Robust 2D and 3D images zero-watermarking using dual Hahn moment invariants and Sine Cosine Algorithm

Achraf Daoui 1 • Hicham Karmouni 2 • Mhamed Sayyouri 1 • Hassan Qjidaa 2

Received: 6 September 2020 / Revised: 25 June 2021 / Accepted: 14 January 2022 / Published online: 23 March 2022
© Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In this paper, we initially provide significant improvements on the computational aspects of dual Hahn moment invariants (DHMIs) in both 2D and 3D domains. These improvements ensure the numerical stability of DHMIs for large-size images. Then, we propose an efficient method for optimizing the local parameters of dual Hahn polynomials (DHPs) when computing DHMIs using the Sine-Cosine Algorithm (SCA). DHMIs optimized via SCA are used to propose new and robust zero-watermarking scheme applied to both 2D and 3D images. On one hand, the simulation results confirm the efficiency of the proposed computation of 2D and 3D DHMIs regarding the numerical stability and invariability. Indeed, the proposed computation method of 2D DHMIs allows to reach a relative error (RE) of the order $\approx 10^{-10}$ for images of size $1024 \times 1024$ with an execution time improvement ratio exceeds 70% ($ETIR \geq 70\%$), which validates the efficiency of the proposed computation method. On the other hand, the simulation and comparison outcomes clearly demonstrate the robustness of the proposed zero-watermarking scheme against various geometric attacks (translation, rotation, scaling and combined transformations), as well as against other common 2D and 3D image processing attacks (compression, filtering, noise addition, etc.).

Keywords
Image reconstruction · 2D and 3D image zero-watermarking · Discrete orthogonal moments · Dual-Hahn moment invariants · Sine cosine algorithm · Numerical stability

* Achraf Daoui
achraf.daoui@usmba.ac.ma

Hicham Karmouni
hicham.karmouni@usmba.ac.ma

Mhamed Sayyouri
mhamed.sayyouri@usmba.ac.ma

Hassan Qjidaa
qjidah@yahoo.fr

Extended author information available on the last page of the article
1 Introduction

Recently, the traditional lifestyle of people changed due to the rapid development of advanced communication technology, big data technology, cloud computing and cloud storage technology. With this technology, the transmission of data (images, videos, digital documents, etc.) has become more rapid and more efficient. Although the transmission of data is not always secure, researchers have developed techniques for protecting the intellectual property rights. These techniques include watermarking and zero-watermarking algorithms. Typically, watermarking algorithms involve modifying the original image (the host image) by embedding a message called the watermark. Watermarking algorithms can be implemented in both spatial and frequency domains. A watermarking algorithm in the spatial domain integrates the watermark by modifying the pixel values of the host image. Whereas, a frequency-domain watermarking algorithm integrates the watermark by modifying the spectral coefficients of the transformed host image. However, the embedded watermark in spatial or frequency domain may destroy relevant information within the host image, particularly in the case of a medical image that contains sensitive pathological information. Therefore, the need for lossless copyright protection technology is extremely important. Indeed, Wen et al. [44] have introduced for the first time the concept of the image zero-watermarking, which is theoretically based on the principle that each image presents its own characteristics that are different from the characteristics of any other one. Imperceptibility, i.e. the fact that the host image remains unchanged, is one of the main advantages of the zero watermarking techniques. Research on zero-watermarking methods has made remarkable progress in recent years. Most of these methods have been reported in the field of 2D images [14, 23, 24, 45–47, 50, 51] and 1D signals [1, 8]. However, only few zero-watermarking schemes was conducted in the area of 3D images [14, 16, 32, 42, 48]. The latter are becoming more and more used, especially in the medical field. For this reason, the development of efficient 3D images zero-watermarking methods is extremely pertinent. Moreover, it is crucial that researchers further improve the robustness of these techniques against different types of attacks. However, extracting robust and stable features from 3D images for designing efficient zero-watermarking schemes of 3D images is challenging task. In addition, the representations and transformations in the 3D domain are more complex and more varied in comparison to the 2D domain. The geometric transformations such as rotation, translation, scaling and combinations of these transformations can easily lead to the inability to recover a watermark from an attacked (transformed) image. Thus, when designing a new zero-watermarking scheme, special attention should be paid to the robustness of the presented scheme against both geometric and common image processing attacks.

Image moment invariants are considered as powerful image features since they are resistant to different types of image attacks. Therefore, they are used in various applications such as pattern recognition [2, 3], image classification [3, 29], image watermarking [49], and 2D image zero-watermarking [14, 24, 45]. These applications are mainly based on the invariance property of the extracted image feature vector with respect to various image transformations (geometric, noise adding, filtering, etc.).

Moment invariants including Legendre [2], Legendre-Fourier [21], Zernike [52], Fourier-Mellin [35], and Laguerre moments [17]. These are based on continuous orthogonal polynomials basis that require the discretization of the continuous interval to discrete summation before they can be applied in digital image processing. As a result, digital errors are introduced during the image feature extraction [18]. To overcome this problem, researchers are currently
substituting continuous orthogonal polynomials basis kernels by discrete ones such as Tchebichef [6, 9, 10, 26], Krawtchouk [11, 26, 27], Hahn [9, 38, 40], Meixner [12, 22, 28, 29], Charlier [13, 15, 19, 39], Dual Hahn [25, 53] and Racah polynomials [4, 54]. The latter are defined on discrete intervals, and therefore no approximation or discretization is involved when using these polynomials in digital signal and image analysis.

Dual Hahn moment invariants (DHMI)s are defined from Dual Hahn polynomials (DHPs) and geometric moment invariants. DHMIs more general compared to the other types of moment invariants (with the exception of Racah moment invariants). Indeed, the discrete orthogonal Tchebichef, Meixner and Krawtchouk polynomials are derived from Dual Hahn polynomials (DHPs) by adapting the local parameters \((a, b)\) of DHPs to specific cases [30]. For this reason, DHMIs are considered in the present work to propose a new and robust 2D and 3D image zero - watermarking scheme. The main advantages of the proposed scheme are: 1) the robustness against geometric and common image processing attacks, 2) numerically stable and suitable for large-size images, 3) it is a standard scheme valid for both 2D and 3D images, and 4) it offers a high level of security and robustness against various attacks. Moreover, to the best of our knowledge, this is the first time that the 3D moment invariants are used in the zero - watermarking application. The proposed scheme is mainly based on the computation of the 2D/3D DHMIs of the 2D/3D image. Next, the computed DHMIs with a binary logo (watermark) are used to generate the zero-watermark image. The latter is then used when verifying the copyrights of a transmitted image. This work also presents considerable improvements concerning the computation of both 2D and 3D DHMIs. Thanks to these improvements, the numerical stability of DHMIs is considerably improved through the elimination of all gamma and factorial functions that are required in the conventional computation of DHMIs. Moreover, the issue of the optimal selection of local DHPs parameter values when computing DHMIs is effectively addressed by using a recent population-based optimization algorithm, namely the Sine Cosine Algorithm (SCA).

The rest of the paper is organized as follows: the second section presents the theoretical background of DHPs used in DHMIs computation. In the third section, we present the computation method of both 2D and 3D DHMIs, as well as the proposed algorithms of DHMIs computation. The fourth section presents the proposed method for the optimal selection of the local DHPs parameter values. In the fifth section, we present the proposed zero - watermarking scheme based on DHMIs and SCA for 2D/3D image copyrights protection. Then, in the sixth section, the simulation and comparison results and are presented to prove the efficiency of the proposed computation of DHMIs, as well as to demonstrate the robustness of our zero-watermarking scheme compared to recent existing ones. The last section presents the conclusion and future work.

### 1.1 Discrete orthogonal dual Hahn polynomials

DHPs of order \(n\) and discrete variable \(s\) are defined by [53]:

\[
H_n^{(a,b,c)}(s) = \frac{(a - b + 1)_n(a + c + 1)_n}{n!} _3F_2\left(-n, a - s, a + s + 1; a - b + 1, a + c + 1; 1\right)
\]

(1)

where \(n = 0, 1, 2, \ldots, N - 1\), \(s = a, a + 1, \ldots, b - 1\), \(-1/2 < a < b, |c| < a + 1, b = a + N\), and \(_3F_2(.)\) is a generalized hypergeometric function given by:
The recursive relation that allows to compute \( F_k \) as follows:

\[
3F_2(a_1, a_2, a_3, b_1, b_2, z) = \sum_{k=0}^{\infty} \frac{(a_1)_k(a_2)_k(a_3)_k}{(b_1)_k(b_2)_k} \frac{z^k}{k!}
\] (2)

The notation \((a)_k\) being the Pochhammer symbol that is defined from the gamma function \(\Gamma(.)\) as follows:

\[
(a)_k = a(a+1)(a+2)(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)}
\] (3)

By using Eqs. (2)–(3) and the relation \((-n)_k = (-1)^k(n)_k\), we can express Eq. (1) as follows:

\[
H_n^{(a,b,c)}(s) = \frac{(a-b+1)_n(a+c+1)_n}{n!} \sum_{k=0}^{n} \frac{(n)_k(s-a)_k(s+a+1)_k}{(a-b+1)_k(a+c+1)_k k!} \] (4)

where

\[
A_n^{(a,b,c)} = \frac{(a-b+1)_n(a+c+1)_n}{n!}
\] (5)

\[
B_{nk}^{(a,b,c)} = \frac{(n)_k}{(a-b+1)_k(a+c+1)_k k!}
\] (6)

\[
F_k^{(a)}(s) = (s-a)_k(s+a+1)_k
\] (7)

The symbol \((a)_k\) is the filling factorial defined as:

\[
\langle a \rangle_k = \begin{cases} 
  a(a-1)(a-2)(a-k+1) = \frac{\Gamma(a+k)}{\Gamma(a-k+1)} k \geq 0, \\ 
  1 \quad k = 0
\end{cases}
\] (8)

We notice that the expressions of the functions given by Eqs. (4)–(7), depend on the factorial and gamma functions. The presence of the latter causes the problem of the numerical overflow of \(A_n^{(a, b, c)}, B_{nk}^{(a, b, c)}\) and \(F_k^{(a)}(s)\) values. Noting the following example of the overflow problem caused by the gamma function using Matlab: \(\Gamma(172) = Inf\). To avoid this problem, \(A_n^{(a, b, c)}, B_{nk}^{(a, b, c)}\) and \(F_k^{(a)}(s)\) terms will be expressed independently of gamma and factorial functions. Indeed, to compute \(A_n^{(a, b, c)}\), the following recursive formula is developed:

\[
A_n^{(a,b,c)} = \begin{cases} 
  \frac{(a-b+n)(a+c+n)}{n} A_{n-1}^{(a,b,c)} & n > 0 \\ 
  1 & n = 0
\end{cases}
\] (9)

The recursive relation that allows to compute \(B_{nk}^{(a,b,c)}\) function is given by [25]:

\[
B_{nk}^{(a,b,c)} = \begin{cases} 
  \frac{(n-k+1)}{(a-b+k)(a+c+k)k} B_{n(k-1)}^{(a,b)} & 0 < k \leq n \\ 
  1 & k = 0
\end{cases}
\] (10)

To recursively compute the function \(F_k^{(a)}(s)\), we use the following formula:

\[
F_k^{(a)}(s) = \begin{cases} 
  (s+a+k)(s-a-k+1) F_{k-1}^{(a)}(s) & 0 < k \leq n \\ 
  1 & k = 0
\end{cases}
\] (11)
For $k = 1$, let’s set $x = s - a$ and $a_1 = 2a + 1$. Then the expression of $F_k(s)$ becomes as follows:

$$F_1(a)(x) = (x + a_1)x = x^2 + a_1x + 0x^0$$

$$F_1(x, a) = \sum_{r=0}^{2} C_{1r}x^r$$ (12)

Using Eq. (11) for $k = 2$ with $a_2 = 2a + 2$, the following equation can be obtained:

$$F_2(a)(x) = (x + a_2)(x - 1)F_1(x)$$

$$F_2(x, a) = (x + a_2)(x - 1)(x^2 + a_1x + 0x^0)$$

$$F_2(x, a) = x^3 + (a_1 + a_2 - 2)x^2 - a_1a_2x + 0x^0$$

$$F_2(x, a) = \sum_{r=0}^{2} C_{2r}x^r$$ (13)

In the general case for $k \geq 1$ and $a_k = 2a + k$, we show that $F_k(x, a)$ can be expressed as follows:

$$F_k(a)(x) = (x + a_k)(x + k)F_{k-1}(x)$$

$$F_k(x, a) = \sum_{r=0}^{2} C_{kr}x^r$$ (14)

where $C_{kr}$ is defined as follows [25]:

$$C_{kr} = \begin{cases} C_{(k-1)(r-2)} + [a_k-(k-1)] \\ \times C_{(k-1)(r-2)-(k-1)a_k C_{(k-1)r}} & \forall k \geq 2, 2 \leq r \leq 2k - 2 \\
C_{(k-1)(2k-3)} + [a_k-(k-1)] \\ - C_{(k-1)(2k-3)} + [a_k-(k-1)]a_k C_{(k-1)r} & \forall k \geq 2, r = 2k - 2 \\
1 & \forall k \geq 2, r = 1 \\
0 & \forall k \geq 2, r = 0 \\
\end{cases}$$ (15)

where $C_{00} = 1$, $C_{10} = 0$, $C_{11} = a_1$, $C_{12} = 1$ with $a_n = 2a + n$.

Using Eqs. (4), (9)–(10) and (14), DHPs can be expressed as follows [25]:

$$H_n(a,b,c)(s) = A_n(a,b,c) \sum_{k=0}^{n} B_{nk}(a,b,c) \sum_{r=0}^{2} C_{kr}x^r$$ (16)

The DHPs satisfy an orthogonal relation of the following form [53]:

$$\sum_{s=a}^{b-1} H_n(a,b,c)(s)H_m(a,b,c) \rho(s)[\Delta x(s-1/2)] = \delta_{nm}d_n^2, \quad n, m = 0, 1, 2, 3, \ldots, N-1$$ (17)

where $\Delta x(s - 1/2) = 2s + 1$ for $s > -1/2$. $\rho(s)$ and $d_n^2$ are successively the weight and the square norm functions that are defined by Eqs. (18)–(19), respectively.

$$\rho(s) = \frac{\Gamma(a + s + 1)\Gamma(c + s + 1)}{\Gamma(s-a+1)\Gamma(b-s)\Gamma(b+s+1)\Gamma(s-c+1)} \quad \text{with} \quad (-1/2 \leq a \leq b-1, |c| < a + 1)$$ (18)

$$d_n^2 = \frac{\Gamma(a + c + n + 1)}{n!(b - a - n - 1)!\Gamma(b - c - n)}$$ (19)

Noting that $\rho(s)$ and $d_n^2$ functions depend on the gamma function, which provokes the numerical instability of the computed DHPs. We can avoid this problem by using the recursive
computation of $\rho(s)$ and $d^2_n$ functions. Indeed, we show that $\rho(s)$ can be expressed as:

$$\rho(s) = \begin{cases} \frac{\Gamma(2a+1)\Gamma(c+a+1)}{\Gamma(b-a)\Gamma(b+a+1)\Gamma(a-c+1)}; & s = a, \\ \frac{(a+s)(b-s)(c+s)}{(s-a)(b+s)(s-c)}\rho(s-1); & s = a + 1, a + 2, \ldots, b \end{cases}$$

(20)

We notice that the first term $\rho(a)$ depends on $\Gamma(.)$. Therefore, if the values of $a, b, c$ are large enough, then this term becomes numerically unstable. In order to overcome this problem, $\rho(a)$ is computed based on $\log\Gamma(x)$ function as follows:

$$\rho(a) = e^{a-B\log A} = \log\Gamma(2a+1) + \log\Gamma(c+a+1)$$

and $B = \log\Gamma(b-a) + \log\Gamma(b+a+1) + \log\Gamma(a-c+1)$

(21)

For $d^2_n$ function is recursively computed with respect to the order $n$ as follows:

$$d^2_i = \begin{cases} \frac{\Gamma(a+c+1)}{\Gamma(b-a)\Gamma(b-c)}; & n = 0 \\ \frac{(a+c+i)(b-c-i)(b-a-i)}{i}d^2_{i-1}; & i = 1, 2, \ldots, n \end{cases}$$

(22)

The term $d^2_0$ is also computed based on $\log\Gamma(x)$ function as follows in order to avoid the numerical instability problem:

$$d^2_0 = e^{\log\Gamma(a+c+1)-\log\Gamma(b-a)-\log\Gamma(b-c)}$$

(23)

We notice that the expressions of $A_n(a, b, c)$, $B_n(a, b, c)$, $F_n(a)(s)$, $\rho(s)$ and $d^2_n$ functions become independent of all gamma and factorial functions thanks to the proposed computation method. Consequently, the numerical stability of these functions is ensured when the values of $n, s, a, b$ and $c$ are high. As a result, the computation DHMIs of large-size images becomes possible.

The orthonormalized DHPs are computed by using the square norm and the weight functions as follows [53]:

$$\hat{H}_n^{(a,b,c)}(s) = H_n^{(a,b,c)}(s)\sqrt{\frac{\rho(s)}{d^2_n}}\left[\Delta x \left( s - \frac{1}{2} \right) \right]$$

(24)

For more details on the properties of the weighted DHPs and its recursive computation, the reader can consult the reference [53]. Figure 1 shows the orthonormalized DHPs computed up to the fifth order for different values of the parameters $a, b$ and $c$.

From Fig. 1 we notice that the values of the local parameters $a, b$ and $c$ influence on the distribution of DHPs values, which means that these parameters can be used for the extraction of the characteristics of an image region of interest (ROI).

### 1.2 2D and 3D dual Hahn moments and moment invariants

In this section, we briefly present the theoretical background of dual Hahn moments (DHMs) and the method of constructing dual Hahn moment invariants (DHMIs) in both 2D and 3D domains. In addition, the proposed improved algorithms for fast and stable computation of DHMIs are presented. Noting that in the rest of this article, we set $x = s - a$ and $y = t - a$ with $s, t \in [a, b - 1]$ to compute DHMs and DHMIs of 2D image $f(x, y)$. Furthermore, we set $z = w - a$ and $w \in [a, b - 1]$ to compute 3D DHMs and DHMIs of 3D image $f(x, y, z)$.
3.1 2D Dual Hahn Moments.

For a 2D image $f(x, y)$ of size $N \times M$, the DHMs of order $(n, m)$ are computed by [53]:

$$HM_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} H_n^{(a,b,c)}(x) \times H_m^{(a,b,c)}(y) \times f(x,y)$$

(25)

with $n = 0, 1, 2, \ldots, N_{\text{max}} \leq N - 1, m = 0, 1, 2, \ldots, M_{\text{max}} \leq M - 1$.

The computation time of DHMs is considerably reduced when using the following matrix formula [40]:

$$HM = H_n^T \times f \times H_m$$

(26)

where $f = \{f(x,y)\}_{x,y=0}^{N-1, M-1}$ denotes the 2D image of size $N \times M$, $H_n$ and $H_m$ are DHPs matrices defined by: $H_n = \{H_n^{(a,b,c)}(x)\}_{n=0,x=0}^{N_{\text{max}},x=0}$ and $H_m = \{H_m^{(a,b,c)}(y)\}_{m=0,y=0}^{M_{\text{max}},y=0}$.

The inverse transformation of the DHMs allows to obtain the reconstructed image through the following formula:

$$\hat{f}(x,y) = \sum_{n=0}^{N_{\text{max}}} \sum_{m=0}^{M_{\text{max}}} H_n^{(a,b,c)}(x) \times H_m^{(a,b,c)}(y) \times HM_{nm}$$

(27)

The matrix representation of Eq. (27) is given by:

$$\hat{f} = H_n \times HM \times H_m^T$$

(28)

where $\hat{f} = \{\hat{f}(x,y)\}_{x,y=0}^{N-1, M-1}$ being the reconstructed image.
1.3 2D dual Hahn moment invariants

This subsection presents the process used to derive DHMIs of a 2D image based on the geometric moment invariants (GMIs) and DHPs.

The geometric moments (GMs) of order \((n, m)\) are defined for a 2D image function \(f(x, y)\) of size \(N \times M\) by [25]:

\[
GM_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^n y^m f(x, y) \tag{29}
\]

In order to reduce considerably the computation time of GMs, we use the following matrix formulation:

\[
GM_{nm} = X^T F Y \quad \text{with} \quad X = \{x^n\}_{x=0}^{N-1}, y=0^{N \times N_{\text{max}}}, \quad Y = \{y^m\}_{y=0}^{M-1}, m=0^{M \times M_{\text{max}}} \quad \text{and} \quad F = \{f(x, y)\}_{x=0}^{N-1}, y=0^{M-1}
\]

The set of geometric invariant moments (GIMs) of order \((n, m)\), denoted \(V_{nm}\), which are independent of translation, scaling and rotation are defined by the following relation [23]:

\[
V_{nm} = GM_{00}^{-\gamma} \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \left\{ \left[ (x - \bar{x}) \cos \theta + (y - \bar{y}) \sin \theta \right]^i \times \left[ (y - \bar{y}) \cos \theta - (x - \bar{x}) \sin \theta \right]^j \right\} f(x, y) \tag{33}
\]

where \(\gamma = \frac{n+m}{2} + 1\) and \(\theta = \frac{1}{2} \tan^{-1} \left( \frac{2 \mu_{11}}{\mu_{20} - \mu_{02}} \right)\).

In accordance with Eq. (33), the values of the angle \(\theta\) are limited to \(-\pi/4 \leq \theta \leq \pi/4\). We refer the reader to see [41] for angle values in different ranges.

The computation time of \(V_{nm}\) by Eq. (33) increases dramatically when the image size \((N \times M)\) becomes too large. In order to accelerate considerably the computation time of \(V_{nm}\), we propose the use of the following semi-matrix formulation:

\[
V_{nm} = GM_{00}^{-\gamma} \sum_{i=0}^{n} \sum_{j=0}^{m} (A^i \odot B^j) \odot F \tag{34}
\]

where \(A = (x - \bar{x}) \cos \theta + (y - \bar{y}) \sin \theta\), \(B = (y - \bar{y}) \cos \theta - (x - \bar{x}) \sin \theta\), \(F = f(x, y)\), \(x = 1, 2, \ldots, N, y = 1, 2, \ldots, M, \bar{x} = \frac{GM_{nx}}{GM_{0n}}, \bar{y} = \frac{GM_{ny}}{GM_{00}}\) and \(\odot\) denotes the Hadamard product [5].
The DHPs formula that is given by Eq. (4) allows to express DHMs (Eq. (27)) as follows [25]:

\[ HM_{nm} = \frac{1}{\sqrt{d_n^2 d_m^2}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} H_n^{(a,b,c)}(x) H_m^{(a,b,c)}(y)f(x,y) \]  

(35)

where \( f(x,y) = \sqrt{\rho(x)\rho(y)}f(x,y) \)

Using Eqs. (16) and (35), the DHMs are represented in terms of GMs as follows:

\[ HM_{nm} = \frac{1}{\sqrt{d_n^2 d_m^2}} \sum_{i=0}^{n} \sum_{j=0}^{m} B_{in}^{(a,b,c)} B_{jm}^{(a,b,c)} \sum_{k=0}^{2i} \sum_{l=0}^{2j} C_{ik} C_{jl} GM_{kl} \]  

(36)

Finally, to compute 2D DHMIs up to the order \((n, m)\), the expression of GMs in Eq. (36) is substituted by GMIs expression \((V_{nm})\) as follows [25]:

\[ HMI_{nm} = \frac{1}{\sqrt{d_n^2 d_m^2}} \sum_{i=0}^{n} \sum_{j=0}^{m} B_{in}^{(a,b,c)} B_{jm}^{(a,b,c)} \sum_{k=0}^{2i} \sum_{l=0}^{2j} C_{ik} C_{jl} V_{kl} \]  

(37)

The conventional computation of 2D image DHMIs is summarized in Algorithm 1, and Algorithm 2 presents the proposed fast and stable computation of DHMIs based on the recursive formulas of \(\rho(s), d_n^2, A_n^{(a,b,c)}\) and \(B_n^{(a,b,c)}\) functions, and on the matrix formulations.

| Inputs | Outputs | Step 1 | Step 2 | Step 3 | Step 4 | Step 5 | Step 6 | Step 7 |
|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| \(\tilde{f}(x,y)\): the normalized image of size \(N \times M\); \((n,m)\): DHMs order; \(a,b\) and \(c\): DHPs parameter values. | \(HMI_{nm}\) the 2D image DHMIs of order \((n, m)\). | Compute \(A_n^{(a,b,c)}\) using Eq. (5) | Compute \(B_k^{(a,b,c)}\) using Eq. (6) | Compute \(d_n^2\) using Eq. (19) | Compute \(C(k,i)\) using Eq. (15) | Compute \(GM_{nm}\) using Eq. (36) | Compute \(V_{nm}\) using Eq. (33) | Compute \(HMI_{nm}\) of \((n,m)\) - \(th\) order using Eq. (37). |

| Inputs | Outputs | Step 1 | Step 2 | Step 3 | Step 4 | Step 5 | Step 6 | Step 7 |
|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| \(\tilde{f}(x,y)\): normalized 2D image of size \(N \times M\); \((n,m)\): DHMs order; \(a,b\) and \(c\): DHPs parameter values. | \(HMI_{nm}\) the image DHMIs of order \((n, m)\) | Compute \(A_n^{(a,b,c)}\) using Eq. (9) | Compute \(B_k^{(a,b,c)}\) using Eq. (10) | Compute \(d_n^2\) using Eqs. (22) – (23) | Compute \(C(k,i)\) using Eq. (15) | Compute \(GM_{nm}\) using Eq. (30) | Compute \(V_{nm}\) using Eq. (34) | Compute \(HMI_{nm}\) of \((n,m)\) - \(th\) order using Eq. (37). |
1.4 3D dual Hahn moments

For a given 3D image $f(x, y, z)$ of size $N \times M \times K$, the DHMs of order $(n, m, k)$ are computed as follows [33]:

$$HM_{mnk} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} \bar{H}_{n}^{(a,b,c)}(x) \bar{H}_{m}^{(a,b,c)}(y) \bar{H}_{k}^{(a,b,c)}(z)f(x, y, z)$$  \hspace{1cm} (38)

The following relation is used for reconstruction a 3D image based on DHMs and DHPs:

$$\hat{f}(x, y, z) = \sum_{n=0}^{N_{max}} \sum_{m=0}^{M_{max}} \sum_{k=0}^{K_{max}} \bar{H}_{n}^{(a,b,c)}(x) \bar{H}_{m}^{(a,b,c)}(y) \bar{H}_{k}^{(a,b,c)}(z)HM_{mnk}$$  \hspace{1cm} (39)

To quantify the reconstruction errors between the reconstructed image $\hat{f}(x, y, z)$ and the original one $f(x, y, z)$, the Mean Square Error (MSE) criterion can be used. The MSE is defined as:

$$MSE = \frac{1}{N \times M \times K} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} (f(x, y, z) - \hat{f}(x, y, z))^2$$  \hspace{1cm} (40)

The Peak Signal-to-Noise Ratio (PSNR) in decibels (dB) is another criterion that is widely used to measure the reconstruction error. PSNR is defined as:

$$PSNR = 10\log_{10}\left(\frac{\text{peak}^2}{MSE}\right)$$  \hspace{1cm} (41)

with peak represents the maximum value of the image intensity matrix.

1.5 3D dual Hahn moments invariants

To derive 3D DHMIs, the 3D DHMs of the 3D image are expressed as a linear combination of 3D GMs. Then, the latter are replaced by the 3D GMIs.

The 3D GMs of a 3D image $f(x, y, z)$ of size $N \times M \times K$ are defined by [18]:

$$GM_{nmk} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} x^n y^m z^k f(x, y, z)$$  \hspace{1cm} (42)

To compute the central moments $\mu_{nmk}$, the following formula is used:

$$\mu_{nmk} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} (x - \bar{x})^n (y - \bar{y})^m (z - \bar{z})^k f(x, y, z)$$

with $\bar{x} = \frac{GM_{100}}{GM_{000}}$, $\bar{y} = \frac{GM_{010}}{GM_{000}}$ and $\bar{z} = \frac{GM_{001}}{GM_{000}}$  \hspace{1cm} (43)

where $(\bar{x}, \bar{y}, \bar{z})$ are the coordinates of the image centroid. Noting that $\mu_{nmk}$ are invariant with respect to translation.

The DHMs that are invariant with respect to the rotation in the 3D domain are defined from the 3D rotation matrix associated with the Euler angle sequence. This matrix is defined by [18]:

 Springer
\[ R(\phi, \theta, \psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \times \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \times \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \] (44)

where angle \( \phi \) is associated with rotation on the \( x \)-axis, angle \( \theta \) is associated with rotation on the \( y \)-axis, and angle \( \psi \) associated with rotation on the \( z \)-axis. It is possible to consider the 3D rotation matrix as a linear transformation of the 3D image coordinates, as follows:

\[
\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_{00}x + a_{01}y + a_{02}z \\ a_{10}x + a_{11}y + a_{12}z \\ a_{20}x + a_{21}y + a_{22}z \end{pmatrix}
\] (45)

where \( a_{00}, a_{01}, \ldots, a_{22} \) are coefficients of the 3D rotation matrix calculated from Eq. (45) as follows:

\[
\begin{align*}
a_{00} &= \cos\theta \cos\psi, \quad a_{01} = \cos\theta \sin\psi, \quad a_{02} = -\sin\theta \\
a_{10} &= \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi, \\
a_{11} &= \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi, \\
a_{12} &= \cos\theta \sin\phi, \\
a_{20} &= \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi, \\
a_{21} &= \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi, \\
a_{22} &= \cos\theta \cos\phi
\end{align*}
\] (46)

By analogy with GMIs extraction method in the 2D domain, we can derive the GMIs up to \( (n, m, k) \) order in the 3D domain. These are invariants with respect to rotation, translation and scaling. The 3D GMIs are calculated by the following relation [24]:

\[
GMI_{nmk} = GM_{000}^{-\gamma} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} \left[ a_{00}(x-x') + a_{01}(y-y') + a_{02}(z-z') \right]^n \times \left[ a_{10}(x-x') + a_{11}(y-y') + a_{12}(z-z') \right]^m \times \left[ a_{20}(x-x') + a_{21}(y-y') + a_{22}(z-z') \right]^k f(x, y, z)
\] (47)

where \( \gamma = \frac{n+m+k}{3} + 1 \), \( \theta = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{10}}{\mu_{20}-\mu_{02}} \right) \), \( \phi = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{01}}{\mu_{02}-\mu_{00}} \right) \) and \( \psi = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{10}}{\mu_{20}-\mu_{02}} \right) \).

For detailed information on the computation process of geometric moment invariants in the 3D domain, the reader can see [18].

The use of Eqs. (16) and (38), allows to express 3D DHMs as follows [33]:

\[
HM_{nmk} = \frac{1}{d_n^2 d_m^2 d_k^2} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} H_n^{(a,b,c)}(x) H_m^{(a,b,c)}(y) H_k^{(a,b,c)}(z) f(x, y, z)
\] (48)

where \( f(x, y, z) = \sqrt{\rho(x)\rho(y)\rho(z)} f(x, y, z) \).
Using Eqs. (3), (33) and (37), the 3D DHMs are expressed in terms of 3D GMs as follows:

\[
HM_{n,m,k} = \frac{1}{\sqrt{d_n^2 d_m^2 d_k^2}} \sum_{i=0}^{n} B_{ni}^{(a,b,c)} \sum_{e=0}^{i} C_{ie} \sum_{j=0}^{m} B_{mj}^{(a,b,c)} \sum_{f=0}^{j} C_{jf} \sum_{k=0}^{i} B_{kh}^{(a,b,c)} \sum_{g=0}^{k} C_{hg} GM_{efg}
\] (49)

Finally, to compute DHMIs up to the order \((n, m, k)\) in the 3D domain, we replace 3D GMs in Eq. (49) by 3D GMIs using the following formula [47]:

\[
HMI_{n,m,k} = \frac{1}{\sqrt{d_n^2 d_m^2 d_k^2}} \sum_{i=0}^{n} B_{ni}^{(a,b,c)} \sum_{e=0}^{i} C_{ie} \sum_{j=0}^{m} B_{mj}^{(a,b,c)} \sum_{f=0}^{j} C_{jf} \sum_{k=0}^{i} B_{kh}^{(a,b,c)} \sum_{g=0}^{k} C_{hg} GM_{efg}
\] (50)

The conventional computation of 3D DHMIs is summarized in algorithm 3, and the proposed improved computation is presented in algorithm 4.

**Inputs:** \( \tilde{f}(x, y, z) \) : normalized 3D image of size \( N \times M \times K \); \((n, m, k)\) : DHMIs order; \(a, b\) and \(c\) : DHPs parameter values

**Outputs:** \(HMI_{n,m,k}\) : DHMIs of order \((n, m, k)\)

**Step 1** Compute \(A_k^{(a,b,c)}\) using Eq. (5)

**Step 2** Compute \(B_k^{(a,b,c)}\) using Eq. (6)

**Step 3** Compute \(d_k^2\) using Eq. (19)

**Step 4** Compute \(C(k,i)\) using Eq. (15)

**Step 5** Compute \(GM_{n,m,k}\) using Eq. (42)

**Step 6** Compute \(GMI_{n,m,k}\) using Eq. (47)

**Step 7** Compute \(HMI_{n,m,k}\) of \((n, m, k)\) - th order using Eq. (50)

**Inputs:** \( \tilde{f}(x, y, z) \) : normalized 3D image of size \( N \times M \times K \); \((n, m, k)\) : DHMIs order; \(a, b\) and \(c\) : DHPs parameter values

**Outputs:** \(HMI_{n,m,k}\) : DHMIs of order \((n, m, k)\)

**Step 1** Compute \(A_k^{(a,b,c)}\) using Eq. (9)

**Step 2** Compute \(B_k^{(a,b,c)}\) using Eq. (10)

**Step 3** Compute \(d_k^2\) using Eqs. (22) – (23)

**Step 4** Compute \(C(k,i)\) using Eq. (15)

**Step 5** Compute \(GM_{n,m,k}\) using Eq. (42)

**Step 6** Compute \(GMI_{n,m,k}\) using Eq. (47)

**Step 7** Compute \(HMI_{n,m,k}\) of \((n, m, k)\) - th order using Eq. (50)
To quantify the invariance error between the feature vector values of the original image denoted \( V = DHMI_s \), and the feature vector values of the geometrically transformed image that is denoted \( V^* = DHMI_s^* \), the following Relative Error (RE) criterion can be used:

\[
RE = \frac{||V - V^*||}{||V||}
\]  

(51)

where \( ||.|| \) denotes the Euclidean norm. Noting that a low \( RE \) value indicates that the method used to extract the feature vector is accurate.

It is worth noting that the computation of DHMIs (Algorithm 2 or 4) in both 2D and 3D domains depends on the local parameters of DHPs \((a, b, c)\), which influence the \( RE \) values. Therefore, the optimal selection of these parameters remains an open challenge.

The literature survey related to the image analysis by the invariant moments method shows that the choice of the local parameters is generally made by empirical methods for few cases of the local parameter values. In order to overcome this limitation, we propose to use a population-based optimization algorithm that is the Sine Cosine (SCA) for selecting the local parameters of the orthogonal polynomials during the computation of DHMIs.

### 1.6 Sine Cosine Algorithm (SCA) for optimal selection of DHPs parameter values

The SCA is introduced by Seyedali Mirjalili [34] to solve optimization problems in various fields of engineering. This algorithm is considered as one of the most recent and efficient methods for global optimization. It is based on a mathematical model of the trigonometric functions of \( \text{Sine} \) and \( \text{Cosine} \). SCA allows to generate and improve a set of solutions through Eqs. (52)–(53) [34].

\[
V_{t+1}^i = V_t^i + r_1 \sin(r_2) \left| r_3 P_i^t - V_t^i \right|
\]  

(52)

\[
V_{t+1}^i = V_t^i + r_1 \cos(r_2) \left| r_3 P_i^t - V_t^i \right|
\]  

(53)

where \( V_t^i \) is a vector representing the current solution of the \( i \)-th solution in the \( t \)-th iteration, \( r_1 \) is a random vector defining the amplitude of the range of \( \sin \) and \( \cos \), \( r_2 \) is a random variable used to define the range of \( \sin \) or \( \cos \). The parameter \( r_3 \) represents a random value for the destination in the definition of the new position of the solution, \( P_t \) represents the position of the destination point in the \( i \)-th dimension, and \( |.| \) indicates the absolute value.

SCA uses the Eqs.(52)–(53) with a 50% probability according to the following Eq. (48):

\[
V_{t+1}^i = \begin{cases} 
V_t^i + r_1 \sin(r_2) \left| r_3 P_i^t - V_t^i \right| & |r_4| < 0.5 \\
V_t^i + r_1 \cos(r_2) \left| r_3 P_i^t - V_t^i \right| & |r_4| \geq 0.5 
\end{cases}
\]  

(54)

where \( r_4 \) is a random number ranging from 0 to 1.

Figure 2 shows the effects of \( \text{Sine} \) and \( \text{Cosine} \) functions on Eqs. (52) and (53). It shows how the equations describe a space between two solutions in the search area.

To stabilize exploitation and exploration, the \( \text{Sine} \) and \( \text{Cosine} \) dimensions are adaptively modified in Eqs. (52)–(54) using the following formula:

\[
r_1 = a - \frac{t}{T} a
\]  

(55)
where $t$ is the current iteration, $T$ is the maximum number of iterations and $a$ is a constant. Figure 3 presents a theoretical model of Sine and Cosine functions of dimension $[-2, 2]$.

Figures 2 and 3 show that the SCA algorithm allows to explore the search space when the Sine and Cosine functions ranges are between $[23, 44]$ and $[-2,-1]$. On the other hand, the search is exploited when the ranges are in the interval $[-1, 1]$.

In this work, we use SCA to optimize the values of the local parameters $a$, $b$ and $c$ of DHPs through the minimization of the relative error ($RE$), which is used as cost function of the SCA algorithm. The pseudo-code of the proposed method for optimal DHMIs computation is summarized in algorithm 5.

| Inputs: | 2D (or 3D) image function, the image size. |
| Outputs: | $a_{opt}$, $b_{opt}$ and $c_{opt}$ : the optimal DHPs parameters, $V_{opt}$ (DHMIs) :the optimal feature vector of the 2D (or 3D) image. |

| Initialization | $SA = 100$ (Number of search agents (possible solutions)). $T = 200$ (Maximum number of iterations). $Dim = 3$ (The problem dimension because we are looking for the values of $a$, $b$ and $c$ HPs parameter values). $Lb = [0, 0]$ Image size (The minimum values of the parameters $a$, $b$ and $c$, respectively). $Ub = [50, 50, 1.5]$ Image size (The maximum values of the parameters $a$, $b$ and $c$, respectively). Initialize an $SA$ set of search agents. |

| Do | Calculate DHMIs of the 2D (or 3D) image for each search agent using Algorithm 2 (or 4). Evaluate the cost function given by Eq. (51) for each research agent. Update the best solution obtained so far ($P = V_i$). Update $r_1$, $r_2$, $r_3$ and $r_4$ values. Update the position of search agents using Eq. (44). |

| While ($t < T$) | Return the best solution obtained until the global optimum is reached. |

1.7 Application of DHMIs and SCA to 2D and 3D images zero-watermarking

In this section, we present the proposed 2D/3D image zero-watermarking scheme based on 2D/3D DHMIs. This scheme is mainly designed to protect the intellectual property rights of both 2D and 3D images. The proposed scheme includes two fundamental phases, namely (i)
the phase of the zero-watermark image generation using a binary logo image (watermark) and
the feature vector (DHMIs) that is extracted from the 2D/3D image to be protected. (ii) The
second phase is the watermark recovering by using the zero-watermark image and the feature
vector (DHMIs) that is computed from the attacked 2D/3D image. Therefore, the proposed
method is blind, i.e. it does not require the original image. The graphical abstract shows the
global scheme of the proposed zero-watermarking method, and the details are given below.

1.8 Zero-watermark image generation phase

The generation of the zero-watermark image is based on the computation of DHMIs of the
original image via algorithm 5. The steps of this phase are summarized in Fig. 4, and further
details are presented below.

(a) Construction of the Feature Matrix

For a 2D image, algorithm 5 is used to obtain the feature vector magnitude \( V = |HMI_{nm}| \) of
the original image up to the order \((n, m)\). Then, we rearrange the elements of \( V \) in a square
matrix of size \( L \times L \) with \( L \leq n \times m \). The latter will be referred as the feature matrix of
the 2D image.

In the case of a 3D image, we calculate the feature vector \( V = |HMI_{nmk}| \) of this image up to
the order \((n, m, k)\) using algorithm 5, then we rearrange \( V \) elements in a square matrix of size \( L \times L \) (with \( L \leq n \times m \times k \)). The latter referred as the feature matrix of the 3D image.

For security purpose, the values of the local parameters \((a, b, c)\) are given as a security
key denoted \textit{KEY 1}.

(b) Construction of the Feature Image

Fig. 3 The sine and cosine functions having the dimension in \([-2, 2]\) tolerate a solution to go beyond (outside the
space between them) or around (inside the space between them) the destination [34]
In this step, we rearrange the elements of the feature matrix of size $L \times L$ into a vector $V_c$ of size $1 \times L^2$, then we compute the median $T$ of this vector in order to binarize their elements as follows:

$$V_b(i) = \begin{cases} 1, & \text{if } V_c(i) \geq T \\ 0, & \text{if } V_c(i) < T \end{cases}; i = 1, 2, \ldots, L^2 \quad (56)$$

Next, the binary vector elements are rearranged in a matrix $C$ of size $L \times L$. The latter called the feature image.

### (c) Permutation of the Feature Image

To further increase the security of the feature image, we randomly generate a binary matrix $B$ of the same size as $C$. Then, we apply the or-exclusive operation between $C$ and $B$ in order to obtain the permuted feature image denoted $C_p$ as follows:

$$C_p = \text{XOR}(C, B) \quad (57)$$

where the operation “or-exclusive” satisfies the following rule:

$$A = \text{XOR}(\text{XOR}(A, B), B) \text{ with } A, B \in \{0, 1\} \quad (58)$$

### (d) Logo Image Permutation

In this step, we use a binary logo image (watermark) of size $L \times L$. Then, we improve the security of this image by using a pseudo-random 2D permutation based on the Arnold’s cat map that is defined as [7]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \varepsilon \\ \eta & \eta + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod(L), 0 \leq x, y \leq n \quad (59)$$

where $\varepsilon$ and $\eta$ are positive integer parameters, $(x, y)$ is the image pixel coordinates of the original logo and $(x', y')$ is the image pixel coordinates of the permuted logo. The most important property of the chaotic Arnold’s transformation is the ability to reorganize the

![Flowchart of the zero-watermark image generation phase](image-url)
original image pixels after $\gamma$ iterations. For security reasons, the values of $\varepsilon$, $\eta$ and $\gamma$ are given as security key denoted $KEY\ 2$.

(c) **Zero - watermark Image Generation**

To create the zero-watermark ($ZW$) image, we use the “or-exclusive” operation as follows:

$$ZW = \text{XOR}(C_p, L_p)$$ \hspace{1cm} (60)

where $C_p$ is the permuted feature image and $L_p$ is the permuted logo image.

**1.9 Watermark image recovery phase**

The watermark image recovery phase allows to validate the intellectual property rights of the protected 2D/3D image. The summary of this phase is presented in Fig. 5, and the details are given below.

(a) **Construction of the Feature Matrix**

For a protected 2D/3D image, $KEY\ 1$ is used to compute the amplitude of DHMIs through algorithm 2 for a 2D image, or via algorithm 4 for a 3D image. Then, the feature matrix of the protected image is generated by the same method described in subsection 5.1 (a).

(b) **Construction of the Feature Image**

In this step, the feature matrix of the protected image is binarized by the same method described in subsection 5.1 (b) to obtain the feature image denoted $C^\ast$.

(c) **Permutation of the Feature Image**

The feature image $C^\ast$ with the binary matrix $B$ that is generated in the feature image permutation step (section 5.1 (c)), are used to create the permuted feature image denoted $C^\ast_p$ as follows:

---

**Fig. 5** Flowchart of the proposed scheme for recovering the logo (watermark) image
(d) Binary Logo Image Recovering

In this step, the permuted logo image is generated from the permuted feature image \( C_p^* \) with the zero-watermark image \( ZW \) as follows:

\[
L_{p^*} = XOR(C_p^*, ZW)
\]

Then we apply the inverse Arnold’s transformation to \( L_{p^*} \) using \( KEY \) 2 to recover the original binary logo image (watermark).

In order to measure the bit errors between the recovered watermark and the original one, we can use the Bit Error Rate (BER) criterion that is defined by:

\[
BER(B, B^*) = \frac{1}{L \times L} \sum_{i=1}^{n} \sum_{j=1}^{m} |B(i, j) - B^*(i, j)|
\]

where \( L \times L \) is the binary logo size, \( B \) is the original binary logo and \( B^* \) is the recovered binary logo. Noting that a low BER value indicates that the zero-watermarking method is efficient.

The normalized cross correlation (NCC) criterion is also used to measure the similarity between \( B \) and \( B^* \) watermarks. NCC is defined by the following relation [37]:

\[
NCC(B, B^*) = \frac{\sum_{x,y} (B - \bar{B})(B^* - \bar{B}^*)}{\sqrt{\sum_{x,y} (B - \bar{B})^2 \sum_{x,y} (B^* - \bar{B}^*)^2}}
\]

where \( \bar{B} \) and \( \bar{B}^* \) are the mean values of \( B \) and \( B^* \), respectively. Note that if the NCC value tends to the value 1, the similarity between \( B \) and \( B^* \) is high.

The structural similarity (SSIM) index can also be used to compute the difference between two image pixels. SSIM criterion is computed as follows [43]:

\[
SSIM(B, B^*) = \frac{(2\mu_B\mu_{B^*} + a_1)(2\sigma_{BB^*} + a_2)}{(\mu_B^2 + \mu_{B^*}^2 + a_1)(\sigma_B^2 + \sigma_{B^*}^2 + a_2)}
\]

where \( \mu_B, \sigma_B \) and \( \sigma_{BB^*} \) represent successively the mean, deviation, and cross correlation between \( B \) and \( B^* \) images. \( a_1 \) and \( a_2 \) are positive constants. If \( SSIM = 1 \), this means that there is no difference between \( B \) and \( B^* \) images.

2 Experimental results

In this section, we present the results of simulations and comparisons that validate the efficiency of the present work. First, we perform a time analysis of the proposed algorithms. Then, we demonstrate the relevance of the local parameters of DHPs in the extraction of both global and local image characteristics. Next, we show the invariability property of DHMIs against geometric and noise attacks. In addition, we demonstrate the advantages of SCA in
optimizing the local parameters of DHPs during the computation of DHMIs. Moreover, we show the robustness of the proposed zero-watermarking scheme against geometric and other common image processing attacks. Finally, we provide several between the proposed zero-watermarking scheme and recent existing ones to demonstrate the superiority of the proposed scheme.

2.1 Timing analysis of DHMIs computation methods

Before applying 2D and 3D DHMIs in the zero-watermarking of 2D and 3D images, we show the superiority of the proposed algorithms (2 and 4) in terms of computation rapidity compared to the conventional algorithms (1 and 3) of DHMIs computation. For that, the 2D and 3D test images are resized to $64 \times 64$ pixels and $64 \times 64 \times 64$ voxels, respectively (Fig. 6) to avoid the numerical instability of DHMIs due to the gamma and factorial functions involved in the conventional calculation methods (algorithms 1 and 3). Table 1 shows a comparison in terms of execution time of 2D DHMIs using algorithms 1 and 3, and Table 3 illustrates the computation time of 3D DHMIs using algorithms 2 and 4. The improvement in the computation time is measured by the execution time improvement ratio criterion (ETIR (%)) that is defined as follows [20]:

$$ETIR(\%) = \left(1 - \frac{Time_1}{Time_2}\right) \times 100$$  \hspace{1cm} (66)

where $Time_1$ represents the average of DHMIs computation time using the improved algorithms, and $Time_2$ is the average computation time of DHMIs using the conventional algorithms. Noting that all the simulations in the present work are conducted using Matlab R2019a installed on a PC with a 2.4 GHz processor and RAM of 4 GB.

On one hand, the results of the timing analysis presented in Table 1 show the proposed algorithm 3 is considerably rapid in comparison to algorithm 1. Indeed, an improvement of $ETIR \geq 70\%$ is obtained for all the moment orders. This is explained by the use of matrix formulations to compute 2D DHMIs (algorithm 3) instead of the direct computation

![Fig. 6](image)

(a) Standard grayscale test images resized to $64 \times 64$ pixels. (b) 3D test images resized to $64 \times 64 \times 64$ voxels
On the other hand, we observe in Table 2 a slight superiority of algorithm 4 over algorithm 2 in terms of 3D DHMIs computation time. This superiority is more noticeable in the lower orders of 3D DHMIs. We explain this result by the fact that the computation of 3D DHMIs either by algorithm 2 or by algorithm 4 is performed through direct formulations, and the slight improvement of algorithm 4 is due to the recursive computation of $A_n(a, b, c)$, $B_{nk}(a, b, c)$, and $d_n^2$ instead of their direct formulations used in algorithm 2. However, the main advantage of algorithms 4 over algorithms 2 is the numerical stability. Therefore, both algorithms 2 and 4 are relevant for the computation of DHMIs of large-size images. This advantage will be exploited in the zero-watermarking application of large-size 2D and 3D images.

### 2.2 Global and local image characteristics extraction using DHPs parameters

Krawtchouk (local descriptors) and Tchebichef (global descriptors) polynomials are particular cases of DHPs. Thus, DHMs are more appropriate to be local and global image descriptors by adjusting the values of their local parameters $(a, b, c)$. In order to show the property of DHMs in extracting the global and local image characteristics, the 2D “Textures” image of size $1024 \times 1024$ and the 3D image “Pelvis” of size $256 \times 256 \times 256$ are reconstructed using DHMs with different values of the parameters $a$, $b$, and $c$. The results of the conducted tests are presented in Figs. 7 and 8.

The results obtained in Figs. 7 and 8 show that the DHMs are suitable for global and local image characteristics extraction for both 2D and 3D images. Indeed, in the case $a = c = 0$ and $b > N$ (with $N \times N$ is the image size), DHMs are able to extract the global image features (global reconstruction). In the case $a > 0$, $c > 0$ and $b = a + N$, DHMs extract the local image features (local reconstruction).

### 2.3 Invariability of 2D DHMIs against 2D image attacks

In the first experiment of this subsection, we prove the invariance of 2D DHMIs against the rotation, scaling, translation, and noise adding. To do this, we select from the database [36] a real medical image named “Covid-19” of size $1024 \times 1024$ (Fig. 9). This image is then transformed by the following geometrical transformations: scale by a factor ranging from 0.5

#### Table 1 The average execution time for computing 2D DHMIs by algorithm 3, and the conventional method (algorithm 1)

| 2D DHMIs orders | (5,5) | (10,10) | (15,15) | (20,20) |
|----------------|-------|---------|---------|---------|
| Execution time in seconds (s) using: Algorithm 1 | 0.2132 | 0.7189 | 1.0305 | 2.6209 |
| Algorithm 3 | 0.0634 | 0.1690 | 0.2866 | 0.7166 |
| ETIR (%) | 70.26 | 71.89 | 72.19 | 72.66 |

#### Table 2 The average time for computing 3D DHMIs by algorithm 4, and the conventional method (algorithm 2)

| 3D DHMIs orders | (5,5,5) | (10,10,10) | (15,15,15) | (20,20,20) |
|----------------|---------|------------|------------|------------|
| Execution time in seconds (s) using: Algorithm 2 | 16.3978 | 80.1698 | 206.4779 | 752.1352 |
| Algorithm 4 | 14.5058 | 76.1563 | 201.4779 | 747.9258 |
| ETIR (%) | 11.5381 | 5.0062 | 2.4216 | 0.5597 |
"Textures" image of size $1024 \times 1024$.

Values of $a$, $b$ and $c$ parameters

| Parameters | Image | Parameters | Image | Parameters | Image | Parameters | Image |
|------------|-------|------------|-------|------------|-------|------------|-------|
| $a = c = 0$ and $b = a + N$ | ![Image 1] | $a = 100$ and $b = a + N$ | ![Image 2] | $a = 500$ and $b = a + N$ | ![Image 3] | $a = c = 0$ and $b = a + N + 5000$ | ![Image 4] |

Fig. 7 Global and local “Textures” image reconstruction based on $a$, $b$ and $c$ DHPs parameters

to 2 with a step of 0.1, rotation by angles ranging from 0° to 360° with a step of 10°, and translation by a vector ranging from $(-20, -20)$ to $(20, 20)$ with a step of $(+4, +4)$. For each transformation, DHMIs are computed by both algorithms 2 and 5 up to the order $(n, m) = (32, 32)$ for different values of the parameters $a$, $b$ and $c$. Then, we compute the $RE$ between the feature vector of the original image $V = \{DHMI_{00}, DHMI_{01}, \ldots, DHMI_{nm}\}$ and the feature vector of the transformed one $V^* = \{DHMI^*_{00}, DHMI^*_{01}, \ldots, DHMI^*_{nm}\}$. The robustness of DHMIs is also tested against two types of noise namely: “Salt & Pepper” and “Gaussian”

"Pelvis" image of size $256 \times 256 \times 256$

Values of $a$, $b$ and $c$ parameters

| Parameters | Image | Parameters | Image | Parameters | Image | Parameters | Image |
|------------|-------|------------|-------|------------|-------|------------|-------|
| $a = c = 0$ and $b = a + N$ | ![Image 5] | $a = 50$ and $b = a + N$ | ![Image 6] | $a = 150$ and $b = a + N$ | ![Image 7] | $a = c = 0$ and $b = a + N + 1000$ | ![Image 8] |

Fig. 8 Global and local 3D “Pelvis” image reconstruction using DHPs parameters with DHMs are computed up to the order $(30, 30, 30)$
noise with different densities up to 5%. Figure 9 shows “Covid-19” test image with some applied attacks, and Fig. 10 shows the RE curves corresponding to the attacked test image by different transformations using various $a$, $b$ and $c$ parameter values.

The results presented in Fig. 10 show on one hand that the RE values are very low ($\approx 10^{-10}$), indicating that the DHMIs computed by the proposed algorithms are accurate and stable
against geometric and noise attacks. On the other hand, we notice that the use of SCA allows obtaining the lowest RE values compared to the other cases when the local parameters of DHPs are empirically selected. Therefore, the SCA guarantees an optimal selection of the parameters $a$, $b$ and $c$ when computing DHMIs. After validating the efficiency of the proposed computation of DHMIs, we will exploit the latter in the copyright protection of large-size 2D and 3D images.

### 2.4 Robustness of the proposed zero-watermarking scheme against various 2D image attacks

In the next test, we use a 2D color image named “Barbara” of size $1024 \times 1024$. Initially we compute DHMIs of this image up to the order $(32, 32)$ using algorithm 5, and then we apply the proposed zero-watermarking scheme to generate the zero-watermark image based on the amplitude of the computed DHMIs and the binary logo image of size $32 \times 32$ (Fig. 11). Noting that the application of the proposed zero-watermarking scheme to a color image requires the decomposition of the latter into three channels R, G and B. Then, we apply the proposed scheme to a single channel that is considered as host image. In the present test, the G-channel is used as host image. Then, “Barbara” image is attacked by 10 transformations illustrated in Fig. 11. The recovered watermarks from the attacked images are shown in Fig. 13 with the corresponding BER, SSIM and NCC values.

![Image]

The original « Barbara » image of size $1024\times1024$, the watermark and zero-watermark images of size $32\times32$

| # (a) JPEG compression quality (50) | # (b) “Salt & Pepper” noise density (3%), # (c) « Gaussian » noise variance (3%), # (d) Median filter size (7x7), # (e) The Average filter size (5x5), # (f) Histogram equalization (Level 200), # (g) Percentage cropping at top left corner (25%), # (h) Rotation (40°), # (i) Translation ($\pm20$, $\pm20$), # (j) Scaling by 0.7 |

*Fig. 11 “Barbara” image under different attacks*
The results in Fig. 12 show that the BER values are very low (close to zero), and the SSIM and NCC values tend towards 1, which clearly indicates that the proposed method is of high efficiency for the zero-watermarking of large-size 2D images.

2.5 Robustness of the proposed zero-watermarking scheme against various 3D image attacks

In the following test, we use the 3D “Vertebrae” test image of size 256 × 256 × 256. Next, algorithm 5 is used to compute DHMIs of this image up to the order (11, 11, 11). Then, 1024 coefficients are randomly selected from the computed DHMIs, and the selected coefficients are rearranged in 2D matrix of size 32 × 32 (Fig. 13) for applying the proposed zero-watermarking scheme. The test image is attacked by 12 transformations including geometric attacks (rotation, translation, scaling and combined transformations), contamination by “Salt & Papper” and “Speckle” noise, cropping, compression, and filtering attacks. The attacked 3D image by various transformations is shown in Fig. 13. Then, the proposed scheme is then used to recover the watermark from each attacked image, and the recovered watermarks with the corresponding BER, SSIM and NCC values are illustrated in Fig. 14.

The achieved results presented in Fig. 14 clearly show the high efficiency of the proposed zero-watermarking scheme for 3D images intellectual property rights protection because the BER values corresponding to the recovered logos are very close to zero and the SSIM and NCC values tend toward one. The efficiency of the proposed 2D / 3D image zero-watermarking method can be explained by the following factors: 1) the proposed method relies on the use of lower-order DHMIs, which allows the extraction of stable and robust 2D/3D image feature vector. 2) The use of SCA for the selection of local parameters when computing DHMIs allows the extraction of an optimal feature vector of the protected image. 3) DHMIs are independent of the image geometric transformations (translation, rotation and scaling).
Original “Vertebrae” image of size $256 \times 256 \times 256$ with logo of size $32 \times 32$ and the generated zero-watermark.

| Transformation #1          | Transformation #2          | Transformation #3            |
|---------------------------|---------------------------|-----------------------------|
| Scaling by 1.2 factor     | Scaling by 0.5 factor     | Rotation by angle (0°, 180°, 90°) |

| Transformation #4          | Transformation #5          | Transformation #6            |
|---------------------------|---------------------------|-----------------------------|
| Rotation by angle (60°, 0°, 0°) | Translate (+15, +15, +15) | Translate (0, -15, -10)     |

| Transformation #7          | Transformation #8          | Transformation #9            |
|---------------------------|---------------------------|-----------------------------|
| Scale 0.75 + rotate by angle (45°, 90°, 180°) + translate (+10, +10, 0) | “Salt & pepper” noise of density 0.03 | “Speckle” noise of density 0.02 |

| Transformation #10         | Transformation #11         | Transformation #12           |
|---------------------------|---------------------------|-----------------------------|
| Gaussian filter size (5 × 5 × 5) | Cropping (25%)          | Compression (50%)            |

Fig. 13 3D “Vertebrae” test image under different types of attacks
scaling), and are robust to different image noise ("Salt & Pepper" and "Gaussian"), which guarantees a high robustness of the proposed method against geometric attacks.

2.6 Comparisons with similar 2D and 3D image zero-watermarking schemes

In this section, we present comparisons of the proposed scheme with recent zero-watermarking schemes of 2D and 3D images. In the first test, we compare the robustness of the proposed 2D zero-watermarking scheme with recent schemes presented in [24, 45–47, 51]. In the comparison experiments, we use four test images each of size $1024 \times 1024$ with a binary logo of size $32 \times 32$ (Fig. 15). It should be noted that the robustness of the test schemes is measured using

\[
\begin{align*}
\text{BER} &= 0.0039 & \text{BER} &= 0.0020 & \text{BER} &= 0.0156 & \text{BER} &= 0.0078 \\
\text{SSIM} &= 0.9999 & \text{SSIM} &= 0.9999 & \text{SSIM} &= 0.9712 & \text{SSIM} &= 0.9998 \\
\text{NCC} &= 0.9918 & \text{NCC} &= 0.9924 & \text{NCC} &= 0.9997 & \text{NCC} &= 0.9835 \\
\text{BER} &= 0.0058 & \text{BER} &= 0.0039 & \text{BER} &= 0.0078 & \text{BER} &= 0.0058 \\
\text{SSIM} &= 0.9998 & \text{SSIM} &= 0.9999 & \text{SSIM} &= 0.9998 & \text{SSIM} &= 0.9998 \\
\text{NCC} &= 0.9776 & \text{NCC} &= 0.9918 & \text{NCC} &= 0.9835 & \text{NCC} &= 0.9776 \\
\text{BER} &= 0.0098 & \text{BER} &= 0.0078 & \text{BER} &= 0.0195 & \text{BER} &= 0.0137 \\
\text{SSIM} &= 0.9998 & \text{SSIM} &= 0.9998 & \text{SSIM} &= 0.9995 & \text{SSIM} &= 0.9997 \\
\text{NCC} &= 0.9797 & \text{NCC} &= 0.9835 & \text{NCC} &= 0.9592 & \text{NCC} &= 0.9713
\end{align*}
\]

Fig. 14 The recovered watermark from each attacked image shown in Fig. 13 with the corresponding BER, SSIM and NCC values

Fig. 15 a–d original images and e binary logo (watermark) used in the comparison test

Springer
the average BER values that correspond to the recovered watermarks from the attacked images. The results of the performed comparison are illustrated in Table 3.

The results illustrated in Table 3 indicate that the average BER values obtained when using the proposed scheme are lower than those obtained by the compared schemes, indicating that the proposed scheme presents excellent performance in comparison to the other schemes. This superiority can be justified by the strong robustness of DHMIs against both geometric and common image processing attacks. In addition, the use of SCA when computing DHMIs ensures the extraction of an optimal and stable feature vector, thus the SCA improves the performance of the proposed zero-watermarking scheme.

The 3D image zero-watermarking schemes remain relatively few. Therefore, the robustness of the proposed scheme is only compared to the latest three schemes presented in [31, 32, 42]. To conduct this comparison, we use six 3D images each of size $256 \times 256 \times 256$ (Fig. 16). These images are attacked by various transformations given in Table 4. Then, the average BER is computed using on the recovered watermark from each attacked image.

The results of this comparison (Table 4) show that the proposed method performs better than the methods presented in [31, 32, 42] because the average BER values obtained by the proposed scheme are always lower than those obtained by the scheme used in this comparison. The superiority of the proposed method over the other compared schemes is justified by the fact that the 3D DHMIs are invariants with respect to the geometric transformations of the 3D image (translation, rotation and scale change). By contrast, the other compared 3D zero-watermarking schemes are influenced by the geometric attacks of the 3D image. In addition,
the lower order 3D DHMIs are stable and robust against different types of noise and common 3D image processing attacks.

Table 4  Robustness comparison in terms of average BER between the proposed scheme and recent 3D image zero-watermarking schemes presented in [31, 32, 42]

| Attacks                      | Proposed | Method [32] | Method [42] | Method [31] |
|------------------------------|----------|-------------|-------------|-------------|
| JPEG compression (QF=25)     | 0.0105   | 0.0355      | 0.0241      | 0.0325      |
| JPEG compression (QF=50)     | 0.0123   | 0.0220      | 0.0355      | 0.0443      |
| Rotation with 90 degree      | 0.0010   | 0.0269      | 0.0830      | 0.0506      |
| Rotation with 180 degree     | 0.0015   | 0.0508      | 0.0326      | 0.0408      |
| Scaling 0.75                 | 0.0072   | 0.0608      | 0.0720      | 0.0608      |
| Scaling 1.5                  | 0.0035   | 0.0594      | 0.0502      | 0.0987      |
| Translation (+5, +5)         | 0.0042   | 0.0998      | 0.0881      | 0.0908      |
| Translation (+10, +10)       | 0.0090   | 0.0821      | 0.0803      | 0.0758      |
| Salt & pepper noise (3%)     | 0.0062   | 0.0500      | 0.0476      | 0.0608      |
| Salt & pepper noise (5%)     | 0.0083   | 0.0251      | 0.0403      | 0.0409      |
| Gaussian noise ($\sigma=0.03$) | 0.0050 | 0.0790      | 0.0623      | 0.0607      |
| Gaussian noise ($\sigma=0.05$) | 0.0042 | 0.0139      | 0.0502      | 0.0309      |
| Average filtering (3 × 3)    | 0.0075   | 0.0302      | 0.0412      | 0.0503      |
| Average filtering (5 × 5)    | 0.0110   | 0.0362      | 0.0390      | 0.0905      |
| Gaussian filtering (3 × 3)   | 0.0069   | 0.0205      | 0.0512      | 0.0603      |
| Gaussian filtering (5 × 5)   | 0.0095   | 0.0405      | 0.0414      | 0.0206      |
3 Conclusion

In this paper, we have improved considerably the computation of 2D and 3D DHMIs. Then, SCA is efficiently used for the optimization of local DHPs parameters when computing DHMIs. These are used to propose a new and robust zero-watermarking scheme used in both 2D and 3D images copyrights protection. The simulation results and comparisons have shown that the numerical stability of DHMIs is significantly improved due to provided computational improvements. On the other hand, the high robustness of the proposed zero-watermarking scheme against different types of geometric and common attacks of 2D/3D images has been demonstrated. In the future, the proposed algorithms of 2D/3D DHMIs computation will be applied in other applications that are based on the invariance property, such as pattern recognition and classification of large-size images.

References

1. Ali Z, Imran M, Alsulaiman M, Shoaib M, Ullah S (2018) Chaos-based robust method of zero-watermarking for medical signals. Futur Gener Comput Syst 88:400–412
2. Benouini R, Batioua I, Zenkouar K, Mrabti F, Fadili HE (2019) New set of generalized legendre moment invariants for pattern recognition. Pattern Recognition Letters 123:39–46. https://doi.org/10.1016/j.patrec.2019.03.001
3. Benouini R, Batioua I, Zani A, Najah S, Qjidaa H (2019) Fractional-order orthogonal Chebyshev moments and moment invariants for image representation and pattern recognition. Pattern Recogn 86:332–343. https://doi.org/10.1016/j.patcog.2018.10.001
4. Benouini R, Batioua I, Zenkouar K, Zahi A, Fadili HE, Qjidaa H (2019) Fast and accurate computation of Racah moment invariants for image classification. Pattern Recogn 91:100–110. https://doi.org/10.1016/j.patcog.2019.02.014
5. Bocci C, Carlini E, Kileel J (2016) Hadamard products of linear spaces. Journal of Algebra 448:595–617. https://doi.org/10.1016/j.jalgebra.2015.10.008
6. Camacho-Bello C, Rivera-Lopez JS (2018) Some computational aspects of Tchebichef moments for higher orders. Pattern Recogn Lett 112:332–339. https://doi.org/10.1016/j.patrec.2018.08.020
7. Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons & Fractals 21:749–761
8. Daoui A, Yamni M, Karmouni H, Sayyouri M, Qjidaa H (2020) Biomedical signals reconstruction and zero-watermarking using separable fractional order Charlier- Krawtchouk transformation and sine cosine algorithm (SCA). Signal Process 107854. https://doi.org/10.1016/j.sigpro.2020.10.7854
9. Daoui A, Yamni M, Ogri OE, Karmouni H, Sayyouri M, Qjidaa H (2020) New algorithm for large-sized 2D and 3D image reconstruction using higher-order Hahn moments. Circuits Syst Signal Process. https://doi.org/10.1007/s00034-020-01384-z
10. Daoui A, Yamni M, Karmouni H, Sayyouri M, Qjidaa H (2020) Efficient Reconstruction and Compression of Large Size ECG Signal by Tchebichef Moments. In: 2020 International conference on intelligent systems and computer vision (ISCV). IEEE, pp 1–6
11. Daoui A, Yamni M, Karmouni H, El Ogri O, Sayyouri M, Qjidaa H (2020) Fast and stable bio-signals reconstruction using Krawtchouk moments. Advances in Intelligent Systems and Computing 1076:369–377. https://doi.org/10.1007/978-981-15-0947-6_35
12. Daoui A, Sayyouri M, Qjidaa H (2020) Efficient computation of high-order Meixner moments for large-size signals and images analysis. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-09739-z
13. Daoui A, Yamni M, El O, Ogri HK, Sayyouri M, Qjidaa H (2020) Stable computation of higher order Charlier moments for signal and image reconstruction. Inf Sci 521:251–276. https://doi.org/10.1016/j.ins.2020.02.019
14. Daoui A, Karmouni H, Sayyouri M, Qjidaa H, Maaroufi M, Alami B (2021) New robust method for image copyright protection using histogram features and sine cosine algorithm. Expert Syst Appl 177:114978
15. Daoui A, Karmouni H, Sayyouri M, Qjidaa H (2021) Efficient methods for signal processing using Charlier moments and artificial bee Colony algorithm. Circuits, Systems, and Signal Processing. https://doi.org/10.1007/s00034-021-01764-z
16. Du Shun ZY, Xinyu W (2013) A zero watermarking algorithm for 3D mesh models based on shape diameter function. Journal of Computer-Aided Design & Computer Graphics 5
17. El Ogri O, Daoui A, Yamni M, Karmouni H, Sayyouri M, Qjidaa H (2020) New set of fractional-order generalized Laguerre moment invariants for pattern recognition. Multimed Tools Appl 79:23261–23294. https://doi.org/10.1007/s11042-020-09084-1
18. Flusser J, Suk T, Zitova B (2016) 2D and 3D image analysis by moments. John Wiley & Sons
19. Hmimid A, Sayyouri M, Qjidaa H (2018) Image classification using separable invariant moments of Charlier-Meixner and support vector machine. Multimed Tools Appl 77:23607–23631. https://doi.org/10.1007/s11042-018-5623-3
20. Hosny KM (2008) Fast computation of accurate Zernike moments. J Real-Time Image Proc 3:97–107. https://doi.org/10.1007/s11554-007-0058-5
21. Hosny KM, Darwish MM, Aboelenen T (2020) New fractional-order Legendre-Fourier moments for pattern recognition applications. Pattern Recognition:107324
22. Jahid T, Hmimid A, Karmouni H, Sayyouri M, Qjidaa H, Rezzouk A (2018) Image analysis by Meixner moments and a digital filter. Multimed Tools Appl 77:19881–19883
23. Jiang F, Gao T (2020) A robust zero-watermarking algorithm for color image based on tensor mode expansion. Multimed Tools Appl 79:7599–7614
24. Kang X, Zhao F, Chen Y, Lin G, Jing C (2020) Combining polar harmonic transforms and 2D compound chaotic map for distinguishable and robust color image zero-watermarking algorithms. J Vis Commun Image Represent 102804
25. Karakasis EG, Papakostas GA, Koulouriots DE, Tourassis VD (2013) Generalized dual Hahn moment invariants. Pattern Recogn 46:1998–2014. https://doi.org/10.1016/j.patcog.2013.01.008
26. Karmouni H, Jahid T, El Affar I, Sayyouri M, Hmimid A, Qjidaa H, Rezzouk A (2017) Image analysis using separable Krawtchouk-Tchebichef’s moments. In: 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). IEEE, pp 1–5
27. Karmouni H, Jahid T, Lakhili Z, Hmimid A, Sayyouri M, Qjidaa H, Rezzouk A (2017) Image reconstruction by Krawtchouk moments via digital filter. In: 2017 Intelligent Systems and Computer Vision (ISCV). IEEE, pp 1–7
28. Karmouni H, Jahid T, Hmimid A, Sayyouri M, Qjidaa H (2019) Fast computation of inverse Meixner moments transform using Clenshaw’s formula. Multimed Tools Appl 78:31245–31265. https://doi.org/10.1007/s11042-019-07961-y
29. Karmouni H, Yamni M, El Ogri O, Daoui A, Sayyouri M, Qjidaa H (2020) Fast computation of 3D Meixner’s invariant moments using 3D image cuboid representation for 3D image classification. Multimed Tools Appl 79:29121–29144. https://doi.org/10.1007/s11042-020-09351-1
30. Koekoek R, Lesky PA, Swarttouw RF (2010) Hypergeometric orthogonal polynomials and their q-analogues. Springer Science & Business Media
31. Lee J-S, Liu C, Chen Y-C, Hung W-C, Li B (2021) Robust 3D mesh zero-watermarking based on spherical coordinate and skewness measurement. Multimed Tools Appl:1–16
32. Liu G, Wang Q, Wu L, Pan R, Wan B, Tian Y (2021) Zero-watermarking method for resisting rotation attacks in 3D models. Neurocomputing. 421:39–50. https://doi.org/10.1016/j.neucom.2020.09.013
33. Mesbah A, Zouhri A, El Mallahi M, Zenkouar K, Qjidaa H (2017) Robust reconstruction and generalized dual Hahn moments invariants extraction for 3D images. 3D Research 8:7
34. Mirjali S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
35. Peng C, Cao D, Wu Y, Yang Q (2019) Robot visual guide with Fourier-Mellin based visual tracking. Frontiers of Optoelectronics 12:413–421
36. Radiopaedia.org, the wiki-based collaborative Radiology resource, (n.d.). https://radiopaedia.org/. Accessed 31 March 2020
37. Satoh S (2011) Simple low-dimensional features approximating NCC-based image matching. Pattern Recogn Lett 32:1902–1911. https://doi.org/10.1016/j.patrec.2011.07.027
38. Sayyouri M, Hmimid A, Qjidaa H (2012) A fast computation of Hahn moments for binary and gray-scale images. In: 2012 IEEE International Conference on Complex Systems (ICCS), pp 1–6. https://doi.org/10.1109/ICoCS.2012.6458538
39. Sayyouri M, Hmimid A, Qjidaa H (2013) Improving the performance of image classification by Hahn moment invariants. J. Opt. Soc. Am. A, JOSAA 30:2381–2394. https://doi.org/10.1364/JOSAA.30.002381
41. Teague MR (1980) Image analysis via the general theory of moments. JOSA. 70:920–930
42. Wang X, Zhan Y (2019) A zero-watermarking scheme for three-dimensional mesh models based on multifeatures. Multimed Tools Appl 78:27001–27028
43. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612
44. Wen Q, Sun TF, Wang S (2001) Based zero-watermark digital watermarking technology, in: the third China information hiding and multimedia security workshop (CIHW). Xidain University Press, Xi’an
45. Xia Z, Wang X, Wang C, Ma B, Wang M, Shi Y-Q (2020) Local quaternion polar harmonic Fourier moments-based multiple zero-watermarking scheme for color medical images. Knowl-Based Syst 106568:106568
46. Xia Z, Wang X, Wang C, Wang C, Ma B, Li Q, Wang M, Zhao T (2021) A robust zero-watermarking algorithm for lossless copyright protection of medical images. Appl Intell:1–15
47. Xia Z, Wang X, Wang C, Ma B, Zhang H, Li Q (2021) Novel quaternion polar complex exponential transform and its application in color image zero-watermarking. Digital Signal Process 103130:103130
48. Xu T, Zhang YL, Sun JQ, ZHENG J, LIN Z (2009) Zero-watermarking scheme for 3D meshes based on geometric property. Journal of Image and Graphics 14:1818–1824
49. Yamni M, Daoui A, El O, Ogri HK, Sayyouri M, Qjidaa H, Flusser J (2020) Fractional Charlier moments for image reconstruction and image watermarking. Signal Process 171:107509. https://doi.org/10.1016/j.sigpro.2020.107509
50. Yamni M, Karmouni H, Sayyouri M, Qjidaa H (2021) Robust zero-watermarking scheme based on novel quaternion radial fractional Charlier moments. Multimedia Tools and Applications:1–30
51. Yang H, Qi S, Niu P, Wang X (2020) Color image zero-watermarking based on fast quaternion generic polar complex exponential transform. Signal Process Image Commun 82:115747
52. Zhang C, van der Baan M (2020) Seismic signal matching and complex noise suppression by Zernike moments and trilateral weighted sparse coding. IEEE Transactions on Geoscience and Remote Sensing
53. Zhu H, Shu H, Zhou J, Luo L, Coatrieux JL (2007) Image analysis by discrete orthogonal dual Hahn moments. Pattern Recogn Lett 28:1688–1704. https://doi.org/10.1016/j.patrec.2007.04.013
54. Zhu H, Shu H, Liang J, Luo L, Coatrieux J-L (2007) Image analysis by discrete orthogonal Racah moments. Signal Process 87:687–708. https://doi.org/10.1016/j.sigpro.2006.07.007

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Achraf Daoui1 · Hicham Karmouni2 · Mhamed Sayyouri1 · Hassan Qjidaa2

1 Laboratory of Engineering, Systems and Applications, National School of Applied Sciences, Sidi Mohamed Ben Abdellah-Fez University, Fez, Morocco
2 Laboratory of Electronic Signals and Systems of Information, Faculty of Science, Sidi Mohamed Ben Abdellah-Fez University, Fez, Morocco