Journal of Applied Mathematics and Modelling
https://ejournal.cibnusantara.org/jamm
E-ISSN: 3090-5524

FFT - Based Watermarking for Copyright Protection of Color Image

Joshua Febrian Wiranata[1], Sri Redjeki Pudjaprasetya[2], Pritta Etriana Putri[3], Riski Kurniawan[4*]

[1] Mathematics Study Program, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, West Java Province, Indonesia
[2] Industrial & Financial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, West Java Province, Indonesia
[3] Combinatorial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, West Java Province, Indonesia
[4] Research Center for Computing, National Research and Innovation Agency (BRIN), Jakarta, Indonesia

* Corresponding email: riski.kurniawan@brin.go.id

Article history:
Received : February 7th, 2025
Accepted : March 20th, 2025
Published : April 5th, 2025

DOI :
https://doi.org/10.64570/jamm.v1i1.1

Abstract: Watermarks are commonly used by authors to prevent unauthorized copying and to protect their intellectual property. When a watermark is applied to a digital image, it is expected that the watermark will not be visible, and the watermark will retain its good quality when extracted. This paper is focused on one type of color image watermarking method, using the Fast Fourier Transform (FFT) algorithm. In this algorithm, the watermark is embedded in the frequency domain of the image. In this way, our watermark algorithm exhibits good quality and demonstrates strong resistance against common attacks. Here, we experimented with compression, blur, object addition, and noise addition.

Keywords: color image watermarking; copyright protection; fast Fourier transform; singular value decomposition.

1. Introduction

In today's digital world, information can spread very quickly. Many things have become much simpler to execute due to technological advancements. However, this ease allows for unauthorized copying or illegal property modification by third parties. Watermarking is one method for marking ownership and protecting copyright (Wang et al., 2024). Watermarking is a method of embedding information into digital data to serve as a form of identification for the original owner, and the information embedded is called watermarks. Digital watermarks incorporate a variety of capabilities, such as verifying authenticity through digital signatures, tracking content via fingerprinting, monitoring media distribution, ensuring secure access, restricting unauthorized duplication, and facilitating private communication (Cox et al., 1997, 2000; Karzenbeisser and Perircolas, 2000)

Watermarking techniques range from the very simple to the extremely complex (Das et al., 2010). There are two types of color image watermarking techniques: spatial domain watermarking and frequency domain watermarking. Spatial domain watermarking algorithms are relatively easy to implement and effective, but mostly do not show robustness to common attacks (Cox et al., 2002; Wolfgang et al. 1999). Frequency domain watermarking algorithms are known to produce high quality watermarking and are more successful in achieving the robustness of the watermarking algorithm to image attacks (Dannstaedter et al., 1998). An important step in frequency domain watermarking is to apply a transformation to convert the original image into a frequency domain. Commonly used transforms are the Discrete Cosine Transform (DCT), the Wavelet Transform (WT), the Discrete Contourlet Transform (CT), and the Discrete Fourier Transform (DFT), as implemented in (Tsui et al., 2008; Li et al., 2021; Sheng et al., 2023; Zhang et al., 2025).

This paper focuses on watermarking using the Fast Fourier Transform (FFT) method combined with the singular value decomposition (SVD) technique. SVD is widely used in color image watermarking (Dogan et al., 2011; Hu et al., 2020; Qu et al., 2021; Eltoukhy et al., 2023; Zhang et al., 2023). The main steps in the watermark embedding process are as follows: first, extract a singular value from the watermark using the SVD method; apply FFT to an image, embed the watermark SVD in the frequency domain of the image (its FFT); and finally, use iFFT to generate a watermarked image. In the context of copyright protection, watermarks embedded in digital content are created in such a way that they are invisible to human observers but easily detected by computer algorithms. Our discussion here is restricted to a black and white watermark embedded in color images. Additionally, we discuss the robustness of the proposed method against various image processing attacks, such as compression, blur, object addition, and noise addition.

The organization of this paper is as follows: Section 2 provides a summary of the methods used in the watermarking algorithm, which are Singular Value Decomposition (SVD) and the Fast Fourier Transform (FFT). Section 3 formulates the FFT-watermarking algorithm, including the embedded and extracting steps. Section 4 tests the robustness of the proposed method against various attacks. The last section contains some conclusions.

2. Methods

Watermarking using SVD-FFT

In this chapter, we explore the implementation of watermarking through the SVD-FFT technique. It covers both the embedding and detection processes, along with an analysis of experimental results and discussions on the effectiveness of the algorithm. Watermarking consists of two important steps, those are embedding watermarks and extracting watermarks.

1. Embedding watermark

In this section, we describe the step-by-step procedure for embedding watermarks into targeted images, outlining the intricate process and methodologies to ensure the safe and effective integration of embedded watermarks. It is known that each color image consists of three colors layers, those are red, green, and blue. The watermark is inserted on each layer of the color image via the following steps.

1) Apply the singular value decomposition (SVD) to get the singular value of the watermark.
2) Divide the original image into smaller blocks of 8 × 8 pixels.
3) Select a small block that has not been embedded with a watermark, and apply FFT to that block. Insert the SVD of the watermark singular value for each block using a formula (for a block at the 1st row and 1st column) as follows
$$\hat{B}(1,1) = \hat{B}(1,1) + m_i\sigma_i,$$
with m_i is a scalar factor for i-th singular value σ_i.
4) Apply iFFT and denote it as B', and this is the image with a watermark. Calculate LPSNR between the original image B and the image with a watermark B'
$$\text{LPSNR} = 10\left(\frac{255^2}{\frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8}\left(B_{ij} - B'_{ij}\right)^2}\right).$$
5) If LPSNR is in the desired range value, save the m_i value and replace B with B'.
6) Otherwise, try a different value of m_i.
7) Repeat steps 3-6, until all singular values are inserted. After that, the embedding process is finished. The seven steps above are illustrated in Figure 1.

Watermarking is applying the above steps to each color layered from the image. Steps 3, 4, 5 of embedding the watermark above are an interactive process to find the optimal value of m_i, so that the value $m_i\sigma_i$ inserted in step 3 is not too small and not too large.

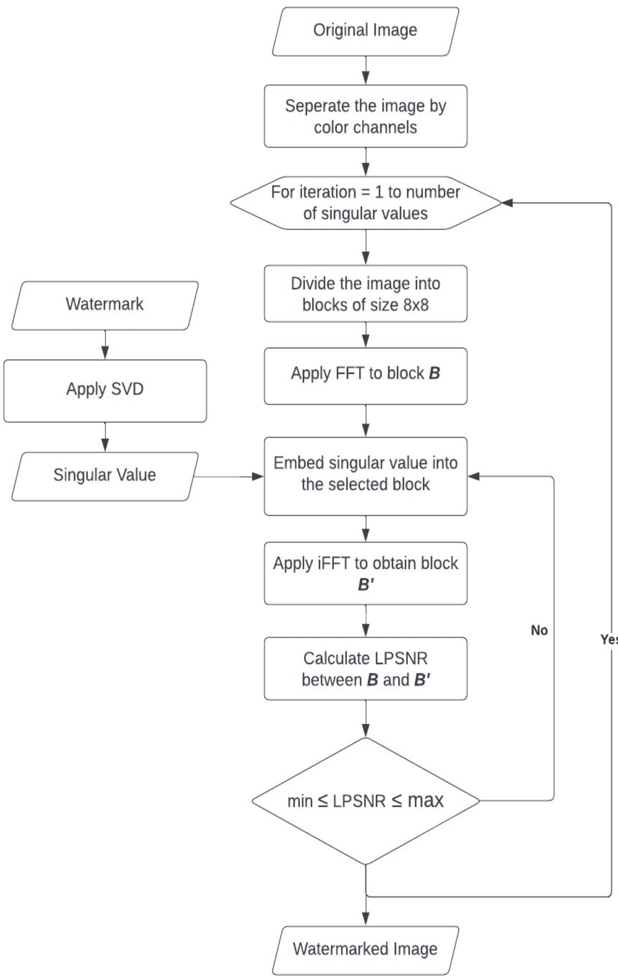


Figure 1. The process of embedding a watermark.

2. Watermark Detection

This section discusses the procedure for detecting watermarks from color images. Here, we explain the methodology used to recover the embedded watermarks. Note that this watermark detection procedure can also be used for images that have been attacked.

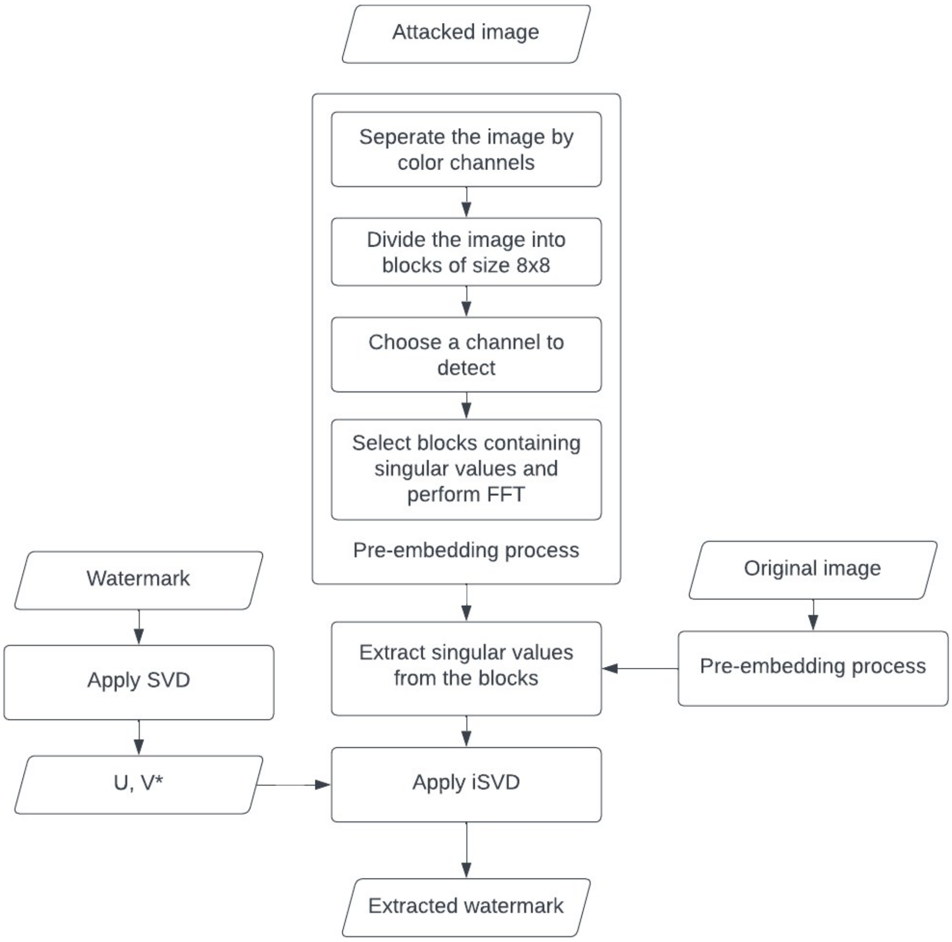


Figure 2. Watermark detection process.

We need to detect watermarks from each colored layer of an image. The process of detecting a watermark on one of the colored layers of an image is depicted in Figure 2, which consists of the following 4 steps.

1) Divide the image and the manipulated image into smaller blocks of 8×8. Use the same block selection flow as the one used for inserting the watermark. Extract the singular value using a formula

$$\sigma_t = \frac{\left(\hat{B}'(1,1) - \hat{B}(1,1)\right)}{m_i}.$$

2) Let $\hat{B}$ denotes the small blocks of the original image and $\hat{B}'$ denotes the small block from the image with a watermark.
3) Perform SVD on the wtaermark image $\hat{B}'$ to obtain U and V^* matrices, then perform SVD inversion using the singular value from the previous step.
4) Finally, all three watermarks can be obtained from each layer of the colored image.

3. Results and Discussion

In this section, we implement the above procedure to embed a watermark into a colored image to obtain a watermark image. Figure 3 below depicts the color image and the grayscale watermark used in the simulation.

Figure 3. (Left) An original image; (middle) a grayscale watermark; (right) a watermarked image.

To quantify the similarity between the watermarked image and the original image, the Peak-Signal-to-Noise-Ratio (PSNR) formula is used, which is as follows:

$$\text{PSNR} = 10\left(\frac{255^2}{\frac{1}{3mn}\sum_{k=1}^{3}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(O_{ijk} - W_{ijk}\right)^2}\right).$$

In the above formula, O denotes the original image, whereas W the watermarked image. The great PSNR value indicates that the original and watermarked images are more identical. For example, above, the PSNR value is 42.4, which indicates that the watermarked image is nearly identical to the original image.

Furthermore, we demonstrate the watermark extraction process. Here, we use a watermarked image that has been attacked with Gaussian blur with a kernel 5 (see Figure 4). By using the original image, the watermark image, and m value from the embedding process, the extracted watermark is obtained as shown below.

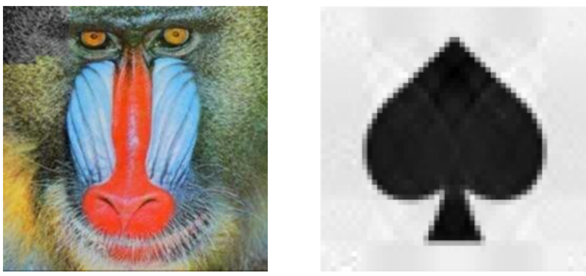


Figure 4. Gaussian blurred image with the extracted watermark.

Based on the results presented above, it is evident that the extracted watermark retains the essential shape characteristics of the original watermark. Extracted watermark from this process can be evaluated by using the Normalized Correlation (NC) formula, as follows

$$\text{NC} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}W'_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[W_{ij}\right]^2},$$

where W denotes the watermark image and W' the extracted watermark. An NC value close to one indicates a strong similarity between W and W', which indicates that the original attributes of the watermark are maintained. Meanwhile, a low NC value indicates a lack of correlation between W and W'. In the next section, we discuss NC values in relation to the robustness of the watermarking against various attacks, including the Gaussian blur.

Robustness Test

The robustness of a watermark procedure is measured by the degree of similarity between the watermark image and the extracted watermark. In this test, four attacks will be carried out on watermarked images obtained in previous simulations, as summarized in Table 1. Watermarked images under various attacks along with the corresponding extracted watermark obtained, are displayed in Table 2.

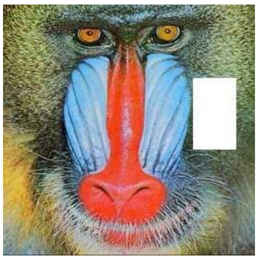
Table 1. Attacks utilized in this study on the watermarked image

Attacks	Description
Compression	The watermarked image file is compressed by 25%
Object	Adding a white block to the watermarked image
Noise	Adding some noise to every single pixel in the watermarked image
Filter	The watermarked image is filtered

Upon analyzing the results presented in Table 2, it becomes evident that the extracted watermark exhibits a remarkable degree of resilience against the tested attacks. Our approach is to first visually inspect the watermark to ensure that the core characteristics are maintained. If the watermark is still visually maintained, then we move to the Normalized Correlation (NC) value, with a target threshold of 0.5. this combined approach allows us to consider both the perceptual appearance of the watermark and a quantitative measure to assess its effectiveness.

This strategy aligns with our project goal, which is to balance the quality of the watermarked image and the robustness of the watermark. The choice of a threshold value of 0.5 NC indicates moderate tolerance to watermark changes, which can be interpreted as long as important features remain recognizable. This approach ensures an appropriate level of security for our specific application. It is worth noting that the extracted watermarks in Table 2 have maintained shape similarity. However, in the case of a noise attack, there is a slight alteration in the basic color, while the filter attack leads to a color swap. In the previous instance, the Gaussian blue attack produced an NC value of 0.908, indicating the watermark robustness to Gaussian attacks. Notably, while compression and object attacks resulted in high NC values of 0.942 and 0.915, respectively, the noise and filter attacks produced lower NC values of 0,56 and 0.259.

Table 2. Attacked images along with their extracted watermark

Attacks	Attacked Watermarked Image	Extracted Watermark
Compression		 NC = 0.942
Object		 NC = 0.915
Noise		 NC = 0.56
Filter		 NC = 0.259

4. Conclusions

In this article, we have successfully applied the SVD-FFT method for embedding a watermark image into a color image. We implemented the method on a color image size 256×256 pixels, while the watermark is a grayscale image size 56×56 pixels. Here, we demonstrate that the SVD-FFT watermark algorithm method exhibits good quality and demonstrates strong resistance against several attacks such as blue, compression, object addition, and noise addition.

Appendix A

This appendix discusses a brief overview of mathematical tools required for the watermarking algorithm, namely: the single value decomposition method (abbreviated as SVD) and the fast Fourier transform method (abbreviated as FFT). Python commands for each method will also be given.

Singular value decomposition (SVD)

The singular value decomposition (SVD) is a factorization of a real or complex matrix. It is a generalization of eigenvalue decomposition that holds for a square matrix.

Let A be a complex $n \times m$ matrix, matrix A is called unitary if its inverse is also its conjugate transpose, that is, if $AA^* = A^*A = I$.

Definition 1. Let a complex matrix $X \in C^{n\times m}$, the singular value decomposition (SVD) of X is a factorization of the form

$$X = U\sum V^*,$$

With $U \in C^{n\times n}$ and $V \in C^{m\times m}$ are complex unitary matrices, whose columns are orthonormal, and $\sum \in \mathbb{R}^{n\times m}$ is a diagonal matrix with non-negative real numbers on the diagonal, such that $\sum_{11} = \sigma_1 \geq \cdots \geq \sigma_n = \sum_{nn}$.

The SVD is not unique, but it is always possible to choose the decomposition so that the singular values $\sum_{ii}$ are in descending order. In this case, $\sum$ is uniquely determined by the matrix A, but not the unitary matrices U and V.

Example: Calculating the SVD of a matrix in Python

Let A be a matrix. The code to calculate the Singular Value Decomposition (SVD) is as follows:

```
import numpy as np
U,S,VT = np.linalg.svd(A)
```

Fast Fourier Transform (FFT)

The discrete Fourier transform (DFT) converts a signal from its original domain (time or space) to a representation in the frequency domain; its vice versa is called the inverse DFT (iDFT).

Let $x_0, x_1, \ldots, x_{N-1}$ be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{n=N-1} x_n e^{-i2\pi kn/N}, \qquad k = 0, \ldots, N-1,$$

where $e^{-i2\pi kn/N}$ is a primitive N^{th} root of 1. Evaluating this definition directly has the tie complexity of $O(N^2)$ operations, because there are N outputs X_k, and each output requires a sum of N terms.

Computing DFT using this definition directly is often too slow. The fast algorithm that computes the discrete Fourier transform (DFT) is the fast Fourier transform (FFT). An FFT rapidly computes such a formula by factorizing the DFT matrix into a product of sparse (mostly zero) factors (Van Loan, 1992). As a result, the FFT algorithm manages to compute the same results in $O(N\log N)$ operations.

This FFT algorithm is widely used for application in signal processing, data compression, and image analysis. Nowadays, most programming languages are equipped with a simple command to compute FFT and its inverse (iFFT).

Example: Using Python language, calculating FFT can be obtained using the command below

```
import numpy as np
fft_results = np.fft.fft(f)
```

In the formula above f denotes the original function, whereas fft_results denotes the resulting transform.

References

Bracewell, R. N. (2000). *The Fourier transform and its applications.* New York: McGraw-Hill.

Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing,* 6(12), 1673-1687. https://doi.org/10.1109/83.650120

Cox, I. J., Miller, M. L., & Bloom, J. A. (2000). Watermarking applications and their properties. *In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)*, 6-10. IEEE. https://doi.org/10.1109/ITCC.2000.844175

Darmstaedter, V., Delaigle, J.-F., Nicholson, D., & Macq, B. (1998). A block based watermarking technique for MPEG2 signals: Optimization and validation on real digital TV distribution links. *In Multimedia Applications, Services and Techniques—ECMATST: Third European Conference, Berlin, Germany, May 26-28, Proceedings, 3*, 190-206. Springer Berlin Heidelberg.

Das, S., Bandyopadhay, P., Paul, S., Chauduri, A., & Banarjee, M. (2010). An invisible color watermarking framework for uncompressed video authentication. *International Journal of Computer Applications,* 1(11), 22-28

Dogan, S., Tuncer, T., Avci, E., & Gulten, A. (2011). A robust color image watermarking with singular value decomposition method. *Advance in Engineering Software,* 42, 336-346. https://doi.org/10.1016/j.advengsoft.2011.02.012

Eltoukhy, M. M., Khedr, A. E., Abdel-Aziz, M. M., & Hosny, K. M. (2023). Robust watermarking method for securing color medical images using Slant-SVD-QFT transforms and OTP encryption. *Alexandria Engineering Journal*, 78, 517-529. https://doi.org/10.1016/j.aej.2023.07.068

Gonzales., R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson Education India.

Hu, H. T., Hsu, L. Y., & Chou, H. H. (2020). An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated. *Information Sciences,* 519, 161-182. https://doi.org/10.1016/j.ins.2020.01.019

Katzenbeisser, S., & Peticolas, F. A. (Eds.). (2000). *Information hiding techniques for steganography and digital watermarking*. Artech House.

Li, L., Bai, R., Lu, J., Zhang, S., & Chang, C. C. (2021). A watermarking scheme for color image using quaternion discrete Fourier transform and tensor decomposition. *Appl. Sci.,* 11, 5006. https://doi.org/10.3390/app11115006

Qu, G., Meng, X., Yang, X., Wu, H., Wang, P., He, W., & Chen, H. (2021). Optical color watermarking based on single-pixel imaging and singular value decomposition in invariant wavelet domain. *Optics and Lasers in Engineering,* 137, 106376. https://doi.org/10.1016/j.optlaseng.2020.106376

Sheng, T., Zeng, W., Yang, B., & Fu, C. (2023). Adaptive multiple zero-watermarking based on local fast quaternion radial harmonic Fourier moments for color medical images. *Biomedical Signal Processing and Control,* 86, 105304. https://doi.org/10.1016/j.bspc.2023.105304

Sumantri, J. S. (2011). Watermarking pada citra warna menggunakan teknik SVD-DCT berdasarkan local peak SNR.

Tsui, T. K., Zhang, X. P.., & Androutsos, D. (2008). Color image watermarking using multidimensional Fourier transforms. *IEEE Transactions on Information Forensics and Security,* 3 (1), 16-28. https://doi.org/10.1109/TIFS.2007.916275

Van Loan, C. (1992). *Computational frameworks for the fast Fourier transform*. Society for Industrial and Applied Mathematics.

Wang, H., Ma, R., & Niu, P. (2024). Color image watermarking using vector SNCM-HMT. *J. Vis. Commun. Image R.,* 105, 104339. https://doi.org/10.1016/j.jvcir.2024.104339

Zhang, H., Li, Z., Chen, Y., Lu, C., & Yan, P. (2025). Fast image reconstruction method using radial harmonic Fourier moments and its application in digital watermarking. *Journal of Franklin Institute,* 36, 107391. https://doi.org/10.1016/j.jfranklin.2024.107391

Zhang, M., Ding, W., Li, Y., Sun, J., & Liu, Z. (2023). Color image watermarking based on a fast structure-preserving algorithm of quaternion singular value decomposition. *Signal Processing,* 208, 108971. https://doi.org/10.1016/j.sigpro.2023.108971