A Comparative performance evaluation of SVD and Schur Decompositions for Image Watermarking

B.Chandra Mohan
Professor, ECE dept
Bapatla Engineering College
Bapatla-522101

K.Veera Swamy
Professor, ECE dept
QIS College of Engg
Ongole

S.Srinivas Kumar
Professor, ECE dept
JNTU College of Engg
Kakinada

ABSTRACT

In this paper, the performance of SVD and Schur decomposition is evaluated and compared for image copyright protection applications. The watermark image is embedded in the cover image by using Quantization Index Modulus Modulation (QIMM) and Quantization Index Modulation (QIM). Watermark image is embedded in the D matrix of Schur decomposition and Singular Value Decomposition (SVD). Watermarking in SVD domain is highly flexible . This is due to the availability of three matrices for watermarking. Singular values in SVD and Schur decomposition are highly stable. Compared to Singular Value Decomposition (SVD), Schur decomposition is computationally faster and robust to image attacks. The proposed algorithms based on SVD and Schur decompositions are more secure and robust to various attacks, viz., rotation, low pass filtering, median filtering, resizing, salt & pepper noise. Superior experimental results are observed with the proposed algorithm over a recent scheme proposed by Chung et al. in terms of Normalized Cross correlation (NCC) and Peak Signal to Noise Ratio (PSNR).

General Terms

Image Processing, Image Watermarking Algorithms

Keywords: Digital Image Watermarking, Schur Decomposition, SVD, PSNR and NCC

1. INTRODUCTION

Distribution of multimedia data such as images, video and audio over the internet requires secure computer networks. Multimedia data can be duplicated and distributed with out the owner's consent. Digital watermarking technique is a viable solution proposed to tackle this complex issue. Digital watermarking is a branch of information hiding which is used to hide proprietary information (company logo) in digital media like digital images, digital music, or digital video. Image watermarking has attracted a lot of attention in the research community compared to video watermarking and audio watermarking. This is due to the availability of various types of images and amount of redundant information present in images. Digital image watermarking also called watermark insertion or watermark embedding represents the scheme that inserts the hidden information into an image known as host or cover image [1]. The hidden information may be the serial number, the random number sequence, copyright messages, logos or any ownership identifiers called the watermark. After inserting or embedding the watermark by using specific algorithms, the cover image will be slightly modified and the modified image is called the watermarked image. There might be no or little perceptible difference between the host image and watermarked image. One major application of digital image watermarking is copyright protection. After embedding the watermark, the watermarked image is sent to the receiver via the Internet or transmission channel.

The watermark image is to be sustained against various attacks, viz., filtering, compression, cropping, and rotation, etc., on watermarked image. The algorithms proposed so far can be classified according to the embedding domain of the cover image. They are, spatial, transform, and hybrid domain watermarking algorithms.

Three major requirements of a digital watermarking system are imperceptibility, robustness, and capacity. Imperceptibility is defined as "perceptual similarity between the original and the watermarked versions of the cover work". Robustness is the "ability to detect the watermark after common signal processing operations". Capacity describes amount of data that should be embedded as a watermark to successfully detect during extraction.

Basically there are two main types of watermarks that can be embedded within an image, viz., pseudo random gaussian sequence and logo (binary or grey) image watermarks. Based on the type of watermark embedded, an appropriate decoder is to be designed to detect the presence of watermark.

Watermarking in transform domain is more secure and robust. Several transforms like Discrete Cosine Transform (DCT) [2,10], Discrete Hadamard Transform (DHT) [5], Discrete Wavelet Transform (DWT) [3,4], Contourlet Transform (CT) [9], and Singular Value Decomposition (SVD) [6,7,8] are common transforms used in image watermarking. Each of these transforms has its own characteristics and represents the image in different ways.

SVD and Schur decompositions are two mathematical tools used to analyze matrices. When some perturbation occurs in the watermarked image, the extraction of the watermark is not affected much. SVD is computationally expensive. Watermarking using Schur decomposition is faster compared to SVD decomposition. In this work, the performance of SVD and Schur decomposition in image watermarking application is compared. Three scalar quantization schemes are adopted for the watermark embedding.

This paper is organized as follows. SVD and Schur decomposition are discussed in section 2. Scalar quantization and proposed algorithm is presented in sections 3 and 4 respectively. Experimental results are presented in section 5. Conclusions are given in section 6.

2. SVD AND SCHUR DECOMPOSITION

SVD is a mathematical tool used to analyze matrices. In SVD, a square matrix is decomposed into three matrices of same size. A real matrix g of size $N \times N$ is decomposed into three matrices $(U, D,$ and $V)$ of same size. U and V are orthogonal matrices, i.e., $U^T U = I$ and $V^T V = I$. Here, I is an identity matrix. Superscript T indicates transpose operation. D is a square diagonal matrix given by $D = diag(\lambda_1, \lambda_2 ... \lambda_r)$, where, the diagonal entries $\lambda_1, \lambda_2 ... \lambda_r$ are known as singular values of g. Here, r is the rank of the matrix g.The columns of U are called left singular vectors of g and the columns of V are called the right singular vectors of g. This decomposition is known as Singular Value Decomposition and can be represented as

$$SVD(g) = [U \quad D \quad V] \qquad (2.1)$$

$$SVD(g) = \lambda_1 U_1 V_1^T + \lambda_2 U_2 V_2^T + ... + \lambda_r U_r V_r^T \qquad (2.2)$$

$$\hat{g} = UDV^T \qquad (2.3)$$

where, U and V are real $N \times N$ unitary matrices with small values. D is a diagonal matrix of $N \times N$ size with large singular values. $\hat{g}$ is the reconstructed matrix after applying inverse SVD transformation. The singular values satisfy the relation $\lambda_1 \geq \lambda_2 ... \geq \lambda_r \geq 0$. Each singular value specifies the luminance of an image layer while the corresponding pair of singular vectors specifies the geometry of the image. SVD matrix of an image has good stability. Singular values have three important properties:

- Singular values of the image are stable, i.e., when a small perturbation is added to an image, variance of its singular values does not occur.
- Singular values exhibit the algebraic and geometric invariance to some extent.
- Singular values represent the algebraic attributions of an image which are intrinsic and not visual.

These three properties of stability, algebraic and geometric invariance of the singular values are utilized to embed the watermark image in the largest singular values of the *D* matrix of the cover image. Image attacks tend to modify the singular values of the cover image. However, this variation in the singular values is less. Hence, the effect on the quality of the watermarked image in terms of $PSNR$ and NCC of the extracted watermark is less.

Advantages of SVD are as follows:

- Optimal matrix decomposition, maximum packing efficiency in the few coefficients.
- Flexible for watermark embedding.
- Highly stable singular values

However, one major drawback of SVD is, it's computational complexity. The computational complexity of Schur decomposition is less compared to SVD and hence it is useful for real time applications. Schur decomposition can be applied to any real matrix. There are two versions of this decomposition: the complex Schur decomposition and the real Schur decomposition. In complex schur decomposition, Decomposition (complex version): $g = UTU'$, where U is a unitary matrix. U' is the conjugate transpose of U, and T is an upper triangular matrix called the complex Schur form which has the eigen values of g along its diagonal. Schur Decomposition (real version) is given by

$$g = VSV'$$

where g, V, S and V' are matrices that contain real numbers only. In this case, V is an orthogonal matrix, V' is the transpose of V, and S is a block upper triangular called the real Schur form. Schur decomposition requires about $\frac{8}{3}N^3$ flops. SVD computation requires $11N^3$ flops. Eigen values in the Schur decomposition are also highly stable.

3. SCALAR QUANTIZATION

There are many scalar quantization techniques that are available in the literature for image watermarking applications. An extensive theoretical study on the scalar quantization techniques for image watermarking applications was carried by Chen and Wornell [11]. Quantization Index Modulation (QIM), Quantization Index Modulus Modulation (QIMM), and Dither Modulation are variants of scalar quantization schemes.

Two methods in QIMM and one method in QIM are presented here. Host signal is $X = \{x_1, x_2, ... x_n\}$ The original watermark $W = \{w_1, w_2, ... w_n\}, w_i \in \{0,1\}$ is the binary message signal. The extracted watermark is denoted by $W' = \{w_1', w_2', ... w_n'\}, w_i' \in \{0,1\}$

Quantization Index Modulus Modulation:

Method I (QIMM1):

- In QIMM, the host signal $X = \{x_1, x_2, ... x_n\}$ is first divided by the quantization step size Δ. It is rounded to the nearest integer value by using

  z= $Q(x_i) = Round(x_i / \Delta)$

- If $\mathrm{mod}(Q(x_i,2) = w_i$

  $X' = Q(x_i') = z * \Delta$

  Else

$$X' = Q(x_i') = \min((z-1) * \Delta, x_i)$$

Watermark bit is detected as follows:

- Received X' is divided by the same quantization step size Δ
- The message is extracted as follows.

$$Q(x_i') = Round(x_i' / \Delta)$$
$$w_i' = Q(x_i') \bmod 2$$

Method II (QIMM-2):

- $X = \{x_1, x_2, \ldots x_n\}$
- If $w_i = 0$

$$v1 = \bmod(x_i, \Delta)$$

$x_i' = x_i - v1 + z'$, where

$$z' = \begin{cases} \Delta/8 & v1 < 3\Delta/8 \\ 5\Delta/8 & 3\Delta/8 \le v1 < 7\Delta/8 \\ 9\Delta/8 & 7\Delta/8 \le v1 < \Delta \end{cases}$$

- If $w_i = 1$

$$v1 = \bmod(x_i, \Delta)$$

$x_i' = x_i - v1 + z'$, where

$$z' = \begin{cases} -\Delta/8 & v1 < \Delta/8 \\ 3\Delta/8 & \Delta/8 \le v1 < 5\Delta/8 \\ 7\Delta/8 & 5\Delta/8 \le v1 < \Delta \end{cases}$$

Watermark bit is detected as follows:

$$w_i' = \begin{cases} 0 & 0 \le v1 < \Delta/4 \quad or \quad \Delta/2 \le v1 < 3\Delta/4 \\ 1 & \Delta/4 \le v1 < \Delta/2 \quad or \quad 3\Delta/4 \le v1 < \Delta \end{cases}$$

Quantization Index Modulation (QIM):

For the quantization of both D and S coefficients, dither quantization is used. Dither quantization is a variant of QIM [10]. Dither quantizers are quantizer ensembles. Each quantization cell in the ensemble is constructed from a basic quantizer. The basic quantizer may be chosen arbitrarily. The basic quantizer is shifted to get the reconstruction point. The shift depends on the watermark bit. The basic quantizer is a uniform scalar quantizer with a fixed step size. The quantized value is the center of the quantizer. Dither quantization of an image $h(i, j)$ is described as follows:

The entire range h_{min} (minimum value of $h(i, j)$) to h_{max} (maximum value of $h(i, j)$) is divided into various bins as shown in Table 1. A step size of T is taken as the difference from one bin to another bin. Each element of $h(i, j)$ is checked for its position in Table 1.

Table 1. Quantization Table

bin no. (n)	*dlow*	*dhigh*
1	$h_{min} - \Delta$	h_{min}
2	h_{min}	$h_{min} + \Delta$
3	$h_{min} + \Delta$	$h_{min} + 2\Delta$
. .	. .	. .
b_{n-1}	$h_{max} - \Delta$	h_{max}
b_n	h_{max}	$h_{max} + \Delta$

After identifying the bin number *n*, $h(i, j)$ is modified as follows:

(i) If watermark bit is '1' then it belongs to Range 1 where Range 1 is defined as

Range 1 = *dlow (n)* to $\dfrac{dlow(n) + dhigh(n)}{2}$

Modification of $h(i, j)$ is

$$h(i, j) = \left(\frac{(dlow(n) + (dlow(n) + dhigh(n))/2}{2} \right)$$

(ii) If watermark bit is '0' then it belongs to Range 2 where Range 2 is defined as

Range 2= $\dfrac{dlow(n) + dhigh(n)}{2}$ to *dhigh (n)*

Modification of $h(i, j)$ is

$$h(i, j) = \left(\frac{(dhigh(n) + (dlow(n) + dhigh(n))/2}{2} \right)$$

Similar quantization table is generated at the receiver and if the watermarked data X' is checked for its position in the table and accordingly the watermark is identified.

1(a) 1(b) 1(c)

Figure 1(a) Host Image LENA

(b) Watermark Image

(c) Watermarked Image

4. PROPOSED ALGORITHM

In the proposed work, both SVD and Schur decomposition are used for watermark embedding. In brief, the steps of watermark embedding and extraction algorithm are as follows.

I. SVD decomposition/ Schur decomposition is applied to the cover image on block wise.

II. Out of the three matrices available in SVD decomposition, D matrix is used for watermark embedding. Watermark is embedded in the largest coefficients

III. In Schur decomposition, two matrices are available for watermark embedding. S matrix is chosen for watermark embedding.

IV. The largest D element in each block is selected for watermark embedding in SVD decomposition.

V. The largest S element in each block is selected for watermark embedding in Schur decomposition.

VI. Matrices with the largest elements are formed and quantized using the method discussed in section 3.

VII. Watermark is extracted from the D and S matrices using appropriate step size as discussed in section 3.

5. EXPERIMENTAL RESULTS

Lena image of size 512x512 is considered as cover image. The watermark image is of size 32x32, which is a binary logo having the letters 'JNTU'. Host image, watermark image and watermarked image are shown in Figure 1(a), (b) and (c) respectively. Step size Δ selected for QIM is 60 and for QIMM-1 & QIMM-2 it is 30.

Various attacks used to test the robustness of the proposed watermarking algorithm are rotation, low pass filtering, median filtering, resizing, JPEG compression and salt & pepper noise. PSNR between the host image and watermarked image is more than 44 dB. The proposed algorithm is superior to Chung et al., [8] in terms of both PSNR and NCC. The comparison is shown in Table 2.

Table 2. Comparison of the proposed algorithm with Chung et al., [8]

Parameter	Chung et al., [8]	Proposed
		Algorithm
Transform	SVD	SVD and Schur Decomposition
Cover image Size	512x512	512x512
Watermark image size	32x32	32x32
PSNR between cover image and watermarked image	38.69 dB	More than 44 dB

Both SVD and Schur decomposition are explored for watermark embedding. The results are listed in Table 3. The performance of both the decompositions is almost similar. In particular, robustness of the Schur based QIM algorithm is superior compared to SVD based QIM algorithm for majority of attacks. Also, Schur decomposition based watermarking scheme is faster compared to SVD decomposition. However, PSNR in SVD based watermarking algorithm is slightly higher compared to Schur based algorithm. .

6. CONCLUSIONS

In this work, the performance of both SVD and Schur decompositions for image watermarking applications is investigated. Scalar quantization schemes (Quantization Index Modulation and Quantization Index Modulus Modulation) are explored for watermark embedding. It is observed that, the performance both the algorithms is comparable. The proposed algorithms are superior compared to Chung et al., [8] both in terms of PSNR and NCC. Schur decomposition is computationally faster compared to SVD decomposition.

7. ACKNOWLEDGMENTS

First author would like to extend his sincere thanks to the Principal and the Management of Bapatla Engineering College, Bapatla for the support extended in the completion of the work.

8. REFERENCES

[1] Chu, W.C., 2003. DCT based image watermarking using sub sampling, IEEE Transactions Multimedia, pp.34-38.

[2] Wang, Y., Alan Pearmain., 2004. Blind image data hiding based on self reference. Pattern Recognition Letters 25, pp. 1681-1689.

[3] Barni, M., Bartolini, M., Piva, F.V., 2001. Improved wavelet based watermarking through pixel-wise masking. IEEE Transactions on Image Processing 10, pp. 783-791.

[4] Lee, C., Lee, H., 2005. Geometric attack resistant watermarking in wavelet transform domain, Optics Express vol.13, no.4, pp.1307-1321.

[5] Chandra Mohan B, Srinivas Kumar S, Chatterjee B.N.,2006. Digital Image watermarking in dual domains at IET Visual Information Engineering, VIE 2006 at Leela Palace, Bangalore, 26-28th September 2006.

[6] Srinivas Kumar S, Chandra Mohan B, Chatterjee B.N., 2007. An oblivious image watermarking scheme using

singular value decomposition. IASTED, International conference on Signal and Image Processing, at Honolulu, Hawaii, USA, August 20-22.

[7] Yongdong, Wu, 2005. On the Security of an SVD based ownership watermarking. IEEE transactions on Multimedia, vol 7. no.4.

[8] K.L.Chung, W.N.Yang, Y.H.Huang, S.T.Wu, and Y.C.Hsu, "On SVD-based watermarking algorithm", Applied Mathematics and Computation, Vol.188, 2007, pp.54-57

[9] B.Chandra Mohan, S.S. Kumar, "Robust digital watermarking scheme using contourlet transform", International Journal of Computer Science and Network Security (IJCSNS), Vol.8, Issue 2, 2008, pp.43-51.

[10] Chandramouli,R., Graubard Benjamin, M., Richmond Colin, R., 2001. A multiple description framework for oblivious watermarking. Proceeding of Security and Watermarking and Multimedia contents III, SPIE vol. 4314.

[11] Chen, B., Wornell, G.W., 1998. Digital watermarking and information embedding using dither modulation. Proceedings of the IEEE workshop on Multimedia Signal Processing (MMSP-98), Redondo Beach, CA, December.

Table 3. Comparison of SVD and Schur Decompositions

	SCHUR DECOMPOSITION			SVD DECOMPOSITION		
Attack	QIMM-1	QIMM-2	QIM	QIMM-1	QIMM-2	QIM
No attack NCC PSNR (dB)	1.000 44.7933	1.0000 45.1797	1.0000 45.1905	1.0000 44.6964	1.0000 45.0842	1.0000 45.2279
Rotation	0.6407 15.3408	0.6598 15.3414	0.9013 15.3530	0.6527 15.3407	0.6580 15.3414	0.8836 15.3531
LPF	0.6460 32.0270	0.6460 32.1030	0.7170 32.0346	0.6831 32.0086	0.6891 32.1013	0.6409 32.0337
Median Filtering	0.8371 35.6718	0.8440 35.8065	0.7283 35.7032	0.8423 35.6268	0.8270 35.7781	0.6599 35.6992
Resizing	0.9953 34.4532	0.9860 34.4896	1.0000 34.4445	0.9953 34.4325	0.9817 34.4803	0.9721 34.4428
JPEG QF=60	0.8455 35.6725	0.8350 36.3454	0.9637 36.3676	0.8734 35.6683	0.8999 36.3293	0.9680 36.3624
Salt & Pepper Noise.1%	0.8188 35.2626	0.8218 34.6557	0.8579 34.7822	0.8240 35.1984	0.7823 34.7724	0.8298 34.6600

Average Embedding Time in seconds	0.355590	0.356347	0.719695	0.434772	0.436829	0.804330
Average Extraction time in seconds	0.240285	0.239323	0.242631	0.317159	0.317372	0.316882