proc. of the IEEE 2nd Int. Conf. on Electrical and Electronics Eng.*, pp. 44-47.
- [14] Mallat, S, 1989. "A theory for multi-resolution signal decomposition: The wavelet representation", *IEEE Trans. Pattern Anal. And Machine Intell.* 11, pp. 674-693.
- [15] Reddy, A., and B. Chatterji, 2005. "A New Wavelet Based Logo-watermarking Scheme", *Pattern Recognition Letters* 26, pp. 1019-1027.
- [16] Andrews,H., and C. Patterson, 1976. "Singular Value decompositions and Digital Image Processing", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 24, pp. 26-53.
- [17] Liu, R., and T. Tan, 2002. "A SVD-Based Watermarking Scheme for Protecting Rightful Ownership", *IEEE Trans. Multimedia* 4, pp.121-128.
- [18] Chang, C., P.Tsai, and C. Lin, 2005. "SVD-based digital image watermarking scheme", *Pattern Recognition Letters* 26, pp.1577-1586.
- [19] Wu, Y, 2005. "On the Security of SVD-Based Ownership Watermarking", *IEEE Trans. Multimedia* 7, pp. 624-627.
- [20] Fabien A.P. Petitcolas, Ross J. Anderson, 1999. "Evaluation of Copyright Marking Systems", *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)* 1, pp. 574-579.
- [21] Voloshynovskiy, S., S. Pereira, and T. Pun, 2001. "Attacks on Digital Watermarks: Classification, Estimation-Based Attacks, and Benchmarks". *Communications Magazine* 39, pp.118-126.
- [22] D. Kundur, K. Su and D. Hatzinakos, 2004. "Digital Video Watermarking: Techniques, Technology and Trends", in *Intelligent Watermarking Techniques*, Chapter 10, P. J.-S. Pan, H.-C. Huang and L. Jain, eds., World Scientific Publishing Company. pp. 265-314.