Digital Watermarking of Color Images in the Singular Domain

Rashmi Agarwal
Department of Computer Science,
Mohanlal Sukhadia University,
Udaipur, India-313 039.

K. Venugopalan
Department of Physics,
Mohanlal Sukhadia University,
Udaipur, India-313 039.

ABSTRACT

A watermarking scheme suitable for color images is given. The algorithm is based on the well-known matrix factorization technique of the singular vector decomposition. This is achieved by using the fact that a color image can be broken into the RGB channels. These channels are then treated separately as matrices on which the matrix factorization is applied. These matrices are then used to present the embedding and extraction algorithms. We also study how our watermarking technique performs under various attacks.

Keywords

Watermarking, Color Images, Singular Value Decomposition.

1. INTRODUCTION

A Watermark is a signal or image embedded into an another image called the host or the carrier signal/image. It can be done for various reasons and the primary one being the protection of digital content against theft. Thus, it is used for copyright and ownership protection. One can also resort to watermarking as a way to hide some information. In this work we will be primarily concerned with the digital watermarking of color images. In the case of images watermarking means embedding one image into another resulting in what is called the watermarked image (WI). The image so placed in the host can be done in such a way that it may be visible or invisible. But just this does not complete the task. One still has to successfully extract the embedded image (EI) from the WI. Only when both the embedding and extraction are satisfactorily done we can say that watermarking procedure is complete.

Broadly watermarking techniques can be classified into spatial [1], [2], [3], [4], [5], [6], [7] and frequency [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] based. Watermarking algorithms that rely on spatial domains, hide the watermark by modifying the pixel values of the host image. On the other hand in frequency based techniques the host image is first converted into different frequencies by a suitable transform. The transformed domain coefficients are then modified by the watermark. To recover the watermark the inverse transform is finally applied on the WI. Some of the popular transform techniques that have been used are the discrete cosine transform (DCT) [10], [16], [8], [14], [9], [12], the discrete Fourier transform (DFT), and the discrete wavelet transform (DWT) [17], [11]. Recently we have given a new watermarking algorithm based on the Hilbert transform [19].

Apart from the above mentioned techniques another watermarking method that has gained quite a bit of popularity is based on the singular vector domain of images [20], [21], [22], [23], [24], [25], [26]. These watermarking algorithms are based on the well known matrix factorization technique of the singular value decomposition (SVD) [27]. We will have more to say about this since our watermarking process is SVD based. Sometimes to improve the watermarking procedure some authors have developed hybrid algorithms [28], [29], [30].

The layout of the manuscript is as follows. In the next section we will present our embedding and extraction algorithm. This is then followed by section 3 wherein, we will explicitly apply the watermarking and the extraction algorithm to a color image. To test the robustness of our algorithm, in section 4 the WI obtained in the previous section is subjected to a number of attacks the results of which have been collected in a table for the sake of convenience. The conclusions of our work are presented in section 5.

2. SVD-BASED WATERMARKING

Any watermarking procedure can be divided into two parts with distinct functions. One is the embedding algorithm, wherein the image that needs to be hidden is placed into another image called the host or the carrier image. The extraction algorithm constitutes part two of the whole procedure, in which the image that was embedded in the host is removed so that we obtain the host and the watermark separately. Our watermarking procedure makes use of the fact that we can treat the images, host as well as the watermark, as a matrix. The elements of this matrix are given by the various pixel values. Now that the image has been converted to a matrix, we employ the well-known matrix factorization technique of SVD to the host and the watermark matrices. This splits a single matrix into product of three matrices. Further details about this factorization can be found in [27]. These three matrices, of the host and the watermark, are then used as the raw material for the construction of embedding and the extraction algorithm which we now describe.

It must be mentioned that the original algorithm was presented, in an earlier work of the first author and her collaborator [24], in the context of gray scale images. The case for the color images was just mentioned in passing. The present study begins where the earlier work left off, in that it takes up the watermarking and robustness tests of color images exclusively. Furthermore this manuscript also demonstrates the applicability of our algorithm to rectangular images.

2.1 The Embedding Algorithm

Before we explicitly derive the algorithm a few things need to told. The algorithm is the same as that used for gray images [24]. But there is one important difference that one must bear in mind and that is color images, the host and the watermark, can be broken into its red, green, and blue channels. Here of course we are restricting ourselves to the RGB color space. Though there are other color spaces but the RGB split is a much more well-known space. Each channel is then treated as a separate image on which the embedding and extraction process is applied separately as if it were a gray image. Thus, the red channel of watermark is embedded in the red channel of the host, similarly the process is repeated for the blue and green channels as well. Finally, these three channels are combined to yield the watermarked color image.

Let H and W be the matrices representing the host and the watermark. As mentioned above, in fact the host/watermark images actually consists of three matrices H_R /W_R, H_G/W_G, H_B /W_B for the red, green, and blue channels respectively. The embedding algorithm is presented for any one arbitrary channel therefore, for the sake of convenience we have dropped the channel label. As a first step, we compute the SVD of host

$$H=U_H D_H V_H^T = A_H V_H^T, \tag{1}$$

and the watermark image

$$W=U_W D_W V_W^T = A_W V_W^T, \tag{2}$$

where $A_{H/W} = U_{H/W} \ D_{H/W}$ are also called the principal components in the language of principal component analysis. Now, we add the scaled eigenvector V_W of watermark to that of the original image,

$$V=V_H+\lambda \ V_W \tag{3}$$

Here λ is the scaling factor. Typically, $0 \leq \lambda \leq 1$, smaller the value of λ lesser is the distortion of the host. As $\lambda \rightarrow 0$, the approximation that V is a orthogonal matrix, i.e., $V V^T \approx I$ gets better. This property is important in the next step for constructing the watermarked image:

$$H_c= A_H V^T \tag{4}$$

Therefore the choice of lambda is crucial in the algorithm varying which we can get transparent as well as visible watermarking. That is if one needs a visible watermark the value of λ is taken to be large and for invisible it is taken to be small. Equations (1-4) constitute the algorithm for watermarking using SVD in eigenvector domain.

2.2 The Extraction Algorithm

In the previous subsection we have shown how to embed a watermark, in the present we will explain how the watermark can be regained. Our starting point will be the watermarked matrix, H_c , that has been obtained after adding the watermark. Once again let us point out that this watermarked image can be split into the red, green and blue channels. The extraction algorithm, given below, will then be applied to each individual channel. The three matrices thereby obtained are combined to yield the color watermark.

This extraction algorithm relies on the fact that we have access to A_H , A_W , V_H and also the value of λ . The recovery of the watermark is a straightforward reversal of the embedding algorithm. Multiplying both sides of Eq. (4) by A_H^{-1} and substituting the expression of V^T from Eq. (3), it is easy to obtain $V_W{}^T$:

$$V_W^T = \frac{A_H^{-1} H_c - V_H^T}{\lambda} \tag{5}$$

Finally, using Eq. (2), the watermark image can be constructed as,

$$\widetilde{W} = A_W V_W^T. \tag{6}$$

Eq. (6), along with Eq. (5) constitutes the watermark extraction algorithm.

3. IMPLEMENTATION OF THE ALGORITHM

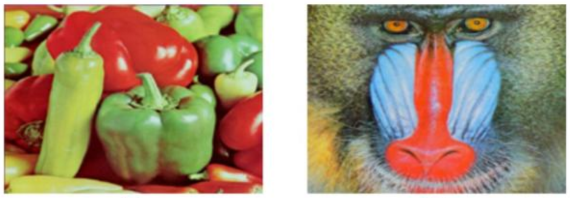
In the previous section we have developed an algorithm for watermarking of color images of arbitrary size. In the present section we will give the results of the implementation of the above algorithm. As the host we take Fig. 1(a) and the watermark is given in Fig. 1(b). Notice that the images chosen are rectangular and are of size 200 × 150. The various figures in (2) show the results of embedding and extraction for different values of lambda. One might wonder what value of λ must be chosen. From the various images of peppers one can notice that even after the watermark has been inserted the visual quality of the host is still quite good. Only at $\lambda = 0.4$, Fig. 2(p), the watermarked image starts showing deterioration. This conclusion is also borne out from the peak signal to noise ratio-root mean square error (PSNR-RMSE) values given below the various figures, which shows a highest value of RMSE and consequently the lowest PSNR value at $\lambda = 0.4$. Hence we will choose a lambda value that is definitely below 0.4. Now we look at the various PSNR-RMSE values of the extracted baboon images. It can be noticed that these values happen to be the best for Fig. 2(h). Thus we choose the $\lambda = 0.08$ for the rest of the manuscript. The correctness of this choice is also clear from the PSNR and RMSE graphs depicted in figures 3(a) and 3(b) respectively. The RMSE formula for color images

$$\varepsilon = \sqrt{\frac{1}{3mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k} [X_k(i,j) - x_k^*(i,j)]^2} \qquad (7)$$

where sum over the index k is over the various channels namely the red, green, and blue. $X_k(i, j)$ is the pixel value at location (i, j) of the original image and $x_r^*(i, j)$ is value of the watermarked image with both the images being of size $m \times n$. The PSNR, measured in decibels, is given by

$$p = 10 \log_{10} (\max z_{i,j}^2 / \varepsilon^2) \qquad (8)$$

where max $z_{i,j}$ represents the maximum value of a matrix whose elements are $z_{i,j}$.



(a) (b)

Figure 1. (a) Host image and (b) the watermark.

4. ROBUSTNESS OF THE ALGORITHM

We have mentioned in the introduction that one of the primary reasons for digital watermarking is copyright protection and ownership assertion. Just because an image has been watermarked does not imply that the content is safe. Many times some simple operations can corrupt the watermarked image in such a way that it may become difficult to extract the watermark. These operations can then be intentionally performed and the digital media be stolen. To prevent this, the watermarking procedure, specifically the extraction process, should be resilient under such attacks. The attacks primarily can be categorized as forming two groups. One that affect the pixel values and the other that affect the geometry of the image. Operations such as noise addition, JPEG and JPEG 2000 compression, and gamma correction are some examples belonging to group one. The geometric transformations are scaling, cropping, and rotation. In the present section we study the results of a number of attacks, specifically all attacks are performed on Fig. (2(g)) for which the scaling factor is 0.08. The visual results of various attacks are given in Fig. 4. The captions mention the specific attack that has been carried out. Since these attacks are well-known we will not describe them further. Let us also mention that the full list of attacks with the corresponding PSNR-RMSE values of the WI and the EI are listed in Table. (1). We can see both from Fig. (2(g)) and the table (1) that our watermarking procedure performs quite satisfactorily.

5. CONCLUSIONS

In the present work we have utilized a SVD based watermarking algorithm, developed in our earlier work for gray images, to the case of color images. We have made use of the fact that a color image can be split up into three separate channels and the watermarking algorithm can then be applied on each channel separately. These channels were later combined to yield the watermarked color image. Similarly the extraction process is carried on the three channels separately.

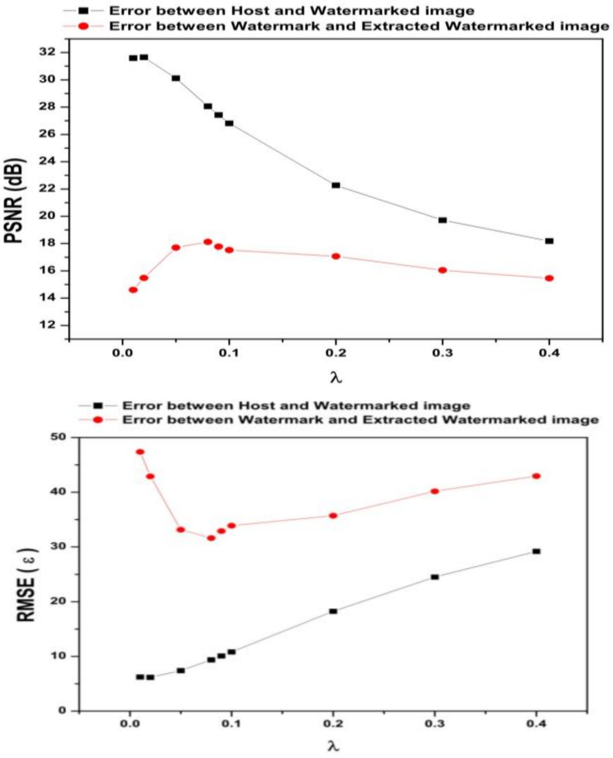


Figure 3. (a) PSNR plot for different values of λ of the watermarked and the extracted images. (b) RMSE plot for different values of λ of the watermarked and the extracted images.

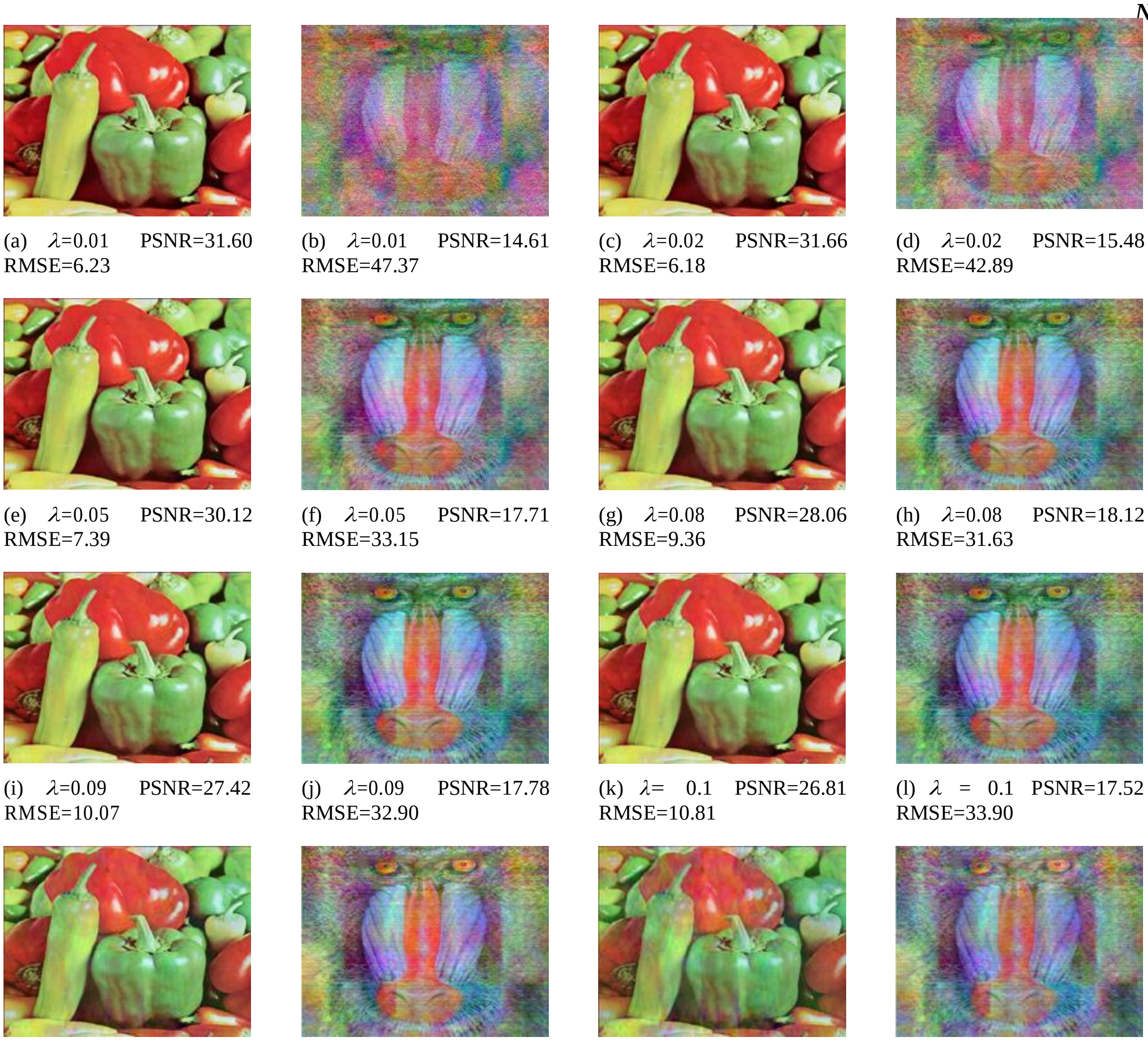
We have also subjected our algorithm to various attacks and have shown that it performs quite well. All the experiments have been shown in Table. (1). Finally we would like to once again mention that this manuscript has appropriately shown that the SVD based watermarking technique presented by us is quite good for rectangular color images of an arbitrary size.

REFERENCES

[1] Lee, Y. K. and Chen, L. H. 2000. High capacity image steganographic model. IEE Proceedings on Vision, Image and Signal Processing/ Vol. 147, no. 3, pp. 288-294.

[2] Ren-Junn H., Chuan-Ho, K., and Rong-Chi, C. 2002. Watermark in color image. Proceedings of the first International Symposium on Cyber Worlds. pp. 225-229.

[3] Kimpan,S., Lasakul, A., and Chitwong, S. 2004. Variable block size based adaptive watermarking in spatial domain. IEEE International Symposium on Communications and Information Technology, ISCIT 2004. Vol. 1, pp. 374-377.



(a) λ=0.01 PSNR=31.60 RMSE=6.23

(b) λ=0.01 PSNR=14.61 RMSE=47.37

(c) λ=0.02 PSNR=31.66 RMSE=6.18

(d) λ=0.02 PSNR=15.48 RMSE=42.89

(e) λ=0.05 PSNR=30.12 RMSE=7.39

(f) λ=0.05 PSNR=17.71 RMSE=33.15

(g) λ=0.08 PSNR=28.06 RMSE=9.36

(h) λ=0.08 PSNR=18.12 RMSE=31.63

(i) λ=0.09 PSNR=27.42 RMSE=10.07

(j) λ=0.09 PSNR=17.78 RMSE=32.90

(k) λ= 0.1 PSNR=26.81 RMSE=10.81

(l) λ = 0.1 PSNR=17.52 RMSE=33.90

(m) λ=0.2 PSNR=22.27 RMSE=18.23

(n) λ= 0.2 PSNR=17.07 RMSE=35.71

(o) λ= 0.4 PSNR=18.19 RMSE=29.18

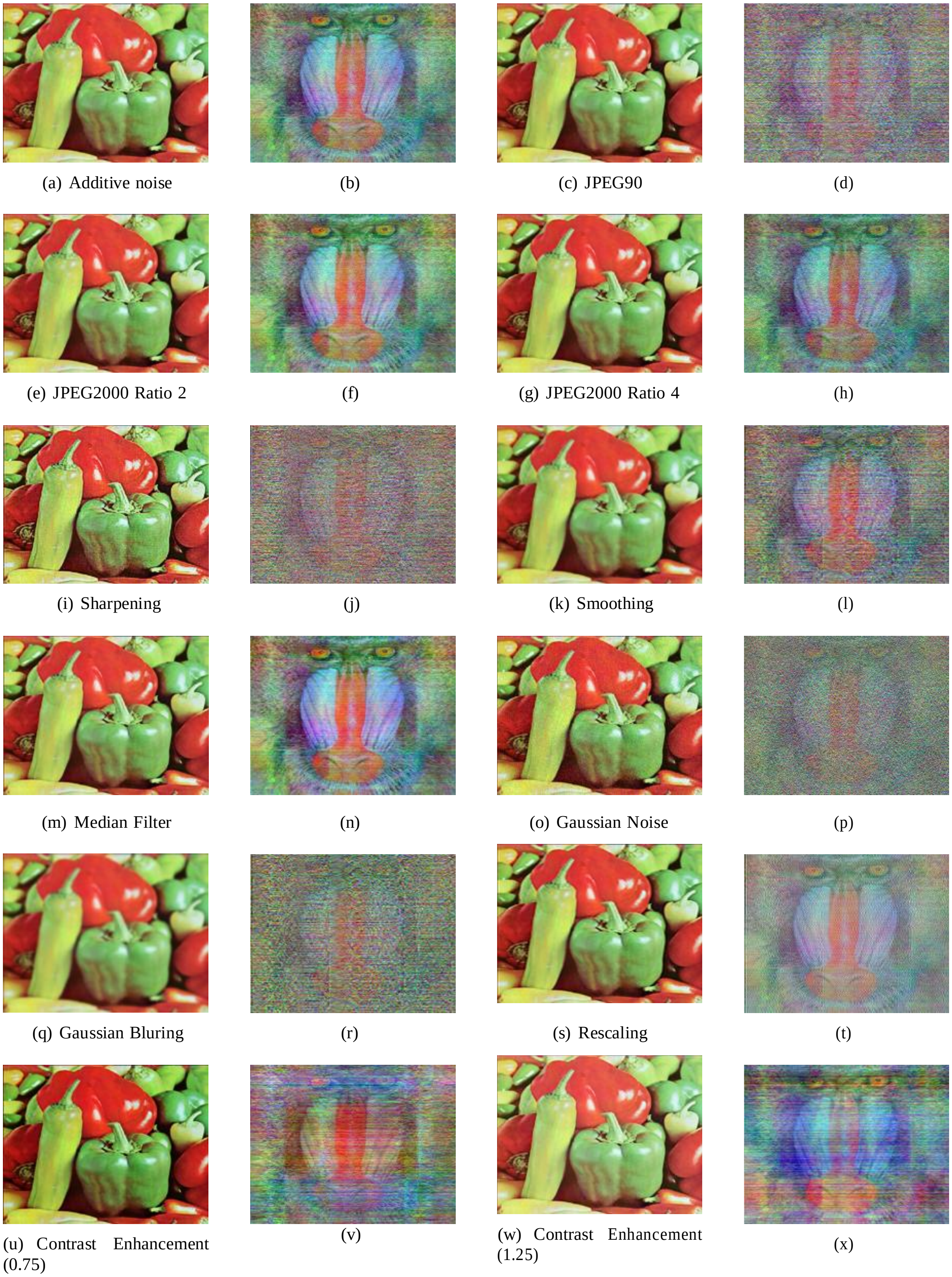
(p) λ= 0.4 PSNR=15.46 RMSE=42.96

Figure 2. Watermarked images (Pepper) using different λ values and Extracted Watermarks (Baboon).

Table 1. A complete list of different types of attacks performed on the watermarked image. The PSNR (dB) and RMSE (ε) values of the watermarked (Papper) as well as the extracted (Baboon) images are given

TABLE I

S. No.	Type of Attack	Software Used	Pepper		Baboon	
			PSNR	RMSE	PSNR	RMSE
0.	No Attack		28.06	9.36	18.12	31.63
1.	Additive Noise	MATLAB R2010a	27.77	9.68	17.18	35.27
2.	JPEG 20%	MATLAB R2010a	22.88	17.00	13.17	55.94
3.	JPEG 40%	MATLAB R2010a	23.88	15.15	13.34	54.89
4.	JPEG 60%	MATLAB R2010a	24.47	14.15	13.18	55.88
5.	JPEG 80%	MATLAB R2010a	25.36	12.78	13.79	52.07
6.	JPEG 90%	MATLAB R2010a	26.09	11.74	13.85	51.71
7.	JPEG2000 (Compression Ratio 2)	MATLAB R2010a	28.00	9.42	17.33	34.66
8.	JPEG2000 (Compression Ratio 4)	MATLAB R2010a	27.90	9.54	15.81	41.28
9.	JPEG2000 (Compression Ratio 8)	MATLAB R2010a	27.40	10.10	14.75	46.63
10.	JPEG2000 (Compression Ratio 16)	MATLAB R2010a	26.18	11.63	13.60	53.22
11.	Sharpening	Corel Graphics Suite 11	24.56	14.01	13.84	51.79
12	Smoothing	Corel Graphics Suite 11	27.25	10.27	15.34	43.59
13.	Histogram equalization	Corel Graphics Suite 11	19.07	26.36	12.88	57.82
14.	Gaussian Noise	Corel Graphics Suite 11	23.14	16.50	12.88	57.83
15.	Gaussian Bluring	Corel Graphics Suite 11	24.61	13.92	13.92	51.32
16.	Scaling	XnView	27.61	9.86	14.96	45.53
17.	Contrast Enhancement (0.75)	Corel Graphics Suite 11	23.12	16.53	13.62	53.09
18.	Contrast Enhancement (1.25)	Corel Graphics Suite 11	19.94	23.85	14.69	46.96
19.	Median Filter (3 ×3)	MATLAB R2010a	24.42	14.23	13.76	52.26



(a) Additive noise

(b)

(c) JPEG90

(d)

(e) JPEG2000 Ratio 2

(f)

(g) JPEG2000 Ratio 4

(h)

(i) Sharpening

(j)

(k) Smoothing

(l)

(m) Median Filter

(n)

(o) Gaussian Noise

(p)

(q) Gaussian Bluring

(r)

(s) Rescaling

(t)

(u) Contrast Enhancement (0.75)

(v)

(w) Contrast Enhancement (1.25)

(x)

Figure 4. Panel depicting the various attacked watermarked images (Pepper) and the corresponding extracted images (Baboon) from the same.

[4] Huang, P. S., Chiang, C. S., Chang, C. P., and Tu, T. M. 2005. Robust spatial watermarking technique for colour images via direct saturation adjustment. IEE Proceedings. of Vision, Image and Signal Processing. Vol. 152, no. 5, pp. 561-574.

[5] Verma, B., Jain, S., Agarwal, D. P., and Phadikar, A. 2006. A New color image watermarking scheme. INFOCOMP Journal of computer science. Vol. 5, no. 3, pp. 37-42.

[6] Wu, X., and Guan, Z.-H. 2007. A novel digital watermark algorithm based on chaotic maps. Physics Letters A. Vol. 365, pp. 403-406.

[7] Nasir, I., Weng, Y., Jiang, J., and Ipson, S. 2010. Multiple spatial watermarking technique in color images. Signal, Image and Video Processing. Vol. 4, no. 2, pp. 145-154.

[8] Bros, A. G., and Pitas, I. 1998. Image watermarking using block site selection and DCT domain constraints. Optics Express. Vol. 3, no. 12, pp. 512-522.

[9] Cox, I. J., Kilian, J., Leighton, F. T., and Shamoon, T. 1997. Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing., Vol. 6, no. 12, pp. 1673-1687.

[10] Barni, M., Bartolini, F., Cappellini, V., and Piva, A. 1998. A DCT domain system for robust image watermarking. IEEE Transactions on Signal Processing., Vol. 66, no. 3, pp. 357-372.

[11] Xie, L., and Arce, G. 1998. Joint wavelet compression and authentication watermarking. in IEEE International Conference on Image Processing. Vol. 2, pp. 427-431.

[12] Langelaar, R. L. G., and Biemond, J., 1999. Watermarking by DCT coef- ficient removal: Statistical approach to optimal parameter settings, Proc. SPIE IS&T/SPIE 11th Annu. Symp on Electronic Imaging: Security and Watermarking of Multimedia Contents. Vol. 3657, pp. 2-13.

[13] Lu, C. S., Huang, S. K., Sze, C. J., Yuan, H., and Liao, M. 2000. Cocktail watermarking for digital image protection. IEEE Transactions on Multimedia. Vol. 2, no. 4, pp. 209-224.

[14] Barni, M., Bartolini, F., and Piva, A. 2002. Multichannel watermarking of color images. Proc. IEEE Trans. Circuit Syst. Video Technol. Vol. 12, no. 3, pp. 142-156.

[15] Jiang, G., Yu, M., Shi, S., Liu, X., and Kim, Y. D. 2002. New blind image watermarking in DCT domain. in Proceedings of the 6th International Conference on Signal Processing. Vol. 2, pp. 1580-1583.

[16] Chu, W. C., 2003. DCT based image watermarking using subsampling. IEEE Trans Multimedia. Vol. 5, no. 1, pp. 34-38.

[17] Raval, M. S., and Rege, P. P., 2003. Discrete wavelet transform based multiple watermarking scheme. Int. Conference on Convergent Tech- nologies for Asia-Pacific Region. Vol. 3, pp. 935-938.

[18] Reddy, A. A., and Chatterji, B. N. 2005. A new wavelet based logo- watermarking scheme. Pattern Recognition Letters. Vol. 26, no. 7, pp. 1019-1027.

[19] Agarwal, R., Krishnan, R., Santhanam, M. S., Srinivas, K., and Venu- gopalan, K. 2010. Digital Watermarking: An approach based in Hilbert transform. arXiv:1012.2965.

[20] Gorodetski, V. I., Popyack, L. J., Samoilov, V., and Skormin, V. A. 2001. SVD-Based Approach to Transparent Embedding Data into Digital Images. International workshop on Mathematical methods, Models and Architectures for Computer Network security. Vol. 2052, pp. 263-274.

[21] Liu, R. and Tan, T. 2002. A SVD-based watermarking scheme for protecting rightful ownership. IEEE Transactions on Multimedia. Vol. 4, no. 1, pp. 121-128.

[22] Ganic, E., Zubair, N., and Eskicioglu, A. M., 2003. An Optimal Watermarking Scheme Based On Singular Value Decomposition. in Pro- ceedings of the IASTED International Conference on Communication, Network, and Information Security, pp. 85-90.

[23] Chang, C. C., Tsai, P., and Lin, C. C. 2005. SVD-based digital image watermarking scheme. Pattern Recognition Letters. Vol. 26, no. 10, pp. 1577-1586.

[24] Agarwal, R., and Santhanam, M. S. 2008. Digital watermarking in the singular vector domain. International Journal of Image and Graphics. Vol. 8, no. 3, pp. 351-368.

[25] Jain, C., Arora, S., and Panigrahi, P. K., 2008. A reliable SVD based watermarking scheme. arXiv:0808.0309v1.

[26] Deng, J. 2009. Color Image Digital Watermarking Algorithm Based on Singular Value Decomposition. International Conference on Multimedia Information Networking and Security, MINES'09. Vol. 2, pp. 130-133.

[27] Golub, G. H., and Loan, C. F. V. 1996. Matrix computations. Johns Hopkins university Press.

[28] Fei, C., Kundur, D., and Kwong, R. H. 2004. Analysis and Design of Watermarking Algorithms for Improved Resistance to Compression. IEEE Transactions on Image Processing. Vol. 13, no. 2, pp. 126-144.

[29] Zhao, Y., Campisi, P., and Kundur, D. 2004. Dual domain watermarking for authentication and compression of cultural heritage image," IEEE Transactions on Image Processing. Vol. 13, no. 3, pp. 430-448.

[30] Ganic, E., and Eskicioglu, A. M. 2005. Robust embedding of visual watermarks using discrete wavelet transform and singular value decom- position. J. Electron. Imaging., Vol. 14, pp. 043004.