Hybrid Color Image Watermarking Algorithm Based on DSWT-DCT-SVD and Arnold Transform

Palak Garg, Lakshita Dodeja, Priyanka and Mayank Dave

Abstract With emergence of new technologies it is now easier to communicate through multimedia like image, audio, video and text. But at the same time the problem of unauthorized access and copyright protection has also emerged. In order to handle these problems digital image watermarking is one of the best technique. In this paper we present an optimized color image watermarking technique to protect an image data from any unauthorized access. The technique presented in the paper uses a combination of Discrete Stationary Wavelet Transform (DSWT), Singular Value Decomposition (SVD), Discrete Cosine Transform (DCT) and Arnold Transform. In this technique we hide a color image watermark into a colored cover image without hampering the perceptibility of the cover image. Peak-signal-to-noise-ratio (PSNR) and Normalized Correlation (NC) are used to analyze the proposed watermarking technique for the imperceptibility and robustness measures.

P. Garg · L. Dodeja (✉) · Priyanka · M. Dave
Department of Computer Engineering, National Institute of Technology,
Kurukshetra, Kurukshetra, India
e-mail: lakshita.dodeja@gmail.com

P. Garg
e-mail: palak.garg2107@gmail.com

Priyanka
e-mail: pdogra66@gmail.com

M. Dave
e-mail: mdave@nitkkr.ac.in

327

1 Introduction

The ease of sharing digital media like audio, video, images and text with the rising information technology has also increased the demand for security, copy control, copyright authentication etc. of the information in digital form. Watermarking is one such technique to tackle with these issues along with others. Digital image watermarking hides ownership data known as watermark into the multimedia data that can be extracted when required to provide the proof of authenticity [1]. The watermark either visible or invisible does not hamper the original use of the multimedia content. Watermark can be inserted either by modifying the pixel values of the image known as spatial domain technique which is computationally efficient but low in robustness and imperceptibility or transform domain techniques that are more popular nowadays because they embed the watermark into the coefficients of transform domain [2]. Discrete cosine transform (DCT), discrete stationary wavelet transform (DSWT) and singular value decomposition (SVD) are popular transformation techniques. Each of these is highly efficient against some of the attacks but not all. The proposed algorithm withstands majority of the attacks by combining various techniques. Robustness, imperceptibility and capacity are the three requirements of any watermarking scheme [3]. In this algorithm, discrete stationary wavelet transform (DSWT) provides protection against geometric attacks and robustness, singular value decomposition (SVD) provides imperceptibility, discrete cosine transform (DCT) provides protection against compression and Arnold transform provides watermark security. Using DSWT also enables us to keep the same size of watermark as the cover and hence the capacity of information to be hidden remains the same.

2 Literature Review

2.1 Discrete Stationary Wavelet Transform (DSWT)

Discrete Wavelet Transform (DWT) transforms an image using wavelets having varying frequency and short time interval. It has an advantage over other transforms as it is able to capture location information in both time and frequency. DWT performs down sampling of the signal after passing it through high pass and low pass filters to give the approximation and detail coefficients.

Discrete Stationary Wavelet Transform (DSWT) is similar to DWT [4] but it does not involve down sampling of signal after decomposing it by applying high pass and low pass filters. This attributes DSWT with the additional property of translation invariance

Fig. 1 Separation of an image into 3 frequency regions by DCT
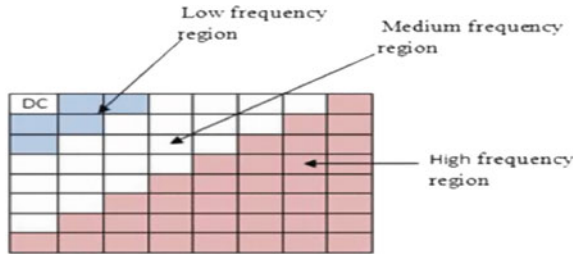
2.2 Discrete Cosine Transform (DCT)

Discrete Cosine Transform is the most popular transform in spectral domain and transforms data points into sum of cosine functions oscillating at different frequencies that can be grouped into three categories low, medium and high frequency. Embedding the image into middle frequencies provide protection against JPEG compression as only high frequencies are lost in the lossy JPEG compression and our eyes being susceptible to low frequency using middle frequency provides imperceptibility (Fig. 1).

2.3 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) is an arithmetical method that is used to covert a matrix into three diagonal matrixes [6]. In image processing an image can be seen as a matrix with non-negative scalar values. The SVD of an image A of size M x N is defined as

$$A = USV^T. \tag{1}$$

where U and V are orthogonal matrices that means multiplication of U and V with their transpose results in identity matrix, $UTU = I$, $VTV = I$, and $S = diag(\lambda 1, \lambda 2, \lambda 3, \ldots, \lambda r)$. The diagonal entries of S depict the luminance of A where r is the rank of A and they are known as singular values of A. The columns of U and V are used to retain the geometrical properties of the image and they are called the left and right singular vectors of A. US are called the principal components of A. UV together called as the SVD subspace of A. SVD is beneficial for watermarking as modification in singular values of images do not disturb the perceptibility of the cover image.

Fig. 2 **a** Original image **b**
Scrambled image using
Arnold transform
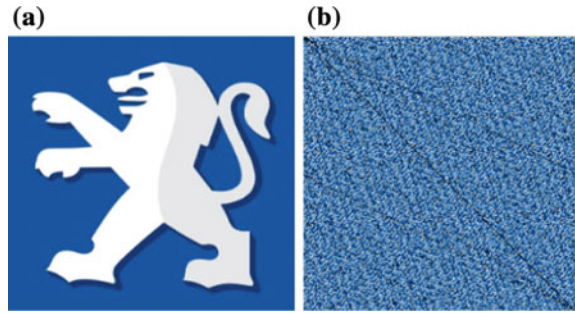
2.4 Arnold Transform

This transform is a splicing and clipping process that realigns the pixel matrix of
digital image. It is applied to the watermark before embedding it in host image. A
two dimensional Arnold transform [6] is shown as follows

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} mod\ N \qquad (2)$$

where x and y are the coordinates of the original image, N is the height or width of
the image and x' and y' are the coordinate of the scrambled image. The transform
changes the position of pixels, and a scrambled image can be obtained by applying
the algorithm several times. The disordered image can be converted back to the
original image after certain number of permutations due the periodicity of the Arnold
transform. For a 256×256 image, this period is 192, after which the scrambled image
becomes equal to the original image. Due to scrambling the pixels of image get evenly
distributed which helps in enhancing the robustness of algorithm (Fig. 2).

3 Proposed Algorithm

The embedding process is as follows-

Step 1: Read the cover and watermark each of 'N x N' pixels in RGB format.
Step 2: Split both the images in three channels, namely Red, Green and Blue.
Step 3: Apply Arnold transform on the watermark with the secret key 10 and 1-level
DSWT on the cover image.
Step 4: Apply DCT on the scrambled watermark and LL sub-band of 1-level DSWT.
Step 5: Apply SVD to the DCT coefficients of both the images.
Step 6: Embed the watermark.
Step 7: Apply Inverse SVD.
Step 8: Apply Inverse DCT.

Step 9: Apply Inverse DSWT that gives the watermarked image in one channel.
Step 10: Repeat steps 3–9 for the other two channels.
Step 11: Merge all the three channels to get the watermarked image (Fig. 3).

The extraction process is as follows-

Step 1: Read the cover, watermark and watermarked images in RGB format.
Step 2: Split all the images in three channels, namely Red, Green and Blue.
Step 3: Apply Arnold transform to the watermark and 1-level DSWT on the cover and watermarked images.
Step 4: Apply DCT on the scrambled watermark and LL sub-band of cover and watermarked images.
Step 5: Apply SVD on the DCT coefficients.
Step 6: Extract the watermark.
Step 7: Apply Inverse SVD.
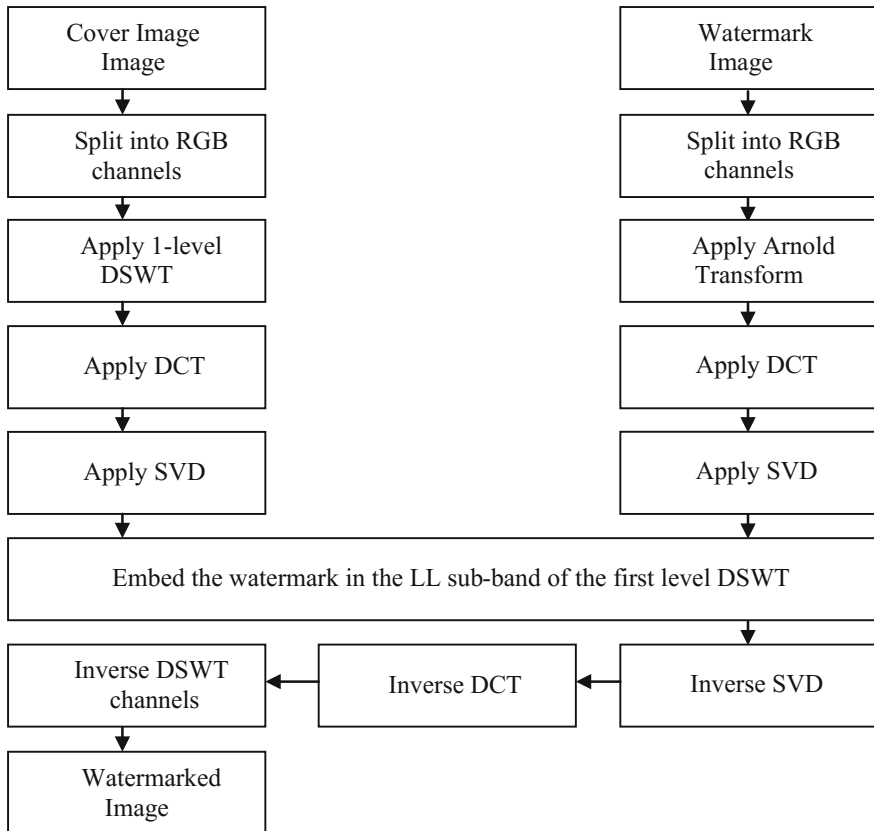*Step 8*: Apply Inverse DCT to get the scrambled watermark back.

Fig. 3 Embedding process

```
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│    Watermarked      │    │     Cover Image     │    │   Watermark Image   │
│       Image         │    │                     │    │                     │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │                          │                          │
           ▼                          ▼                          ▼
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│   Split into RGB    │    │   Split into RGB    │    │   Split into RGB    │
│     channels        │    │     channels        │    │     channels        │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │                          │                          │
           ▼                          ▼                          ▼
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│   Apply 1-level     │    │   Apply 1-level     │    │   Apply 1-level     │
│       DSWT          │    │       DSWT          │    │       DSWT          │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │                          │                          │
           ▼                          ▼                          ▼
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│     Apply DCT       │    │     Apply DCT       │    │     Apply DCT       │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │                          │                          │
           ▼                          ▼                          ▼
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│     Apply SVD       │    │     Apply SVD       │    │     Apply SVD       │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │                          │                          │
           ▼                          ▼                          ▼
┌─────────────────────────────────────────────────┐       ┌─────────────────────┐
│  Extract the watermark from the LL sub band      │ ────▶ │    Inverse SVD      │
└─────────────────────────────────────────────────┘       └─────────────────────┘
                                                                      │
                                                                      ▼
┌─────────────────────┐    ┌─────────────────────┐    ┌─────────────────────┐
│   Merge the RGB     │ ◀──│   Inverse Arnold    │ ◀──│    Inverse DCT      │
│     channels        │    │     Transform       │    │                     │
└─────────────────────┘    └─────────────────────┘    └─────────────────────┘
           │
           ▼
┌─────────────────────┐
│     Extracted       │
│     Watermark       │
└─────────────────────┘
```
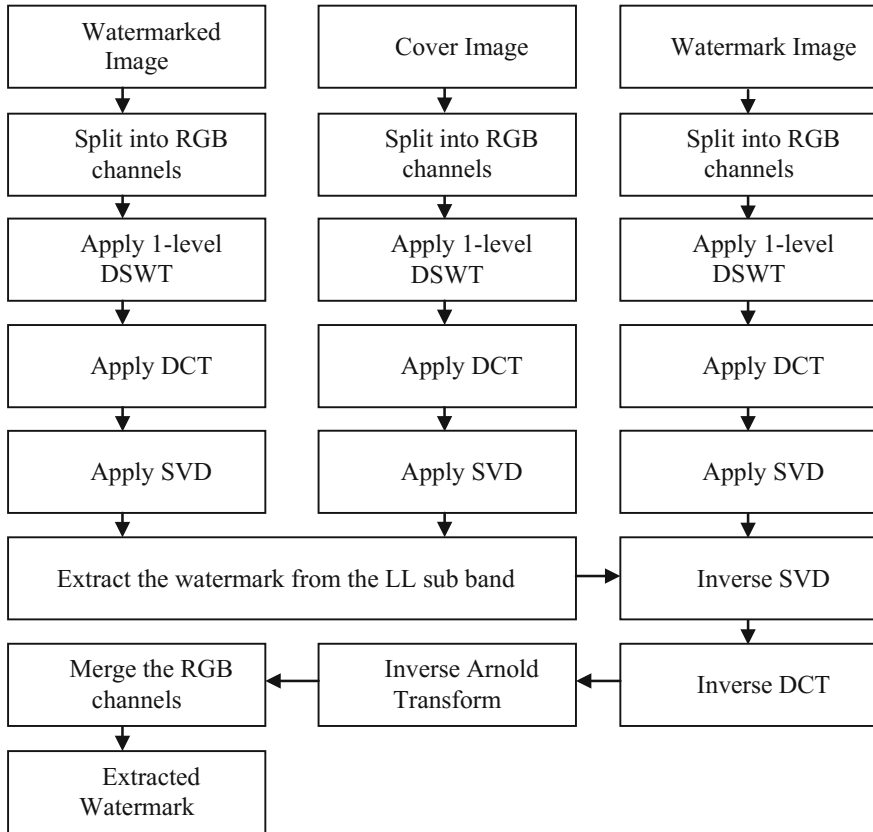
Fig. 4 Extraction process

Step 9: Apply Inverse Arnold transform to get one channel of the watermark.
Step 10: Repeat steps 3–9 for the other two channels.
Step 11: Merge all the three channels to get the final extracted watermark image (Fig. 4).

4   Experimental Results

We tested our proposed algorithm using Lena as the cover image (Fig. 5a) and Peugeot logo as the watermark image (Fig. 5b). The size of cover image and watermark image is 256×256. Both the images used are colored having 24 bit depth.

Where x and y are the coordinates of the original image, N is the height or width of the image and x' and

Fig. 5 a Cover Image **b**
Watermark image

We used PSNR (Peak Signal to Noise Ratio) and NC (Normalised Correlation) to
measure the imperceptibility of the watermarked image and to measure the similarity
between the extracted watermark and original watermark respectively. PSNR [7] was
calculated using the formula

$$PSNR = 10\log\frac{255^2}{MSE} \tag{3}$$

where MSE is the mean Square error given by

$$MSE = \frac{1}{MN}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(C - WM)^2 \tag{4}$$

where C is the Cover Image and WM is the Watermarked Image. Higher the value of
PSNR, more imperceptible the algorithms is. We have recorded the highest PSNR
value for our proposed algorithm as 79.1674 for 0.1 as α.

Normalised Correlation [8] was evaluated using the formula

$$NC = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(E \times W)}{\sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}E^2 \times \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}W^2}} \tag{5}$$

where E is the Extracted watermark and W is the Original watermark. We got 0.9488
as the highest NC value for our proposed algorithm for 0.8 as alpha. The values of
PSNR and NC for different values of α are shown in Table 1. The results have been
calculated when there are no attacks but in practical implementation the watermarked
image can be attacked. We have compared our result with Azadeh et al.'s [9] method
under various common attacks such as JPEG Compression, salt and pepper noise,
median filter, wiener filter and rotation. We have taken the value of α as 0.5 for
comparison.

From Table 2 we can observe that when there is no attack, the proposed method
gives a better PSNR by 21.75%, this is because the proposed method contains SVD
which makes the algorithm imperceptible whereas Method [9] contains only DWT.

Table 1 Experiment results of proposed method for different α values

A	PSNR	NC
0.1	79.1674	0.9083
0.2	73.1470	0.9328
0.3	69.6251	0.9371
0.4	67.1264	0.9383
0.5	65.1882	0.9419
0.6	63.6045	0.9397
0.7	62.2656	0.9397
0.8	61.1058	0.9488
0.9	60.0827	0.9424

Table 2 Comparison of proposed method with DSWT, DCT and SVD

Attack	DSWT	DCT	SVD	Proposed method
No attack	58.5137	31.6187	5.7790	65.1882
JPEG compression	58.2266	5.7538	5.7570	60.8265
Salt and pepper	58.4197	28.6493	5.7789	65.0423
Rotation	55.8512	11.9222	5.7711	57.6388
Cropping	56.6978	12.1125	5.7664	59.0375
Gaussian noise	58.2143	31.6181	5.7790	63.9163
Median filter	58.5105	30.2074	5.7789	65.4643
Averaging filter	58.5254	29.3464	5.7787	65.6652
Sharpening filter	56.7147	17.6405	5.7813	55.9165

Fig. 6 Attacked images for **a** No attack **b** JPEG compression 90% **c** JPEG compression 80% **d** JPEG compression 50% **e** Salt and pepper 0.002 **f** Salt and pepper 0.003 **g** Wiener filter **h** Median filter **i** Rotation 90 deg

We can also see that even under various attacks the PSNR values of the proposed method remains quite high whereas the PSNR values of Method [9] becomes significantly low, especially during the rotation attack (Figs. 6, 7 and Table 3).
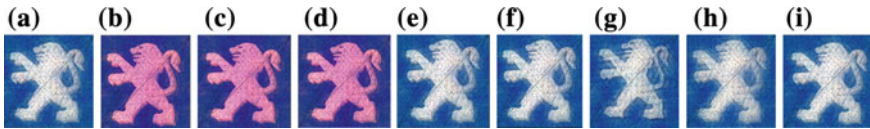
Fig. 7 Extracted images for **a** No attack **b** JPEG compression 90% **c** JPEG compression 80% **d** JPEG compression 50% **e** Salt and pepper 0.002 **f** Salt and pepper 0.003 **g** Wiener filter **h** Median filter **i** Rotation 90 deg

Table 3 Experimental results of proposed method and Method [9] under different attacks

Attack	Proposed method		Method [9]	
	PSNR	NC	PSNR	NC
No attack	65.1882	0.9419	53.5404	0.9986
JPEG compression 90	60.8718	0.9300	34.5933	0.9988
JPEG compression 80	60.8265	0.9303	33.2654	0.9988
JPEG compression 50	60.8691	0.9291	31.4634	0.9988
Salt and pepper 0.002	65.0857	0.9423	32.0912	0.9989
Salt and pepper 0.003	65.0423	0.9427	25.1745	0.9988
Wiener filter	65.5027	0.9304	37.3339	0.9980
Median filter	65.4643	0.9304	35.3360	0.9988
Rotation (90 deg)	57.6388	0.9414	11.7095	0.9986

5 Conclusion

In this research paper we have presented an imperceptible, robust and secure algorithm for Image Watermarking of colored images. All the advantages of DSWT, DCT and SVD are combined to generate this hybrid algorithm and Arnold Transform is added to make it even more secure. The data hiding capacity of the cover image, generally lesser has been equal to its size. Using this method, we are able to get higher values of PSNR by a percentage of 21.75% and hence the proposed algorithm is very imperceptible. The values of NC though less but are comparable to that of Method [9]. Future work can be done towards increasing NC values.

References

1. Lala, H.: Digital image watermarking using discrete wavelet transform. Int. Res. J. Eng. Technol. (IRJET). **04**(01), 1682–1685 (2017)
2. Tyagi, S., Singh, H.V.: Agarwal, R., Gangwar, S.K.: Digital watermarking techniques for security applications. In: International Conference on Emerging Trends in Electrical, Electronics and Sustainable Energy Systems (ICETEESES-16), pp. 379–382. IEEE (2016)
3. Ramamurthy, N., Varadarajan, S. Dr.: Effect of various attacks on watermarked images. Int. J. Comput. Sci. Inf. Technol. (IJCSIT), **3**(2), 3582–3587 (2012)

4. Al-Afandy, K.A., Faragallah, O.S., EL-Rabaie, El.-S.M., Abd El-Samei, F.E., ELmhalawy, A.:
   A hybrid scheme for robust color image watermarking using DSWT in DCT domain. In: 2016
   4th IEEE International Colloquium on Information Science and Technology (CiSt), pp. 444–449
   (2016)
5. Barni, M., Bartolini, F., Cappellini, V., Piva, A.: A DCT domain system for robust image water-
   marking. Signal Process **66**(3), 357–372 (1998)
6. Gaur, S., Srivastava, V.K.: A hybrid RDWT-DCT and SVD based digital image watermark-
   ing scheme using arnold transform. In: 4th International Conference on Signal Processing and
   Integrated Networks (SPIN), pp. 399–404 (2017)
7. Qianli, Y., Yanhong, C.: A digital image watermarking algorithm based on discrete wavelet trans-
   form and discrete cosine transform. In: International Symposium on Information Technologies
   in Medicine and Education, pp. 1102–1105. IEEE (2012)
8. Saravanan, P., Sreekara, M., Manikantan, K.: Digital image watermarking using daubechies
   wavelets. In: 3rd International Conference on Signal Processing and Integrated Networks (SPIN),
   pp. 57–62. IEEE (2016)
9. Karimian, A., Vahidi, J.: A new color image watermarking algorithm using 3-level discrete
   wavelet transform. Int. J. Mechatron. Electr. Comput. Technol. (IJMEC), **7**(26), 3633–3643
   (2017)