Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

# Algebraic Decomposition Method for Zero Watermarking Technique in YCbCr Space

**Nada S. Mohammed***, **Areej M. Abduldaim**

Division of Mathematics and Computer Applications Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.

*Corresponding author Email: as.18.97@grad.uotechnology.edu.iq

## HIGHLIGHTS

- . Build the zero watermarking techniques used the algebraic Hessenberg decomposition method to extract features from the host images for the first time.
- The employing of the HDM is successful, not only algebraically, but the strength and importance of the obtained outcomes are equivalent to the best outcomes obtained from employing any common transform that extracts the features of the image optimally.
- Based on the outcomes of the NC values obtained from the two methods mentioned in this paper, it is found that the NC values improved with the employ of DCT.
- The suggested technique gave good robustness contra various kinds of popular attacks.

## ARTICLE INFO

**Handling editor:** Rana F. Ghani

**Keywords:** Discrete Cosine Transform (DCT); Hessemberge Decomposition Method (HDM); YCbCr Space; Zero Watermarking.

## ABSTRACT

The close connection between mathematics, especially linear algebra, and computer science has greatly impacted the development of several fields, and the most important is image processing. Algebraic methods aroused interest in building digital image watermarking techniques and are used to find the features of the image to hide the watermark. This paper aims to use the algebraic Hessemberge decomposition method (HDM) for the first time as a transformation to extract the features of the image without using any popular transformation for building zero watermarking. To achieve the aim, two techniques are used, HDM with and without discrete cosine transform (DCT); both depend on the advantage of the algebraic HDM to convert the image to another domain in the YCbCr space. After applying eleven common attacks to images in both techniques, the results showed that the NC values under the influence of many attacks were higher in the second technique than the NC values in the first technique. In contrast, the NC values for salt and pepper attack in the first technique are higher than the NC values in the second technique.

## 1. Introduction

The possibility of representing digital images in the form of matrices facilitated the work in image processing. Applying mathematical methods has been very effective, especially algebraic decompositions. Besides, these algebraic methods have contributed to building many algorithms related to digital watermarking [1]. The watermark is generally embedded in the cover image or a part of the cover image. After that, new ideas have appeared in building these techniques that relied on including the watermark image into the matrix features extracted from the original image. These techniques are called zero watermarking, which does not depend on changing the original image to hide the watermark image so that the original image remains without changes. The effectiveness of these techniques is measured by the two important properties: robustness and imperceptibility [2].In this paper, two techniques for zero watermarking are suggested depending on the algebraic Hessemberge decomposition method (HDM), which is considered an algebraic transformation to extract the features in the YCbCr space. The YCbCr is applied to the color image in the first technique to obtain the Y channel that exemplifies the gray-scale image. Then the HDM is executed to generate the master secret. In the second technique, the DCT is employed first on the Y channel, and then the HDM is applied to the matrix of the DC values to construct the master secret.



http://doi.org/10.30684/etj.v40i4.2028
Received 28 February 2021; Accepted 19 April 2021; Available online 25 April 2022
2412-0758/University of Technology-Iraq, Baghdad, Iraq

The organization of this paper is as follows. In section 2, some works of literature are reviewed that are concerned with the algebraic tools used to build watermarking and zero watermarking techniques. Then, the basics of the Hessemberge decomposition method (HDM), equation system of the YCbCr space, and discrete cosine transform (DCT) are given in section 3,. The suggested zero watermarking techniques depending on HDM with and without DCT are introduced in section 4. In section 5, the results and measurements of the robustness and imperceptibility are computed, and HDM's effect and significance are discussed. Section 6 is devoted to comparing the two suggested zero watermarking techniques and, on the other hand, comparing these two techniques with previous techniques. Finally, the conclusion is given in section 7.

## 2. Literature review

Hessenberg's decomposition method is utilized in various ways in multiple papers to create watermark techniques. In [3], the algorithm is performed to embed a color watermark image into a color host image in the RGB space after every channel of the watermark image is permuted by Arnold transform. While, Abodena and Agoyi, in [4] offer a blueprint for a blind watermarking scheme for color images utilizing two-level discrete wavelet transform (DWT) and fast Walsh Hadamard transform (FWHT). A watermarking method is suggested in [5] that includes multi-level discrete wavelet transform (DWT) with singular value decomposition (SVD) on both the original host image and the watermark image to perform the embedding procedure. The technique suggested in [6] is utilized just the discrete wavelet transform (DWT) on the original cover image. It is worth mentioning that, as far as we know, there is no zero watermarking technique that uses the HDM.

On the other hand, another algebraic decomposition method can be used in watermarking with various tools and transforms. The algebraic LU decomposition method is used in [7] with the Mamdani fuzzy inference system (FIS) and lifting wavelet transform (LWT) to build a watermarking technique. The FIS is employed in the original image to produce the controller parameter, which is working on attaining a trade-off between imperceptibility and robustness and is combined with the LU decomposition method to embed the watermark using LWT. The results test shows that the suggested algorithm is strong contra popular attacks. Also, many authors combine the SVD with different transforms to achieve the best results of robustness and imperceptibility in zero watermarking techniques. A zero watermarking technique is suggested in [8] to extract the features matrix from a color image using the SVD and DWT in the YCbCr space. A robust zero watermarking scheme of a color image depends on SVD, and visual cryptography is proposed in [9]. Next, for more improvement, Waleed et al. in [10] suggested a technique in which the most important features are extracted to create a watermark in the YCbCr space using the SVD and Shuffled Frog Leaping (SFL) Algorithm for the optimization. The implementation analysis presents that the suggested technique is highly strong under different attacks.

## 3. Background

This section is dedicated to giving the required basic information utilized to build the suggested techniques in this paper.

### 3.1 Hessenberg decomposition method (HDM)

Hessenberg decomposition method (HDM) is one of the algebraic decomposition and is described in [11] as the decomposition of a common matrix C by orthogonal similarity transformations as follows:

$$C = PHP^T \tag{1}$$

As in general, if C is an $n \times n$ matrix

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

then

$$C = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \times \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & hp_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nn} \end{bmatrix} \times \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} \tag{2}$$

The two matrices that outcome of the Hessenberg decomposition method (HDM) are $H$ and $P$. The matrix $H$ is an Upper-matrix such that $h_{ij} = 0$ where $i > j + 1$ and the matrix $P$ expresses an orthogonal matrix. Typically HDM is calculated through Householder matrices. Where The Householder matrix $B$ represents the orthogonal matrix of the form:

$$B = \frac{I_n - v^T v}{v\, v^T} \tag{3}$$

where $v$ is a vector in $R^n$ and is a non-zero, the $I_n$ is the $n \times n$ identity matrix. That means there are the $n-2$ steps in the overall procedure when the size of $B$ is $n \times n$. Thus, the (HDM) is calculated as:

$$H = (B_1 B_2 \dots B_{n-2})^T \, C \, (B_1 B_2 \dots B_{n-2}) \tag{4}$$

$$H = P^T C P \tag{5}$$

where $P = B_1 B_2 \dots B_{n-3} B_{n-2}$The inverse Hessenberg decomposition method (HDM) can stand for as:

$$C = PHP^T \tag{6}$$

For example, if $C$ is a $3 \times 3$ matrix

$$C = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

then

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.496 & -0.868 \\ 0 & -0.868 & -0.496 \end{bmatrix} \times \begin{bmatrix} 1 & -3.597 & -0.248 \\ -8.062 & 14.046 & 2.8308 \\ 0 & 0.8308 & -0.046 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.496 & -0.868 \\ 0 & -0.868 & -0.496 \end{bmatrix}$$

where

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.496 & -0.868 \\ 0 & -0.868 & -0.496 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & -3.597 & -0.248 \\ -8.062 & 14.046 & 2.8308 \\ 0 & 0.8308 & -0.046 \end{bmatrix}$$

### 3.2 YCbCr space

The YCbCr space represented by a system of equations is one of the spaces that transform the color of images from the RGB space into a new space. This space transforms the image matrix into a three-channel. The difference between the RGB space and YCbCr space is that RGB represents red, green, and blue colors. While YCbCr embodies color as brightness and two color difference signals. In the YCbCr space, the Y channel is the color's brightness (luma) channel, while Cb represents the blue channel and Cr represents the red channel [12].

In [13] it is explained how the YCbCr space can be represented as a system of equations:

$$\mathrm{Y} = 0.299\,\mathrm{R} + 0.587\,\mathrm{G} + 0.114\,\mathrm{B} \tag{7}$$

$$\mathrm{Cb} = -0.1687\,\mathrm{R} - 0.3313\,\mathrm{G} + 0.5\,\mathrm{B} + 12 \tag{8}$$

$$\mathrm{Cr} = 0.5\,\mathrm{R} - 0.4187\,\mathrm{G} - 0.0813\,\mathrm{B} + 128 \tag{9}$$

This space represents the image more efficiently, as it detaches the channels of luminance and coloring in a specific perspective and uses fewer parts to color luminance.

### 3.3 Discrete cosine transform (DCT)

In [14], the DCT is given to transform an image from the spatial domain to the frequency domain by two-dimensional DCT that allows the image to restore from the DCT domain to the frequency domain by applying the inverse of the two-dimension DCT. Any image of dimension $M \times N$ can be represented as a function $f(x, y)$ of two space variables $x$ and $y$: $(x = 0,1,\dots, M-1, y = 0,1,\dots, N-1)$. The DCT and its inverse are applied through equations (10) and (11), respectively:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{\pi(2x+1)u}{2M}\right] . \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{10}$$

$$f(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} C(x,y) \cos\left[\frac{\pi(2x+1)u}{2M}\right] . \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{11}$$

Where $C(u, v)$ is the DCT coefficient of the image $f(x, y)$ at the location $(u, v)$, $M \times N$ is the dimension of the image $f(x, y)$ and $u$ and $v$ are the horizontal and the vertical locations ($u = 0,1,\dots, M-1$, $v = 0,1,\dots, N-1$). The values of $\alpha(u)$ and $\alpha(v)$ are obtained by applying equations (12) and (13), respectively.

$$\alpha(u) = \begin{cases} \sqrt{1/M}, & u = 0 \\ \sqrt{2/M}, & 1 \leq u < M - 1 \end{cases} \quad (12)$$

$$\alpha(\boldsymbol{v}) = \begin{cases} \sqrt{1/N}, & \boldsymbol{v} = 0 \\ \sqrt{2/N}, & 1 \leq \boldsymbol{v} < N - 1 \end{cases} \quad (13)$$

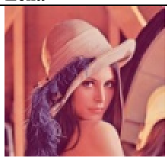
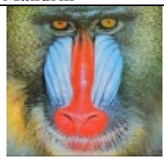
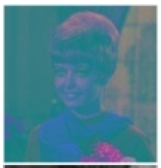
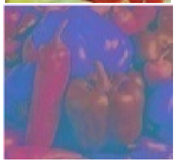
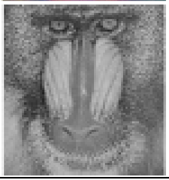
The DCT consists of four frequency bands: the single direct current (DC) coefficient, the low-frequency coefficients (BL), the height frequency band (BH), and the middle-frequency coefficients (BM).

## 4. Suggested techniques

This section is split into two parts, the first part utilizing HDM without any common transform. While in the second part, the DCT is performed before applying HDM. In the two parts, HDM is used to extract the features of the Y component, which represents the luma channel of the YCbCr space of the color image.

Table 1 displays the host color images, the outcome images using the YCbCr space, and the Y gray-scale images.

**Table 1:** The host color images, images of the YCbCr, and Y gray-scale images

| | Lena | Girl | Mandrill | Peppers |
|---|---|---|---|---|
| Host color image |  |  |  |  |
| YCbCr image |  |  |  |  |
| Y gray-scale images |  |  |  |  |

### 4.1 The proposed technique depending on HDM without transforming

The first technique of zero watermarking based on HDM is given to embed and extract a watermark in the original images without utilizing DCT.

#### 4.1.1 Embedding procedure

In this part of the process, the embedding starts by processing the host image to get the features by performing the Hessenberg decomposition method (HDM) on channel Y in the frequency domain. The HDM is suggested to apply to every one of all the $8 \times 8$ blocks obtained from the $512 \times 512$ Y channel after splitting into $8 \times 8$ non-overlapping blocks to get $64 \times 64$ blocks. The algebraic HDM works on decomposing every block of size $8 \times 8$ into two matrices of the same size, one of them is an upper triangular matrix, and the other is orthogonal. The upper triangular matrix of every block is taken to generate one matrix representing the feature matrix that has a size equal to the size of the Y channel. Then this matrix is transformed into a binary matrix and performs XORing between the binary matrix and the watermark. For more safety, the chaotic logistic map is utilized in the resulting matrix from XORing to change the values places to obtain the master secret. Figure 1 represents the embedding process of the watermark in the feature matrix extracted from channel Y.

Step1: Input the host color image of size $n \times n$ (in our case, $512 \times 512$).

Step2: Perform YCbCr on the host image to get the three channels Y, Cb, and Cr.

Step3: Select the Y channel.

Step4: Split the Y channel into $8 \times 8$non-overlapping blocks.

Step5: Perform the algebraic Hessenberg decomposition method (HDM) on every block in step 4 to get the form $P_i$ and $H_i$ and select the matrices $H_i$ of every block to form a new matrix $M$.

Step6: Transform the matrix $M$ to a binary matrix $M_b$ as follows:

$M_b\ (i,j)\ = 1$ if $M_b\ (i,j) > M_b\ (i,j+1)$

$M_b\ (i,j)\ = 0$ otherwise

Where $1\ \leq i \leq\ 512$ and $1\ \leq j \leq\ 512$

Step7: Input watermark image $W$.

Step6: Apply XOR logical operation between the binary features of the Y channel $M_b$ and the binary watermark $W_b$ to get K.

Step8: To increase safety, the chaotic logistic map is used $K$to change the values places to get the master secret.

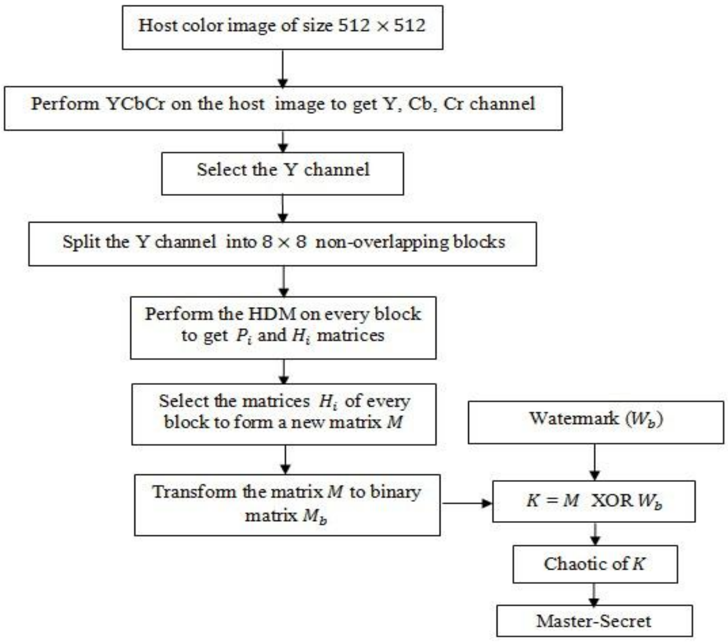



**Figure 1:** The diagram of the embedding procedure using HDM without DCT

### 4.1.2 Extraction procedure

Now, the process of extracting the watermark is given in the following steps:
Step1: Input the host color image of size $n \times n$ (in our case, $512 \times 512$).
Step2: Perform YCbCr on the host image to get the three channels Y, Cb, and Cr.
Step3: Select the Y channel.
Step4: Split the Y channel into $8 \times 8$ non-overlapping blocks.
Step5: Perform the algebraic Hessenberg decomposition method (HDM) on every block in step 4 to get the form $P_i$ and $H_i$ and select the matrices $H_i$ of every block to form a new matrix $M$.
Step6: Transform the matrix $M$ to a binary matrix $M_b$ as follows:

$M_b\ (i,j)\ = 1 \quad \text{if} \quad M_b\ (i,j) > M_b\ (i,j+1)$

$M_b\ (i,j)\ =\ 0 \quad \text{otherwise}$

Where $1\ \leq i \leq\ 512$ and $1\ \leq j \leq\ 512$

Step7: Input the master secret and apply the chaotic logistic map on the master secret to get a matrix K.
Step8: Apply XOR logical operation between the binary matrix $K$ and the binary features $M_b$of the Y channel to get Watermark. Figure 2 illustrates the extracting process of the watermark.

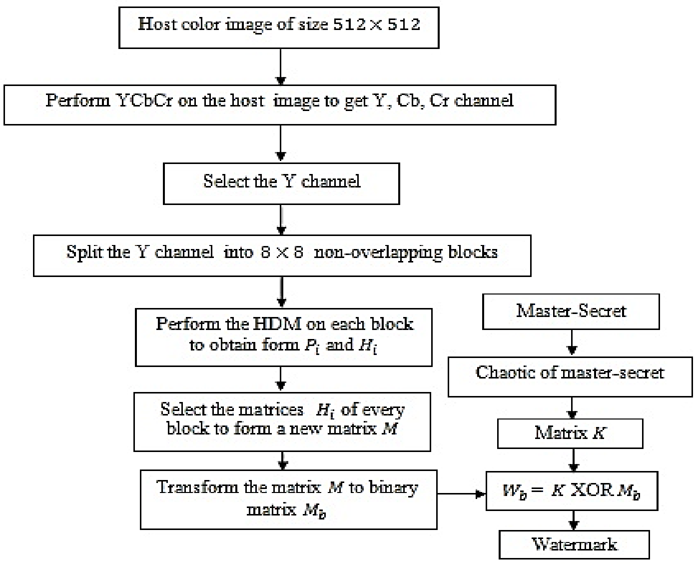



**Figure 2:** The diagram of the extraction procedure using HDM without DCT

## 5. The proposed technique depending on HDM with Dct

The second technique of zero watermarking based on HDM is to embed and extract a watermark in host images utilizing DCT.

### 5.1 Embedding procedure

In this procedure, the effect of the DCT on the embedding steps is demonstrated. Split The Y channel of size $512 \times 512$ into $8 \times 8$ non-overlapping blocks to get $64 \times 64$ blocks, and each block is decomposed by the DCT. The most significant portion represented by DC coefficients is chosen, the element in the position (1,1) in every block, to create one matrix of size $64 \times 64$. After that, divide this new matrix into $8 \times 8$ non-overlapping blocks, applying the HDM on each block to get the orthogonal matrices $P_i$ and upper matrices $H_i$. Collect the upper matrices $H_i$ and put them in one matrix to create a matrix of the features of size $64 \times 64$. Next, transforming this matrix into a binary matrix and XORing the binary matrix and the watermark. Finally, to increase safety, the chaotic logistic map is utilized in the resulting matrix from XORing to replace the location of the values to obtain the master secret. Figure 3 represents the embedding process of this technique.

Step1: Input the host color image of size $n \times n$ (in our case, $512 \times 512$).
Step2: Perform YCbCr on the host image to get the three channels Y, Cb, and Cr.
Step3: Select the Y channel.
Step4: Split the Y channel into $8 \times 8$ non-overlapping blocks.
Step5: Perform the DCT on each block and choose the DC element, the element in the position (1,1) in every block.
Step6: Collect all the DC elements to create one new matrix $D$ of size $64 \times 64$.
Step7: Divide the matrix $D$ into $8 \times 8$ non-overlapping blocks.
Step8: Perform the HDM on each block of the matrix $D$ to obtain the form $P_i$ and $H_i$ and select the matrices $H_i$ of every block to create the feature matrix $M$.
Step9: Transform the matrix $M$ to a binary matrix $M_b$ as follows:
$M_b\,(i,j) = 1 \quad \text{if} \quad M_b\,(i,j) > M_b\,(i,j+1)$
$M_b\,(i,j) = 0 \quad \text{otherwise}$
Where $1 \leq i \leq 64$ and $1 \leq j \leq 63$
Step10: Input watermark image $W$.
Step11: Perform XOR logical operation between the binary features $M_b$ and the binary watermark $W_b$ to get.
Step12: To increase safety, the chaotic logistic map is used $K$ to change the values places to get the master secret.

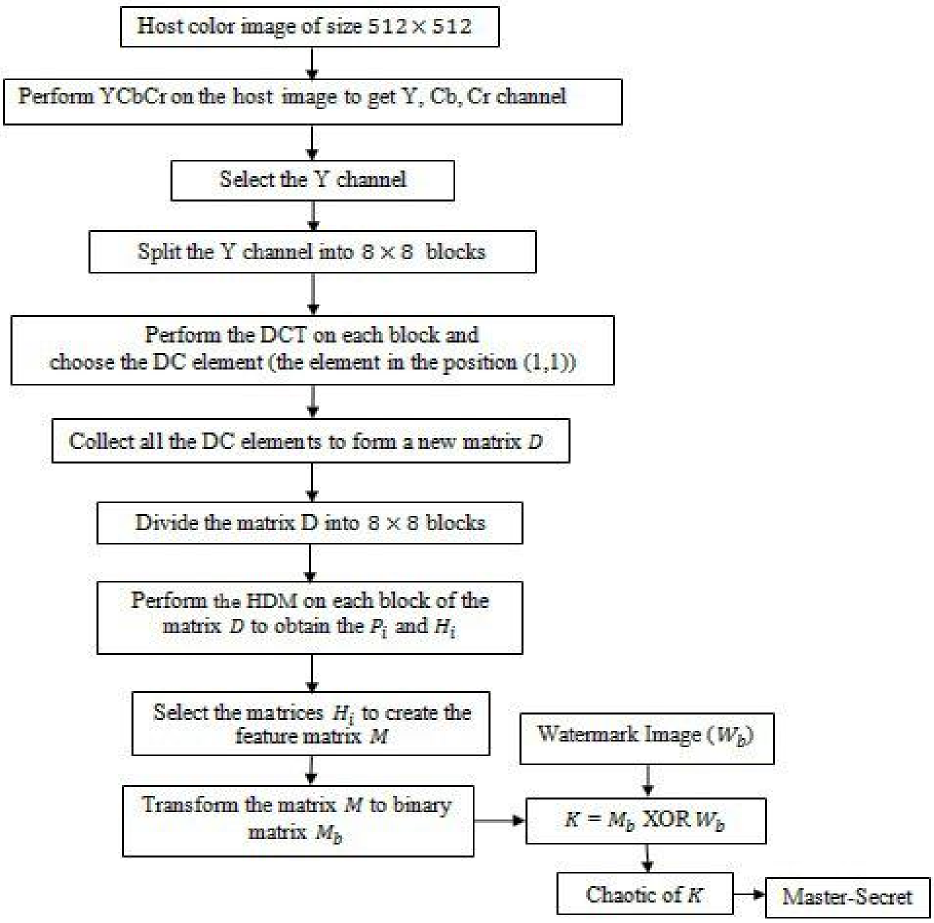



**Figure 3:** The diagram of the embedding procedure using HDM with DCT

### 5.1.1 Extraction procedure

The stage in the process illustrates extracting the watermark, as shown in Figure 4.

Step1: Input the host color image of size $n \times n$ (in our case, $512 \times 512$).

Step2: Perform YCbCr on the host image to get the three channels Y, Cb, and Cr.

Step3: Select the Y channel.

Step4: Split the Y channel into $8 \times 8$ non-overlapping blocks.

Step5: Perform the DCT on each block and choose the DC element, the element in the position (1,1) in every block.

Step6: Collect all the DC elements to form a new matrix $D$.

Step7: Divide the matrix $D$ into $8 \times 8$ non-overlapping blocks.

Step8: Perform the HDM on each block of the matrix $D$ to obtain the form $Pi$ and $Hi$ and select the matrices $H_i$ of every block to create the feature matrix $M$.

Step9: Transform the matrix $M$ to a binary matrix $M_b$ as follows:

$M_b\ (i,j)\ = 1 \quad \text{if} \quad M_b\ (i,j) > M_b\ (i,j+1)$

$M_b\ (i,j)\ = \ 0 \qquad \text{otherwise}$

Where $1\ \leq i \leq\ 64$ and $1\ \leq j \leq\ 63$

Step10: Input the master secret and applies the chaotic logistic map on the master secret to get a matrix $K$.

Step11: Apply XOR logical operation between the binary matrix $K_b$ and the binary features $M_b$ of the DC to get the watermark.

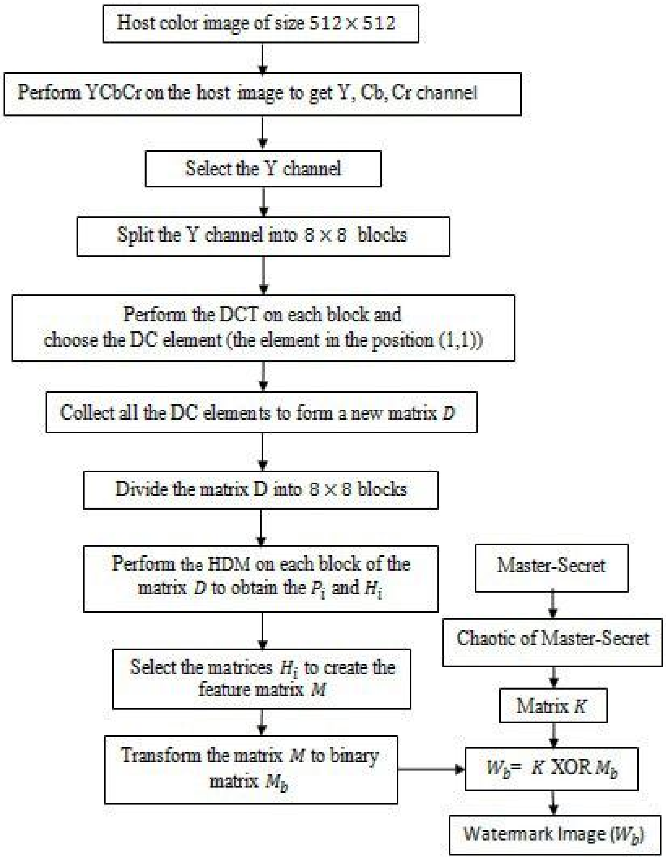



**Figure 4:** The diagram of the extraction procedure using HDM with DCT

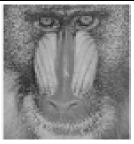
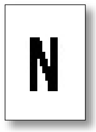
## 6. The experimental outcomes of the HDM working

In this section, the results of the images before and after the attack influence are presented and explained in three cases with and without DCT. These cases are the precision of the proposed technique, the effect and the significance of HDM, and the outcomes of implementing the HDM algorithm after attacks.

### 6.1 The precision of The proposed technique

Before exposing the images to any attack, some tests are implemented to evaluate the imperceptibility and robustness of the suggested zero watermarking techniques. First, the suggested zero watermarking technique is tested without DCT on gray-scale images represented by the Y channel that resulted from using the system of equations of the YCbCr space on the original color image of size $512 \times 512$. The image of size $512 \times 512$ is used as the watermark image. While in the case of employing DCT, the watermark image is used of size $64 \times 64$. Table 2 shows the Y gray-scale images and the binary watermark image used to test the suggested algorithm.

**Table 2:** The Y gray-scale images and the binary watermark image used to test the suggested algorithm

| Lena | Girl | Mandrill | Peppers | Watermark |
|---|---|---|---|---|
|  |  |  |  |  |

Now, without employing DCT, the Y channel of size $512 \times 512$ that expresses the gray-scale images are decomposed into $8 \times 8$ non-overlapping blocks to obtain $64 \times 64$ blocks. Transformed each $8 \times 8$ block utilizing HDM to the frequency domain that split each block into two matrices $P_i$ and $H_i$ where $H_i$ exemplifies upper matrices and $P_i$ exemplifies the orthogonal matrices in all blocks. The upper matrices $H_i$ are selected from all blocks and set in one matrix of size $512 \times 512$. This matrix represented the matrix of features that XORing with the binary watermark to obtain the master secret.

Next, in the case of DCT, the gray-scale images of size $512 \times 512$ are expressed by the Y channel that is split into $8 \times 8$ non-overlapping blocks to get $64 \times 64$ blocks. Each block is decomposed by a DCT. The most significant portion represented by DC coefficients is chosen, the element in the position (1,1) in every block, to create one matrix of size $64 \times 64$. After that, split this new matrix into $8 \times 8$ non-overlapping blocks, using the HDM on each block to get the orthogonal matrices $P_i$ and upper matrices $H_i$ group the upper matrices $H_i$ and put them in one matrix to create a new matrix of size $64 \times 64$. This matrix represents the matrix of features that XORing with the watermark to obtain the master-secret. In principle, to check that the suggested algorithm is working right to obtain logical and precise results from a mathematical point of view, the NC measurement is used to determine whether the outcomes got are passable or not (the virtual value of NC is equal to 1).

In both cases (with and without) applying DCT, the NC is equal to 1 before exposing the images to any attack. Table3 shows the results obtained after implementing the suggested techniques.

**Table 3:** The value of NC of images without attacks

| Images | Lena | Girl | Mandrill | Peppers |
|---|---|---|---|---|
| NC | 1 | 1 | 1 | 1 |

## 6.2 The effect and The significance of HDM

The primary goal of applying algebraic methods is to extract the feature matrix from the Y channel to embed the binary watermark. The embedded watermark is the same after the extraction process. After dividing the Y channel into non-overlapping $8 \times 8$ blocks, the suggested embedding algorithm used the HDM only without any popular transform to decompose every block into lower frequencies (lower important information in the image) expressed by the orthogonal matrix $P_i$ and higher frequencies (important information in the image) expressed by the upper matrix $H_i$ in several steps. The algebraic matrix decomposition way works roughly as a transformation. Because both of them detach the image into high and low frequencies. According to all of the above, it is noted that the embedded watermark can be extracted as same as the original watermark image. Moreover, the way of extracting the watermark from the attacked images provided good and acceptable outcomes.

Furthermore, the important aim of using DCT is to attempt to improve beforehand obtained outcomes. The Y channel is split into 8×8 non-overlapping blocks. After that, apply the DCT on every block and choose the most important part represented by the DC element of each block. Select the DC element, the element in the position (1,1), to create one matrix of size $64 \times 64$. Divide this matrix into $8 \times 8$ non-overlapping blocks. The HDM is performed on every block to get the orthogonal matrices $P_i$ and upper matrices $H_i$ set the upper matrices $H_i$from each block and put them in one matrix to create a matrix of the features of size $64 \times 64$.When performing the suggested algorithms, the capability to extract the watermark perfectly identical to the original watermark became a real thing. Moreover, the process of extracting the watermark from the images under attack is acceptable.

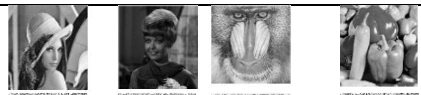
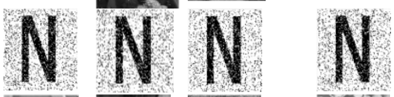
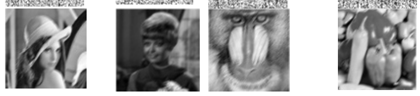
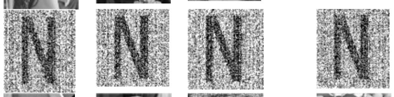
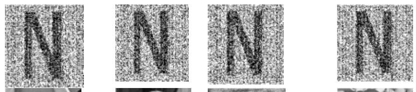
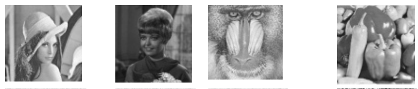
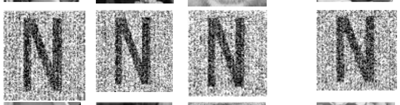
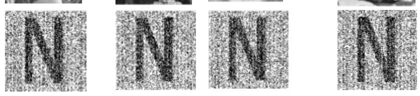
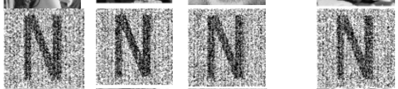
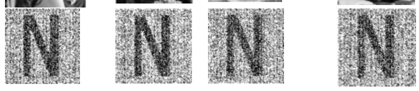
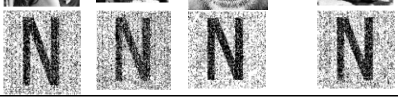
## 6.3 The outcomes of Implementing the Hdm algorithm after attacks

After splitting the Y channel into $8 \times 8$ blocks, the HDM is implemented. This case is before employing the DCT. To get improved results, the DCT is applied on the Y channel and then performed the HDM with the DC band to show the difference between the two cases. To test the robustness and imperceptibility of the zero watermarking techniques with and without employing DCT, Table 4 shows all the outcomes of PSNR and NC after extracting the watermark from the attacked images. While Table 5 gives the extracted watermark image after the images are attacked.

**Table 4:** The NC and PSNR values for images employing HDM without DCT

| Attacks | Lena | | Girl | | Mandrill | | Peppers | |
|---|---|---|---|---|---|---|---|---|
| | *PSNR* | *NC* | *PSNR* | *NC* | *PSNR* | *NC* | *PSNR* | *NC* |
| Salt&Pepper noise1% | 25.4779 | 0.908338 | 24.9784 | 0.90729 | 25.5722 | 0.914883 | 25.3765 | 0.909106 |
| Jpeg_Compression | 36.0368 | 0.773203 | 44.1647 | 0.781181 | 29.6229 | 0.789459 | 34.8115 | 0.769885 |
| Imadjust | 19.2416 | 0.910365 | 18.9837 | 0.857076 | 17.9147 | 0.946341 | 18.3925 | 0.936073 |
| | 36.9599 | 0.792364 | 43.4714 | 0.79121 | 29.4818 | 0.804427 | 35.9493 | Wiener |
| | 28.2566 | 0.778637 | 33.9565 | 0.79198 | 20.6762 | 0.777319 | 28.0312 | Ordfilter |
| Histequalization | 17.2495 | 0.841091 | 10.5599 | 0.816066 | 16.2737 | 0.881157 | 17.9648 | 0.853169 |
| Gaussian Low Pass Filter | 31.9246 | 0.784098 | 38.8328 | 0.789562 | 24.8087 | 0.786007 | 32.1989 | 0.780348 |
| Motion Filter | 26.0047 | 0.767995 | 29.1194 | 0.770087 | 21.2038 | 0.766485 | 25.8218 | 0.767163 |
| Average Filter | 31.9246 | 0.784098 | 38.8328 | 0.789562 | 24.8087 | 0.786007 | 32.1989 | 0.780348 |
| Median Filtering (2 × 2) | 30.3126 | 0.78318 | 35.7099 | 0.790527 | 24.4649 | 0.784247 | 30.2598 | 0.779636 |
| Disk Filter | 32.6027 | 0.7731 | 41.2078 | 0.784994 | 23.8729 | 0.773103 | 32.7611 | 0.769124 |

**Table 5:** The extracted binary watermark from the attacked images employing HDM without DCT

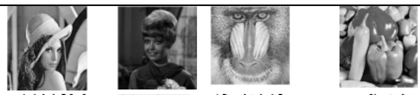
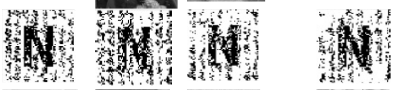
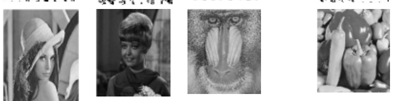
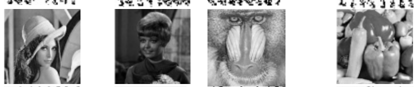
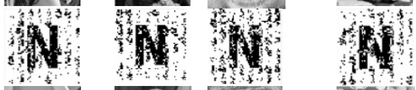
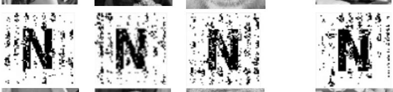
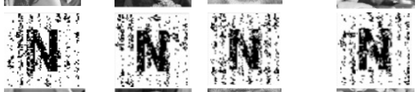
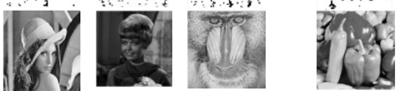
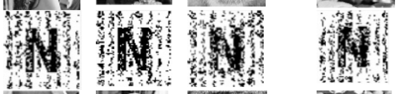
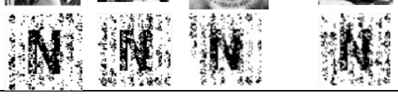
| Attacks | Lena | Girl | Mandrill | Peppers | Attacks | Lena | Girl | Mandrill | Peppers |
|---|---|---|---|---|---|---|---|---|---|
| Salt&Pepper noise1% |  | | | | Gaussian Low Pass Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Jpeg_ Compression |  | | | | Motion Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Imadjust |  | | | | Average Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Wiener |  | | | | Median Filtering (2 × 2) |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Ordfilter |  | | | | Disk Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Hist equalization |  | | | | | | | | |
| Extracted watermark |  | | | | | | | | |

The robustness and imperceptibility of the zero watermarking technique in the case of applying DCT are experimented with to show the difference between the two cases and give improved outcomes. It is shown that most of the NC values are improved. Table 6 shows the PSNR and the NC outcomes of the images subjected to some common attacks and extracted the watermark image after performing the attacks. While Table 7 manifestation the watermark image extracted after images have been attacked.

**Table 6:** The NC and PSNR values for images employing HDM with DCT

| Attacks | Lena | | Girl | | Mandrill | | Peppers | |
|---|---|---|---|---|---|---|---|---|
| | *PSNR* | *NC* | *PSNR* | *NC* | *PSNR* | *NC* | *PSNR* | *NC* |
| Salt&Pepper noise1% | 25.4779 | 0.871279 | 24.9784 | 0.843069 | 25.5722 | 0.88448 | 25.3765 | 0.885346 |
| Jpeg_Compression | 36.0368 | 0.932382 | 44.1647 | 0.912926 | 29.6229 | 0.940819 | 34.8115 | 0.937644 |
| Imadjust | 19.2416 | 0.94909 | 18.9837 | 0.940235 | 17.9147 | 0.953633 | 18.3925 | 0.948795 |
| Wiener | 36.9599 | 0.935886 | 43.4714 | 0.924138 | 29.4818 | 0.91316 | 35.9493 | 0.934942 |
| Ordfilter | 28.2566 | 0.880767 | 33.9565 | 0.877098 | 20.6762 | 0.858957 | 28.0312 | 0.876132 |
| Histequalization | 17.2495 | 0.907868 | 10.5599 | 0.874285 | 16.2737 | 0.905754 | 17.9648 | 0.884557 |
| GaussianLow Pass Filter | 31.9246 | 0.91629 | 38.8328 | 0.905758 | 24.8087 | 0.88321 | 32.1989 | 0.912406 |
| Motion Filter | 26.0047 | 0.856253 | 29.1194 | 0.821601 | 21.2038 | 0.811593 | 25.8218 | 0.841641 |
| AverageFilter | 31.9246 | 0.91629 | 38.8328 | 0.905758 | 24.8087 | 0.88321 | 32.1989 | 0.912406 |
| MedianFiltering(2 × 2) | 30.3126 | 0.916788 | 35.7099 | 0.911211 | 24.4649 | 0.887219 | 30.2598 | 0.909475 |
| Disk Filter | 32.6027 | 0.943224 | 41.2078 | 0.952421 | 23.8729 | 0.903539 | 32.7611 | 0.946431 |

**Table 7:** The extracted binary watermark from the attacked images employing HDM with DCT

| Attacks | Lena | Girl | Mandrill | Peppers | Attacks | Lena | Girl | Mandrill | Peppers |
|---|---|---|---|---|---|---|---|---|---|
| Salt&Pepper noise1% |  | | | | Gaussian Low Pass Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Jpeg_ Compression |  | | | | Motion Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Imadjust |  | | | | Average Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Wiener |  | | | | Median Filtering (2 × 2) |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Ordfilter |  | | | | Disk Filter |  | | | |
| Extracted watermark |  | | | | Extracted watermark |  | | | |
| Hist equalization |  | | | | | | | | |
| Extracted watermark |  | | | | | | | | |

## 7. Comparison

In this work, the HDM has been utilized to transform algebraically the Y channel, obtained using the system of equations of the YCbCr space, into the algebraic frequency domain to generate the master-secret with and without DCT. First, the results obtained were acceptable according to the NC values. Second, a lot of the NC outcomes are improved using the DCT. Comparing the outcomes of the NC in both cases after implementing several attacks on the images, it has been observed that the values of the NC for the salt and pepper attack in the first case are better than the values of the NC in the second one. But the values of the

NC in the rest of the attacks for the second case are better than the values of the NC for the first case in general. Table 8 compares the NC values using the HDM with and without transforming.

**Table 8:** The comparison between the NC values of the two proposed techniques

| Attacks | NC values for images after attacks for the suggested technique of HDM without DCT | | | | NC values for images after attacks for the suggested technique of HDM with DCT | | | |
|---|---|---|---|---|---|---|---|---|
| | *Lena* | *Girl* | *Mandrill* | *Peppers* | *Lena* | *Girl* | *Mandrill* | *Peppers* |
| Salt&Pepper noise1% | 0.908338 | 0.90729 | 0.914883 | 0.909106 | 0.871279 | 0.843069 | 0.88448 | 0.885346 |
| Jpeg_ Compression | 0.773203 | 0.781181 | 0.789459 | 0.769885 | 0.932382 | 0.912926 | 0.940819 | 0.937644 |
| Imadjust | 0.910365 | 0.857076 | 0.946341 | 0.936073 | 0.94909 | 0.940235 | 0.953633 | 0.948795 |
| Wiener | 0.792364 | 0.79121 | 0.804427 | 0.78673 | 0.935886 | 0.924138 | 0.91316 | 0.934942 |
| Ordfilter | 0.778637 | 0.79198 | 0.777319 | 0.776405 | 0.880767 | 0.877098 | 0.858957 | 0.876132 |
| Hist equalization | 0.841091 | 0.816066 | 0.881157 | 0.853169 | 0.907868 | 0.874285 | 0.905754 | 0.884557 |
| Gaussian Low Pass Filter | 0.784098 | 0.789562 | 0.786007 | 0.780348 | 0.91629 | 0.905758 | 0.88321 | 0.912406 |
| Motion Filter | 0.767995 | 0.770087 | 0.766485 | 0.767163 | 0.856253 | 0.821601 | 0.811593 | 0.841641 |
| Average Filter | 0.784098 | 0.789562 | 0.786007 | 0.780348 | 0.91629 | 0.905758 | 0.88321 | 0.912406 |
| MedianFiltering (2 × 2) | 0.78318 | 0.790527 | 0.784247 | 0.779636 | 0.916788 | 0.911211 | 0.887219 | 0.909475 |
| Disk Filter | 0.7731 | 0.784994 | 0.773103 | 0.769124 | 0.943224 | 0.952421 | 0.903539 | 0.946431 |

**Table 9:** The comparison between the NC values of the proposed techniques and the technique given in [15] for Lena's image

| Technique | Attacks | | |
|---|---|---|---|
| | *Salt&Pepper noise1%* | *Median Filtering (2 × 2)* | *Average Filter* |
| The First Proposed Zero Watermarking Tech.(HDM) | 0.908338 | 0.78318 | 0.784098 |
| The Second Proposed Zero Watermarking Tech. (HDM+DCT) | 0.871279 | 0.916788 | 0.91629 |
| The Watermarking Tech. + HDM + ContourletTransform [15] | 0.9663 | 0.8907 | 0.9059 |

**Table 10:** The comparison between the NC values of the proposed techniques and the techniques given in [16] for the Histequalization attack

| Technique | Peppers Image | Lena Image | Mandrill Image |
|---|---|---|---|
| The First Proposed Zero Watermarking Tech. (HDM) | 0.853169 | 0.841091 | 0.881157 |
| The Second Proposed Zero Watermarking Tech. (HDM+DCT) | 0.884557 | 0.907868 | 0.905754 |
| The Scheme-I of the Zero Watermarking Tech. + DWT+SVD [16] | 0.9788 | 0.9819 | 0.9920 |
| The Scheme-II of the Zero Watermarking Tech. + DWT+SVD [16] | 0.9478 | 0.9909 | 0.8747 |

**Table 11:** The comparison between the NC values of the proposed techniques and the technique given in [17] for Lena image and Salt&Pepper noise1% attack

| Technique | NC |
|---|---|
| The First Proposed Zero Watermarking Tech. (HDM) | 0.908338 |
| The Second Proposed Zero Watermarking Tech. (HDM+DCT) | 0.871279 |
| The Zero watermarking Tech. + Quaternion Singular ValueDecomposition [17] | 0.9766 |

Many papers considering watermarking and zero watermarking techniques have utilized different methods to extract features from the host images. These methods vary between employing different transformations, algebraic decompositions, or transformations with the algebraic methods. Until now, no existing zero watermarking techniques have used the algebraic Hessenberg decomposition method. In the following three tables, the comparison depends on the common decomposition or the common image or the common attack between the two proposed techniques and the techniques given in [15- 16- 17].

## 8. Conclusion

In this paper, zero watermarking techniques are suggested using the algebraic Hessenberg decomposition method (HDM) for the first time. The first suggested technique relied on HDM without employing any transform. The second technique relied on HDM and the DCT to extract the matrix of features from the Y channel obtained from applying the YCbCr space on the host color images of size $512 \times 512$ to generate the master-secret that was preserved with a third party to save the ownership rights from manipulating. The employing of the HDM is successful, not only algebraically, but the strength and importance of the obtained outcomes are equivalent to the best outcomes obtained from employing any common transform that extracts the image features optimally. Furthermore, based on the outcomes of the NC values obtained from the two methods mentioned above, it is found that the NC values improved with the employment of DCT. The suggested technique gave good robustness contra various kinds of popular attacks.

### Author contribution

All authors contributed equally to this work.

### Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

### Conflicts of interest

The authors declare that there is no conflict of interest.

## References

**[1]** M. Allali, Linear algebra and image processing, Int. J. Math. Educ. Sci. Technol., 41 (2010) 725–741. https://doi.org/10.1080/00207391003675133

**[2]** D. Fan, Y. Li, S. Gao, W. Chi, C. Lv, A Novel Zero Watermark Optimization Algorithm Based On Gabor Transform And Discrete Cosine Transform, Concurr. Comput. Pract. Exp., 34 (2020) 1-13.

**[3]** Q. Su, Novel blind color image watermarking technique using Hessenberg decomposition, IET Image. Proc., 10 (2016) 817-829.

**[4]** O. Abodena, M. Agoyi, Colour Image Blind Watermarking Scheme Based on Fast Walsh Hadamard Transform and Hessenberg Decomposition, Stud. Inform. Control., 27 (2018) 339-348. https://doi.org/10.24846/v27i3y201809

**[5]** J. Liu, J. Huang, Y. Luo, L. Cao, S. Yang, D. Wei, R. Zhou, An Optimized Image Watermarking Method Based on HD and SVD in DWT Domain, IEEE Access, 7 (2019) 80849-80860. https://doi.org/10.1109/ACCESS.2019.2915596

**[6]** A. M. Abduldaim, J. Waleed, A. N. Mazher, An Efficient Scheme of Digital ImageWatermarking Based on Hessenberg Factorization and DWT, Int. Conf. Comput. Sci. Soft. Eng., (2020) 180-185. https://doi.org/10.1109/CSASE48920.2020.9142096

**[7]** R. I. Sabri, A. M. Abduldaim, J.Waleed, Mamdani FIS Combined With LU Decomposition Method and Two-Level LWT For Image Watermarking Technique, Int. Conf. Eng. Technol. Appl., (2020) 12-17. https://doi.org/10.1109/IICETA50496.2020.9318829

**[8]** Y. Zhao, J. Li, Z. Zhou, Multipurpose Zero Watermarking Algorithm for Color Image Based on SVD and DBN, Boletín Técnico., 55 (2017) 21-32.

**[9]** X. Liu, B. Chen, G. Coatrieux, H. Shu, Color Image Zero-Watermarking Based on SVD and Visual Cryptography in DWT Domain, Eighth Int. Conf. Graphic . Image Proc., (2017)10225. https://doi.org/10.1117/12.2267045

**[10]** J. Waleed, A. M. Abduldaim, H. H. Alyas, A. Q. Mohammed, An Optimized Zero-Watermarking Technique Based On SFL Algorithm, Int. Conf. Electr. Commun. Comp. Power .Cont. Eng., (2019) 171-175. https://doi.org/10.1109/ICECCPCE46549.2019.203768

**[11]** G.H. Golub, G.H. Loan, C.F. V. Matrix Computations, John Hopkins university press, Baltimore, MD, 1983.

**[12]** N. K. El Abbadi, E. Saleem, Auto Colorization of Gray-Scale Image Using Ycbcr Color Space, Iraqi J.Sci., 61 (2020) 3379-3386. https://doi.org/10.24996/ijs.2020.61.12.26

**[13]** Hamilton, E. JPEG File Interchange Format Version 1.02, California, USA, September 1, 1992.

**[14]** A. Nasir. How I Came Up With the Discrete Cosine Transform, Digital Signal Process., 1(1991) 4–5.

**[15]** Q. Su, G. Wang, G. Lv, X. Zhang, G. Deng, B. Chen, A novel blind color image watermarking based on Contourlet transform and Hessenberg decomposition, Multimed. Tools. Appl., 76 (2017) 8781–8801. https://doi.org/10.1007/s11042-016-3522-z

**[16]** A. Rani, A. K. Bhullar, D. Dangwal, S. Kumar, A Zero-Watermarking Scheme using Discrete Wavelet Transform, Proc.Comput. Sci., 70 (2015) 603-609. https://doi.org/10.1016/j.procs.2015.10.046

**[17]** F. Liu, L.Ma, C. Liu, Zero Watermarking Scheme Based on U and V Matrices of Quaternion Singular Value Decomposition for Color Images, J. Inf. Hiding. Multimed. Sig. Proc., 9 (2018) 629-640