Multimedia Tools and Applications (2021) 80:32947–32973
https://doi.org/10.1007/s11042-021-11206-2

Fast and stable computation of higher-order Hahn polynomials and Hahn moment invariants for signal and image analysis

Achraf Daoui[1] · Hicham Karmouni[2] · Mhamed Sayyouri[1] · Hassan Qjidaa[2]

Received: 7 October 2020 / Revised: 14 January 2021 / Accepted: 28 June 2021 /
Published online: 11 August 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

This article presents, on the one hand, new algorithms for the fast and stable computation of discrete orthogonal Hahn polynomials of high order (HPs) based on the elimination of all gamma and factorial functions that cause the numerical fluctuations of HPs, and based on the use of appropriate stability conditions. On the other hand, a new method for the fast and numerically stable computation of Hahn moment invariants (HMIs) is also proposed. This method is mainly based on the use of new recursive relations of HPs and of matrix multiplications when calculating HMIs. To validate the efficiency of the algorithms proposed for the calculation of HPs, several signals and large images ($\geq 4000 \times 4000$) are reconstructed by Hahn moments (HMs) up to the last order with a reconstruction error tending towards zero ($MSE \simeq 10^{-10}$). The efficiency of the proposed method for calculating HMIs is demonstrated on large medical images (2048×2048) with a very low relative error ($RE \simeq 10^{-10}$). Finally, comparisons with some recent work in the literature are provided.

Keywords Image reconstruction · Signal reconstruction · Discrete orthogonal Hahn moments · Numerical stability · Hahn moment invariants · Higher-order moments

✉ 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

The discrete orthogonal moments (DOMs) are widely applied in various fields of signal and image analysis. Applications of DOMs include signal and image reconstruction [8, 9, 50], image classification [3, 16, 20, 39], face recognition [33], image watermarking [24, 41, 44], signal zero-watermarking [10], edge detection [35], image encryption [43], signal compression [11, 2] and image compression [13, 42]. The computation of DOMs involves the computation of kernel discrete orthogonal polynomials (DOPs) such as Tchebichef [27, 28], Krawtchouk [45, 47], Hahn [46, 50], Meixner [12, 21], Charlier [8, 19, 34], dual Hahn [20, 48], Racah [49] and Shmaliy [26] polynomials.

In general, the computation of DOPs is performed recursively either with respect to the polynomial order n or with respect to the discrete variable x. Mukundan et al. discuss in [27] that the recursive computation of high-order Tchebichef polynomials with respect to the order n is unstable because of the propagation of numerical errors. To minimize this problem, Mukundan et al. use the recursive computation of Tchebichef polynomials with respect to the variable x rather than the order n. Similarly, Zhu et al. [50] propose a general form for the recursive computation of Tchebichef, Krawtchouk, Hahn, Meixner and Charlier polynomials with respect to the variable x for minimizing the propagation of numerical errors. Although the recurrence approach with respect to the variable x minimizes the propagation of numerical errors, this approach remains limited because it is unable to generate the polynomial values of high-order polynomials [1, 9]. To overcome this problem for Tchebichef polynomials, Abdulhussain et al. [1] propose a method based on the sequential integration of the traditional recurrence relations with respect to order n and with respect to variable x that were introduced by Mukundan et al. [27]. However, this proposed method is not stable for higher order Tchebichef polynomials due to the problem of numerical errors propagation [5]. To overcome this problem, Camacho-bello et al. [5] propose the computation of higher-order Tchebichef polynomials via the modified Gram-Schmidt orthonormalization process (GSOP). This method succeeded to ensure the numerical stability of higher-order Tchebichef polynomials. More recently, GSOP method is used by Daoui et al. to introduce a numerically stable computation of Hahn [9], Charlier [8] and Meixner [12] polynomials.

Even if the problems of recursive computation of certain high-order polynomials are solved, the computation of high-order Hahn polynomials (HPs) is still limited by the following two major problems: (*i*) the numerical instability of the initial values, which are defined from gamma and factorial functions. (*ii)* The propagation of round-off error when computing high-order HPs, which leads to the loss of the orthogonality property of these polynomials. To solve the problem (*i*), new algorithms for HPs computation are proposed in this work. These algorithms are free from any gamma and factorial functions. To overcome the problem (*ii*), a new numerical stability method is proposed. This method that is based on appropriate stability conditions, is very low in terms of computational cost compared to the GSOP method that is presented in [9]. As result, the computation of high-order Hahn moments (HMs) becomes stable and fast when reconstructing large-size signals and images. It is worth mentioning that the high-order moments are required for better describe the content of the images [6, 38].

The present work also discusses some computational aspects related to Hahn's moment invariants (HMIs) computation, with a view to overcoming the encountered problems.

Generally, the derivation process of the orthogonal moment invariants relies on the algebraic relation between the orthogonal moments and the geometric ones, to express the orthogonal moment invariants as a linear combination of geometric moment invariants. [3, 14, 18, 20, 30]. However, this process is limited by two major drawbacks: (i) the numerical instability due to gamma and factorial functions involved in the computation of the orthogonal moment invariants, and (ii) the high computational cost [3]. To overcome these problems when computing Racah moment invariants (RMIs), a fast and accurate computation method is proposed in [3]. This method is based on the use of recursive formulas of certain factors involved in RMIs computation. However, this method and others presented in [3, 17, 20, 39] are not suitable for computing the invariant moments of large-size images (≥1024 × 1024), since these methods involve the use of certain terms depending on gamma and factorial functions when computing the image moment invariants. Motivated by overcoming these problems, a fast and numerically stable computation of large-size image moment invariants is proposed in this paper. For this purpose, significant improvements are made on the conventional computation of HMIs presented in [36]. On one hand, the proposed improvements allow to compute HMIs independently of gamma and factorial functions, which guarantees the numerical stability of the computed moment invariants. On the other hand, HMIs execution time is considerably reduced by using matrix multiplications and recursive formulas. It is important to note that the fast computation of moment invariants is relevant for real-time applications [3].

The rest of this work is organized as follows: in the second section, the conventional and the proposed methods for HPs computation are presented. The third section contains the theoretical background of Hahn moments (HMs) in the 1D and 2D domains. The fourth section presents the conventional and the proposed method for HMIs computation. In the fifth section, the results of simulations and comparisons are presented to validate the efficiency of the proposed algorithms, and the final section includes some conclusions and future works.

2 Conventional and proposed computation of HPs

This section presents the conventional computation methods of HPs and introduces the proposed algorithms for fast and stable HPs computation.

2.1 Conventional computation of HPs

The HPs of order n and discrete variable x are defined as follows [46]:

$$h_n^{(\alpha,\beta)}(x;N) = \frac{(-1)^n(\beta+1)_n(N-n)_n}{n!} F_2\left(\begin{matrix}-n,-x,n+1+\alpha+\beta \\ \beta+1,1-N\end{matrix};1\right) \tag{1}$$

where $n, x = 0, 1, \ldots, N-1$, $\alpha > -1$, $\beta > -1$, and ${}_3F_2(.)$ is a hyper-geometric function defined as:

$$F_2(a_1,a_2,a_3;b_1,b_2;z) = \sum_{k=0}^{\infty}\frac{(a_1)_{\mathrm{k}}(a_2)_{\mathrm{k}}(a_3)_{\mathrm{k}}z^k}{(b_1)_{\mathrm{k}}(b_2)_{\mathrm{k}}k!}; (a)_k = a(a+1)(a+2)\ldots(a+k-1) \tag{2}$$

Using Eqs. (1) and (2), HPs can be explicit as follows:

$$h_n^{(\alpha,\beta)}(x;N) = A_n^{(\beta)}\sum_{k=0}^{n} B_{n,k}^{(\alpha,\beta)}\langle x\rangle_k \tag{3}$$

with

$$A_n^{(\beta)} = \frac{(-1)^n(\beta+1)_n(N-n)_n}{n!} \tag{4}$$

$$B_{n,k}^{(\alpha,\beta)} = \frac{(-n)_k(n+\alpha+\beta+1)_k}{(-1)^k k!(\beta+1)_k(1-N)_k} \tag{5}$$

According to [7], $\langle x\rangle_k$ is defined as follows:

$$\langle x\rangle_k = (-1)^k(-x)_k = \sum_{i=0}^{k} S(k,i)x^i \text{ with } k\geq 1 \text{ and } \langle x\rangle_0 = 1 \tag{6}$$

with$S(k, i)$ are the Stirling numbers of the first kind, obtained by the following recurrence relation [40]:

$$\begin{array}{l} S(k,i) = S(k-1,i-1)-(k-1)S(k-1,i) \\ \text{with } S(k,0) = S(0,i) = 0 \ \ \forall k\geq 1 \ \ \forall i\geq 1 \ \text{ and } \ S(0,0) = 1 \end{array} \tag{7}$$

HPs satisfies the following orthogonality relation [23]:

$$\sum_{x=0}^{N-1} h_n^{(\alpha,\beta)}(x;N)h_m^{(\alpha,\beta)}(x;N)\omega(x) = d_n^2\delta_{nm} \tag{8}$$

where δ_{nm} denotes the delta of Kronecker, $\omega(x)$ is the weight function, and d_n^2 represents the square norm function. These functions are defined as follows [29]:

$$\omega(x) = \frac{\Gamma(N+\alpha-x)\Gamma(\beta+1+x)}{\Gamma(N-x)\Gamma(x+1)} \tag{9}$$

$$d_n^2 = \frac{(\alpha+n)!(\beta+n)!(\alpha+\beta+n+1)_N}{(\alpha+\beta+2n+1)(N-n-1)!n!} \tag{10}$$

When the order n of HPs is high, the computation of these polynomials becomes unstable and expansive in terms of computation time because of gamma and factorial functions that define HPs (Eq. (1)) [9, 27]. To improve the computation speed and the numerical stability of high-order HPs, researchers use the recurrence relations with respect to the order n or with respect to the variable x [46, 50]. In order to derive the recurrence relations of the normalized HPs, the following form is used [46]:

$$\widetilde{h}_n^{(\alpha,\beta)}(x) = h_n^{(\alpha,\beta)}(x)\sqrt{\frac{\omega(x)}{\rho(n)}} \tag{11}$$

The following three-term recurrence relation allows the computation of the normalized with respect to the order n [50]:

Table 1 A, B, C, D and E coefficients of $\widetilde{h}_n^{(\alpha,\beta)}(x)$

A	$\sqrt{n(\alpha+\beta+n)}\ (\alpha+n)\ (\beta+n)\ (N-n)\ \frac{(\alpha+\beta+n+N)}{(a+\beta+2n)}$
B	$\frac{\alpha-\beta+2N-2}{4}-\left(\beta^2-\alpha^2\right)\ \frac{(\beta+\alpha+2N)}{4(\alpha+\beta+2n-2)}\ (\alpha+\beta+2n)$
C	$\sqrt{(\alpha+\beta+2n)}\ ^2-1$
D	$\sqrt{(n-1)}\ (\alpha+\beta+n-1)\ (\alpha+n-1)\ (\beta+n-1)\ (N-n+1)\ (\alpha+\beta+n-1)\ \frac{+N)}{(\alpha+\beta+2n-2)}$
E	$\sqrt{\frac{\alpha+\beta+2n+1}{\alpha+\beta+2n-3}}$

$$\widetilde{h}_n^{(\alpha,\beta)}(x)=\frac{(x-B)C}{A}\widetilde{h}_{n-1}^{(\alpha,\beta)}(x)-\frac{DE}{A}\times\widetilde{h}_{n-2}^{(\alpha,\beta)}(x) \tag{12}$$

Where the coefficients *A, B, C, D* and *E* are defined in Table 1.

The initial values of HPs with respect to the order *n* are computed as follows [50]:

$$\widetilde{h}_0^{(\alpha,\beta)}(x)=\sqrt{\frac{\omega(x)}{\rho(0)}} \tag{13}$$

$$\widetilde{h}_1^{(\alpha,\beta)}(x)=[-(\beta+1)(N-1)+(\alpha+\beta+2)x]\sqrt{\frac{\omega(x)}{\rho(1)}} \tag{14}$$

The recursive computation of HPs is also possible with respect to the variable *x* by using the following three-term recurrence formula [3]:

$$\widetilde{h}_n^{(\alpha,\beta)}(x,N)=\frac{1}{\sigma(x-1)+\tau(x-1)}\times\left[\begin{array}{l}(2\sigma(x-1)+\tau(x-1)-\lambda_n)\sqrt{\frac{\omega(x)}{\omega(x-1)}}\widetilde{h}_n^{(\alpha,\beta)}(x-1,N)\\-\sigma(x-1)\sqrt{\frac{\omega(x)}{\omega(x-2)}}\widetilde{h}_n^{(\alpha,\beta)}(x-2,N)\end{array}\right]$$

with

$$\sigma(x)=x(N+\alpha-x),\tau(x)=(\beta+1)(N-1)-(\alpha+\beta+2)x\ \text{ and }\ \lambda_n=n(\alpha+\beta+n+1) \tag{15}$$

The initial values of HPs with respect to the variable *x* are given by [50]:

$$\widetilde{h}_n^{(\alpha,\beta)}(0)=(1-N)_n\binom{n+\beta}{n}\sqrt{\frac{\omega(0)}{\rho(n)}} \tag{16}$$

$$\widetilde{h}_n^{(\alpha,\beta)}(1)=\frac{(n+\beta+1)(N-n-1)-n(N+\alpha-1)}{(\beta+1)(N-1)}\times\sqrt{\frac{\omega(1)}{\omega(0)}}\widetilde{h}_n^{(\alpha,\beta)}(0) \tag{17}$$

The recursive computation method of HPs provided in [46, 50] is limited by the problem of numerical instability of HPs initial values, either with respect to the order n or with respect to the variable x. This problem comes from $\omega(x)$ and $\rho(n)$ functions, which are defined from gamma and factorial functions. Noting that $\Gamma(172) = Inf$ under Matlab, Python, C++, etc. This Overflow problem leads to numerical instabilities of HPs when the values of the order n, the variable x and the parameters α and β become high. Therefore, the analysis of large-size signals and images by HPs is not possible.

To overcome the instability problem of high-order HPs, new algorithms are introduced in the next section.

2.2 Proposed computation of high-order HPs

2.2.1 HPs computation with respect to the order n

To avoid the numerical instability of the initial values $\widetilde{h}_0^{(\alpha,\beta)}(x)$, Eq. (13) is used to develop the following formula:

$$\widetilde{h}_0^{(\alpha,\beta)}(x) = \begin{cases} \sqrt{\dfrac{(N-x)(\beta+x)}{x(N+\alpha-x)}}\widetilde{h}_0^{(\alpha,\beta)}(x-1) \ ; & x > 0 \\ \sqrt{\dfrac{(\alpha+\beta+1)\Gamma(\alpha+1+\beta)\Gamma(N+\alpha)}{\Gamma(\alpha+1)\Gamma(N+\alpha+\beta+1)}} & ; \quad x = 0 \end{cases} \tag{18}$$

To eliminate $\Gamma(x)$ function from the term $\widetilde{h}_0^{(\alpha,\beta)}(0)$ (Eq. (18)), the following recursive formula is used:

$$\widetilde{h}_0^{(\alpha,\beta)}(0) = \widetilde{h}_i^{(\alpha,\beta)} = \begin{cases} 1; \quad i = 1 \\ \sqrt{\dfrac{(i+\alpha-1)}{(i+\alpha+\beta)}}\widetilde{h}_{i-1}^{(\alpha,\beta)}; \quad i = 2, 3, \ldots, N \end{cases} \tag{19}$$

Based on Eqs. (13)–(14), the expression of $\widetilde{h}_1^{(\alpha,\beta)}(x)$ can be expressed as follows:

$$\widetilde{h}_1^{(\alpha,\beta)}(x) = [(\alpha+\beta+2)x-(\beta+1)(N-1)]\sqrt{\frac{(\alpha+\beta+3)}{(\alpha+1)(\beta+1)(N-1)(N+\alpha+\beta+1)}}\widetilde{h}_0^{(\alpha,\beta)}(x) \tag{20}$$

The method thus developed to compute HPs initial values via Eqs. (18)–(20) is numerically stable since it is independent from gamma and factorial functions. Algorithm 1 summarizes the main steps of the proposed method for HPs computation with respect to the order n.

Algorithm 1 Proposed algorithm for HPs computation with respect to the order *n*.

Inputs :	*N*: The maximum value of the variable *x*, *Nmax*: The maximum order of HPs, α and β : HPs Parameters
Outputs:	$h_{1:N_{\max}}^{(\alpha,\beta)}(1:N)$: HPs Matrix
Step 1	Compute $h_0^{(\alpha,\beta)}(0)$ using Eq. (19)

Step 2	**for** *x=1:N-1* **do**
	Compute $h_0^{(\alpha,\beta)}(x)$ using Eq. (18)
	end for

Step 3	**for** *x=0:N-1* **do**
	Compute $h_1^{(\alpha,\beta)}(x)$ using Eq. (20)
	end for

Step 4	**for** *n=2:Nmax-1* **do**
	for *x=0:N-1* **do**
	Compute $h_n^{(\alpha,\beta)}(x)$ using Eq. (12)
	end for
	end for

The following subsection includes the proposed computation of HPs with respect to the variable *x*.

2.2.2 HPs computation with respect to the variable *x*

To ensure the numerical stability of HPs initial values $\tilde{h}_n^{(\alpha,\beta)}(0)$ and $\tilde{h}_n^{(\alpha,\beta)}(1)$, Eqs. (16)–(17) are extended to obtain the formulas below.

$$\tilde{h}_n^{(\alpha,\beta)}(0) = -\sqrt{\frac{(\alpha+\beta+2n+1)(\alpha+\beta+n)(N-n)(n+\beta)}{n(\alpha+n)(\alpha+\beta+N+n)(\alpha+\beta+2n-1)}}\tilde{h}_{n-1}^{(\alpha,\beta)}(0) \qquad (21)$$

with $\tilde{h}_0^{(\alpha,\beta)}(0)$ is given by Eq. (18).

$$\tilde{h}_n^{(\alpha,\beta)}(1) = ((n+\beta+1)(N-n-1)-n(N+\alpha-1))\sqrt{\frac{1}{(\beta+1)(N-1)(N+\alpha-1)}}\tilde{h}_n^{(\alpha,\beta)}(0)(22)$$

For $x > 1$, the numerical stability of $\tilde{h}_n^{(\alpha,\beta)}(x)$ is achieved by eliminating $\omega(x)$function from Eq. (15) as follows:

$$
\begin{aligned}
&\tilde{h}_n^{(\alpha,\beta)}(x) = A \times \tilde{h}_n^{(\alpha,\beta)}(x-1) + B \times \tilde{h}_n^{(\alpha,\beta)}(x-2)\\
&A = \frac{2(x-1)(N+\alpha-x+1)+(\beta+1)(N-1)-(\alpha+\beta+2)(x-1)-n(\alpha+\beta+n+1)}{(x-1)(N+\alpha-x+1)+(\beta+1)(N-1)-(\alpha+\beta+2)(x-1)}\sqrt{\frac{(\beta+x)(N-x)}{(N+\alpha-x)x}}\\
&B = -\frac{(x-1)(N+\alpha-x+1)}{(x-1)(N+\alpha-x+1)+(\beta+1)(N-1)-(\alpha+\beta+2)(x-1)}\sqrt{\frac{(\beta+x)(N-x)}{(N+\alpha-x)x}\frac{(\beta+x-1)(N-x+1)}{(N+\alpha-x+1)(x-1)}}
\end{aligned}
\tag{23}
$$

The fundamental steps used for HPs computation with respect to the variable x are presented in algorithm 2.

Algorithm 2 Proposed algorithm for HPs computation with respect to the variable x.

Inputs :	N: The maximum value of the variable x; *Nmax*: The maximum order of HPs; α and β : HPs Parameters
Outputs:	$h_{1:N_{\max}}^{(\alpha,\beta)}(1:N)$: matrix of HPs
Step 1	Compute $h_0^{(\alpha,\beta)}(0)$ using Eq. (19)

Step 2	**for** *n=1:Nmax-1* **do**
	Compute $h_n^{(\alpha,\beta)}(0)$ using Eq. (21)
	end for

Step 3	**for** *n=0:Nmax-1* **do**
	Compute $h_n^{(\alpha,\beta)}(1)$ using Eq. (22)
	end for

Step 4	**for** *n=0:Nmax-1* **do**
	for *x=2:N-1* **do**
	Compute $h_n^{(\alpha,\beta)}(x)$ using Eq. (23)
	end for
	end for

It is important to mention that the computation of HPs initial values by Eqs. (20)–(21) presents the Underflow problem for high values of the order n. Indeed, Table 2 illustrates an example of this problem.

From Table 2 it can be seen that the values of $\tilde{h}_n^{(\alpha,\beta)}(0)$ become zero (Underflow) when $n >$ 2558. This problem limits the recursive computation of high-order HPs. To overcome this problem, new method is proposed (Fig. 1) for stable computation of high-order HPs with

Table 2 $\tilde{h}_n^{(\alpha,\beta)}(0)$ values for $N=$ 4000, $\alpha=\beta=100$, $n=0{:}2559$ and $x=0$

Order n	2557	2558	2559
$\tilde{h}_n^{(\alpha,\beta)}(0)$ value	−1.50000000000000e-323	4.90000000000000e-324	0

respect to the variable *x*. The details of this method are described in algorithm 3. Noting the proposed method is inspired from the method presented in [1], which is used for stable computation of high-order Tchebichef polynomials.

Algorithm 3 Proposed Algorithm for HPs computation with respect to the variable *x*.

Inputs :	*N*: The maximum value of the variable *x*; *Nmax*: The maximum order of HPs; α and β : HPs Parameters
Outputs:	$h_{1:N_{\max}}^{(\alpha,\beta)}(1:N)$: matrix of HPs
Step 1	Compute $h_0^{(\alpha,\beta)}(0)$ using Eq. (19)
Step 2	**for** *x=1:N-1* **do**
	Compute $h_0^{(\alpha,\beta)}(x)$ using Eq. (18) **end for**

Step 3	**for** *x=0:N-1* **do**
	Compute $h_1^{(\alpha,\beta)}(x)$ using Eq. (20)
	end for

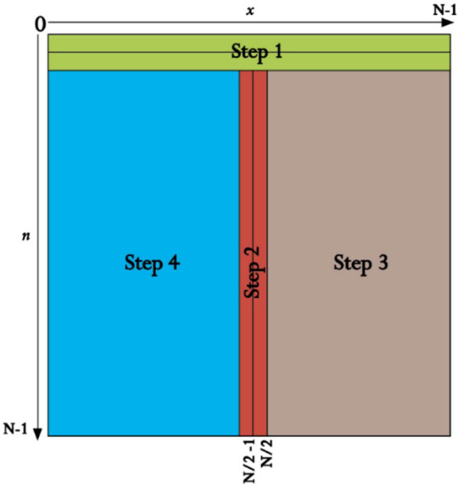
Step 4	**for** *n* = 2 : *Nmax - 1* **do**
	for *x=N/2-1* : *N/2* **do**
	Compute $h_n^{(\alpha,\beta)}(x)$ using Eq. (12) for
	end for
	end for

Step 5	**for** *n* = 2:*Nmax- 1* **do**
	for *x=N/2+1* : *N-1* **do**
	Compute $h_n^{(\alpha,\beta)}(x)$ using Eq. (23) for
	end for
	end for

Step 6	**for** *n* = 2:*Nmax- 1* **do**
	for *x=N/2-2* :-1:0 **do**
	Compute $h_n^{(\alpha,\beta)}(x)$ by the following symmetry relation for $\alpha=\beta$:
	$h_n^{(\alpha,\beta)}(x)=(-1)^n h_n^{(\alpha,\beta)}(\mathrm{N}-x-1)$
	end for
	end for

Since the computation of HPs using the proposed algorithms is recursive, this leads to the appearance and propagation of rounding error. This problem is discussed by Camacho-bello et al. in [5] when computing high-order Tchebichef polynomials. To overcome the encountered problem for Tchebichef, Meixner and Hahn polynomials, the modified Gram-Schmidt orthonormalization process (GSOP) is used [5, 8, 9, 12]. However, the correction of numerical instability by GSOP is very time-consuming because GSOP involves a complex calculation process [5, 12]. This problem limits the fast reconstruction of large-size signals and images by high-order HPs. Indeed, Table 3 shows a comparison in terms of execution time in second (*s*) by using algorithm 1, and GSOP method presented in [9]. The computation of HPs by these

Fig. 1 Steps of the proposed method for HPs computation with respect to the variable x



methods is performed 100 times, and then the average execution time is computed and illustrated in Table 3. It is worth mentioning that all paper tests are carried out under Matlab R2019a installed on PC of 2.4 GHz processor and 4 GB of RAM.

The results presented in Table 3 and 4 clearly indicate that the GSOP method significantly increases the computation time of HPs compared to the proposed methods (algorithms 1 and 3). This is why a new numerical stability method is introduced in the following section.

2.3 Proposed method for ensuring the numerical stability of higher-order HPs

The proposed method is based on a numerical analysis of the polynomial values in HPs matrix. Indeed, Fig. 2 present 2D plot of HPs matrix computed by algorithm 1 for $n, x = 0:512$, $\alpha = 5$, $\beta = 180$. This matrix is displayed in two parts: the first one that is shown in Fig. 2 (a) displays HPs values that are belong to the orthonormalization domain $(\widetilde{h}_n^{(\alpha,\beta)}(x) \in [-1, 1])$), while the second part displayed in Fig. 2 (b) shows the unstable values $(\widetilde{h}_n^{(\alpha,\beta)}(x) \notin [-1, 1])$, and Fig. 2 (c) shows HPs matrix computed by GSOP method [9].

From Fig.1 (b), it is noticeable that the unstable values of HPs appear when $n \rightarrow N$ (high polynomial order). These values are positioned in the blue colored regions called “instability regions”. After using GSOP method (Fig. 2 (c)), one can observe that the previously observed instable values become extremely low and tend towards zero. Indeed, Fig.3 shows the intervals of HPs values computed by GSOP method when $n \rightarrow N$.

Table 3 Comparison of HPs execution time for n, $x = 0$:512 using Algorithm 1 and GSOP method [9]

HPs order		50	100	200	512
Execution time in seconds (s) using:	Algorithm 1	0.0022	0.0056	0.0095	0.0458
	GSOP method [9]	0.0582	0.4301	3.2701	52.2399

Table 4 The execution time of HPs (with $n, x = 0:2000$ and $\alpha = \beta = 50$) by using Algorithm 3 (with the stability condition), and by GSOP method [9]

HPs orders		500	1000	2000
Execution time in seconds (s) by using:	The Proposed algorithm 3	0.1390	0.8196	7.3199
	GSOP method [9]	47.8710	2.7514e+3	1.2578e+6
ETIR (%)		**72.84%**	**99.99%**	**99.99%**

From Fig. 3, it can be noticed that the values of $\left|\tilde{h}_n^{(\alpha,\beta)}(x)\right|$ gradually decrease and tend towards zero when $n \rightarrow N$. Therefore, it can be concluded that the numerical instability occurs only when HPs values become very low and toward to zero ($\left|\tilde{h}_n^{(\alpha,\beta)}(x)\right| \leq 10^{-7}$). According to the conducted numerical analysis, appropriate "stability conditions" are introduced during the recursive computation of HPs to ensure the numerical stability of these polynomials. Indeed, the following stability condition is inserted after the three-term recurrence relation of HPs with respect to the order n.

$$\hat{h}_n^{(\alpha,\beta)}(x) = \begin{cases} 0 \quad if \quad \left(\left|\tilde{h}_{n-1}^{(\alpha,\beta)}(x)\right| < 10^{-7}\right) \& \left(\left|\tilde{h}_n^{(\alpha,\beta)}(x)\right| > \left|\tilde{h}_{n-1}^{(\alpha,\beta)}(x)\right|\right) \\ \tilde{h}_n^{(\alpha,\beta)}(x) \;\; \text{with} \;\; n > N_{\max}/8 \text{ and } x = 0, 1, 2, \dots, N-1 \end{cases} \tag{24}$$

Note that this condition is activated from the order $n = N_{\max}/8$ (lower order) where N_{max} is the maximum order of HPs, this ensures that all unstable values are detected and eliminated (replaced by zero).

The following stability condition is implemented after the computation of each HPs value using the three-term recurrence relation with respect to the variable *x:*

$$\hat{h}_n^{(\alpha,\beta)}(x) = \begin{cases} 0 \quad if \quad \left(\left|\tilde{h}_n^{(\alpha,\beta)}(x-1)\right| < 10^{-7}\right) \& \left(\left|\tilde{h}_n^{(\alpha,\beta)}(x)\right| > \left|\tilde{h}_n^{(\alpha,\beta)}(x-1)\right|\right) \\ \tilde{h}_n^{(\alpha,\beta)}(x) \;\; \text{with} \; x > N/8 \text{ and } n = 0, 1, \dots, N-1 \end{cases} \tag{25}$$

To evaluate the performance of the proposed Algorithms that including the stability conditions, HPs are computed by these Algorithms up to the order $n = 8000$ with $x = 0:8000$ and $\alpha = \beta =$ 50. Fig. 4 shows 2D plot of HPs matrices computed by Algorithms 1 and 3. The obtained results show that the computed HPs are stable since the polynomial values belong to the orthonormality domain [−1, 1], which justifies the efficiency of the proposed Algorithms for the stable computation of high-order HPs.

The following test presents a comparison in terms of execution time using the proposed method (algorithm 3) and the GSOP method presented in [9]. The execution time improvement ratio (ETIR (%)) defined by $ETIR\ (\%) = (1 - T\,1/T\,2) \times 100$ [23] is used as a comparison criterion, where $T\ 1$ is the execution time consumed by the proposed method, and $T\ 2$ represents the execution time consumed by GSOP method [9]. The obtained results which are displayed in Table 4 demonstrate that.

the proposed method (algorithm 3 with the stability condition) is very fast compared to the GSOP method since $ETIR \geq 99\%$. This is especially true when the polynomial order n is high.

It is worth mentioning that the computation time of HPs is significantly reduced when using the following HPs symmetry relation defined for $\alpha = \beta$ [29]:

$$\widetilde{h}_n^{(\alpha,\beta)}(x) = (-1)^n \widetilde{h}_n^{(\alpha,\beta)}(\mathrm{N}-1-x) \tag{26}$$

It is also important to note that Krawtchouk (KPs) and Tchebichef (TPs) polynomials are special cases of HPs. Indeed, KPs ($\widetilde{k}_n^{(p)}(x)$) are obtained from HPs when $\alpha = pt$ and $\beta = (1-p)t$ with $t \to \infty$ [29]. In the case $\alpha = \beta = 0$, HPs are reduced to TPs ($\widetilde{t}_n(x)$) [22].

In the following sections, HPs will be used as kernel polynomials to define both Hahn moments (HMs) and Hahn moment invariants (HMIs) designed for large-size signal and image analysis.

3 Hahn moments

This section briefly presents the theoretical background of Hahn moments (HMs) in the 1D and 2D domains.

The 1D HMs of order n are defined for a 1D-signal function $f(x)$ by the following relation:

$$HM_n = \sum_{x=0}^{N-1} \widetilde{h}_n^{(\alpha,\beta)}(x) f(x) = P_n^T \times f \quad \text{with} \quad \text{with} \quad P_n = \left[\widetilde{h}_0^{(\alpha,\beta)}(x), \widetilde{h}_1^{(\alpha,\beta)}(x), \ldots, \widetilde{h}_{N-1}^{(\alpha,\beta)}(x)\right]^T \tag{27}$$

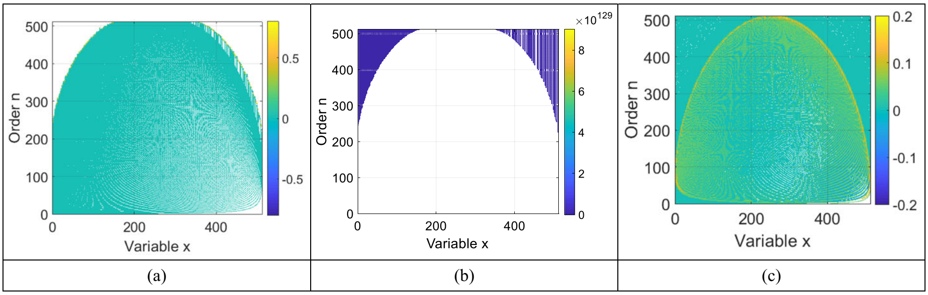
The reconstructed signal $\widehat{f}(x)$ is computed from the 1D HMs as follows:

$$\widehat{f}(x) = \sum_{x=0}^{N-1} H_n \widetilde{h}_n^{(\alpha,\beta)}(x) = P_n \times HM_n \tag{28}$$

The reconstruction error can be measured by the Mean Square Error (MSE) criterion, which is defined as follows:

$$MSE = \frac{1}{N} \times \sum_{x=0}^{N-1} \left(f(x) - \widehat{f}(x)\right)^2 \tag{29}$$

where $f(x)$ is the original signal, and $\widehat{f}(x)$ represents the reconstructed one.

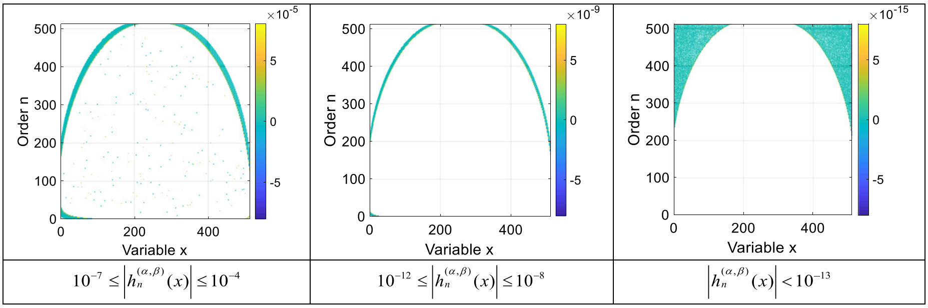


(a)

(b)

(c)

Fig. 2 2D plot of HPs matrix computed by Algorithm 1 with **(a)** $\widetilde{h}_n^{(\alpha,\beta)}(x) \in [-1, 1]$, **(b)** $\widetilde{h}_n^{(\alpha,\beta)}(x) \notin [-1, 1]$, and **(c)** HPs matrix computed by GSOP method [9] for $n, x = 0:512.$ and $\alpha = 5$, $\beta = 180$



$10^{-7} \leq \left\lvert h_n^{(\alpha,\beta)}(x) \right\rvert \leq 10^{-4}$	$10^{-12} \leq \left\lvert h_n^{(\alpha,\beta)}(x) \right\rvert \leq 10^{-8}$	$\left\lvert h_n^{(\alpha,\beta)}(x) \right\rvert < 10^{-13}$

Fig. 3 The numerical behavior of HPs values computed by GSOP for $n, x \rightarrow 500$ and $\alpha = 5$, $\beta = 180$

For a digital image $f(x, y)$ of size $N \times M$, the 2D HMs are defined by [50]:

$$HM_{nm} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\tilde{h}_n^{(\alpha,\beta)}(x)\tilde{h}_m^{(\alpha,\beta)}(y) \tag{30}$$

with $n, m = 0, 1, 2, \ldots, N-1$, $x \in [0\ldots \mathrm{N}-1]$, and $y \in [0\ldots M-1]$,.

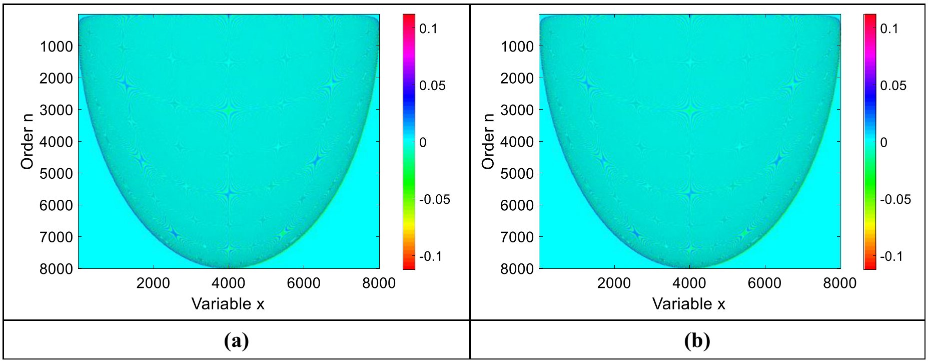
The computation time of HMs is considerably reduced by using the following matrix formula [3]:

$$HM_{nm} = {H_n}^T \times f_{xy} \times H_m \tag{31}$$

where $f_{xy} = \{f(x,y)\}_{x,y=0}^{x=N-1,y=M-1}$ denotes the image function with H_n^T and H_m are HPs matrices defined by: $H_n^T = \left[\left\{\tilde{h}_n^{(\alpha,\beta)}(x)\right\}_{n,x=0}^{x=N-1,y=N-1}\right]^T$, $H_m = \left\{\tilde{h}_m^{(\alpha,\beta)}(x)\right\}_{m,y=0}^{m=M-1,y=M-1}$.

The inverse transformation of 2D HMs allows to obtain the reconstructed image via the following formula [3]:

$$\hat{f}(x,y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \tilde{h}_n^{(\alpha,\beta)}(x)\tilde{h}_m^{(\alpha,\beta)}(y)HM_{mn} \tag{32}$$



(a)	**(b)**

Fig. 4 **(a)-(b)** 2D plots of of HPs matrices computed successively by Algorithms 1 and 3 including the stability conditions with $n, x = 0 : 8000$ and $\alpha = \beta = 50$.

Table 5 The average time for computing HMIs by using the proposed method (algorithm 5), and the conventional one (algorithm 4)

HMIs orders		(5,5)	(10,10)	(15,15)	(20,20)
Execution time in seconds (s) using:	The conventional method (algorithm 4) [36]	0.2034	0.5067	1.0305	2.6209
	The proposed method (algorithm 5)	0.0636	0.1280	0.2466	0.5281
ETIR (%)		**68.71**	**74.73**	**76.07**	**79.85**

The matrix formulation of the inverse 2D HMs is given by:

$$\widehat{f}_{xy} = H_n \times HM_{nm} \times H_m^T \tag{33}$$

where $\widehat{f}_{xy} = \left\{\widehat{f}(x,y)\right\}_{x,y=0}^{x=N-1,y=M-1}$ is the reconstructed image of size $N \times M$.

In order to quantify the reconstruction error by HMs, the MSE criterion is used as follows:

$$MSE = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left(f(x,y) - \widehat{f}\left(x,y\right)\right)^2 \tag{34}$$

The Peak Signal-to-Noise Ratio (PSNR) criterion in decibels (dB) is also used to quantify the reconstruction error. This criterion is defined from MSE as:

$$PSNR = 10\log_{10}\left(\frac{peak^2}{MSE}\right) \tag{35}$$

where *peak* is the maximum value of the 2D image function.

4 Hahn moment invariants

This section presents both the conventional and the proposed method for deriving Hahn moment invariants (HMIs) from geometric moment invariants (GMIs).

4.1 The conventional computation of HMIs

The following relation is used to compute the geometric moments (GMs) of *f (x, y)* image of size $N \times M$ [14]:

$$GM_{nm} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x^n y^m f(x,y) \tag{36}$$

The central geometric moments denoted μ_{nm} are derived from GMs by the following formula [35]:

$$\mu_{nm} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left(x - \overline{x}\right)^n \left(y - \overline{y}\right)^m f(x,y) \text{ with } \overline{x} = \frac{GM_{10}}{GM_{00}} \text{ and } \overline{y} = \frac{GM_{01}}{GM_{00}} \tag{37}$$

The set of geometric moment invariants (GMIs) of order *(n, m)*, denoted V_{nm}, which are independent of translation, scaling and rotation are defined by the following relation [35]:

$$V_{nm} = {GM_{00}}^{-\gamma} \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left\{ \begin{array}{l} \left[\left(x - \overline{x} \right) \cos\theta + \left(y - \overline{y} \right) \sin\theta \right]^{i} \\ \times \left[\left(y - \overline{y} \right) \cos\theta - \left(x - \overline{x} \right) \sin\theta \right]^{j} \end{array} \right\} f(x,y) \tag{38}$$

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. (39), the values of the angle θ are limited to $-\pi/4 \leq \theta \leq \pi/4$. For angle values in different ranges, the article [40] is recommended.

HMs given by Eq. (31) can be expressed in terms of HPs (Eq. (3)) as follows [36]:

$$HM_{nm} = \frac{A_n^{(\alpha,\beta)} A_m^{(\alpha,\beta)}}{\sqrt{d_n^2 d_m^2}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} h_n^{(\alpha,\beta)}(x) h_m^{(\alpha,\beta)}(y) \widetilde{f}(x,y) \tag{39}$$

where $\widetilde{f}(x,y) = \sqrt{\omega(x)\omega(y)} f(x,y)$ is the normalized image.

Based on Eqs. (3), (30) and (39), HMs can be expressed in terms of GMs as follows [36]:

$$HM_{nm} = \frac{A_n^{(\beta)} A_m^{(\beta)}}{\sqrt{d_n^2 d_m^2}} \sum_{j=0}^{m} \sum_{i=0}^{n} B_{mj}^{(\alpha,\beta)} B_{ni}^{(\alpha,\beta)} \sum_{l=0}^{j} \sum_{k=0}^{i} S(j,l) S(i,k) GM_{kl} \tag{40}$$

The substitution of GM_{nm} by V_{nm} allows to derive HMIs as follows [36]:

$$HMI_{nm} = \frac{A_n^{(\alpha,\beta)} A_m^{(\alpha,\beta)}}{\sqrt{d_n^2 d_m^2}} \sum_{j=0}^{m} \sum_{i=0}^{n} B_{mj}^{(\alpha,\beta)} B_{ni}^{(\alpha,\beta)} \times \sum_{l=0}^{j} \sum_{k=0}^{i} S(j,l) S(i,k) V_{kl} \tag{41}$$

Algorithm 4 summarizes the conventional steps of the HMIs derivation process.

Algorithm 4 The conventional method for HMIs computation [36].

Inputs :	$\widetilde{f}(x,y)$: normalized image of size $N \times M$; (n,m) : HMIs order ; α and β : HPs parameter values
Outputs:	HMI_{nm} : HMIs of order *(n, m)*
Step 1	Compute $A_n^{(\beta)}$ using Eq. (4)
Step 2	Compute $B_{nk}^{(\alpha,\beta)}$ using Eq. (5)
Step 3	Compute d_n^2 using Eq. (10)
Step 4	Compute $S(k,i)$ using Eq. (7)
Step 5	Compute GM_{nm} using Eq. (36)
Step 6	Compute V_{nm} using Eq. (38)
Step 7	Compute HMI_{nm} of (n,m) *-th* order using Eq.(41)

The conventional computation of HMIs via algorithm 4 is numerically instable because of $A_n^{(\beta)}$, $B_{nk}^{(\alpha,\beta)}$, $\omega(x)$ and d_n^2 functions as they are defined in terms of gamma and factorial functions. In addition, this conventional computation is time-consuming due to the use of several discrete summations. These problems limit the computation of HMIs especially for large-size images. There is therefore a clear need for a considerable improvement in HMIs computation in terms of rapidity and numerical stability.

4.2 Proposed computation of HMIs

In this section, the proposed method for fast and stable computation of HMIs is presented. This method is mainly based on the use of matrix and recursive formulas.

In order to significantly accelerate the computation of GMs, the following matrix formulation can be used:

$GM_{nm} = X.F.Y^T$ with $X = \{x^n\}_{x,n=0}^{x,n=N-1}$, $Y = \{y^m\}_{y,m=0}^{y,m=M-1}$ and $F = \{f(x,y)\}_{x,y=0}^{x=N-1,y=M-1}$ (42)

The computation time of μ_{nm} can also be reduced through the use of the following formula:

$$\mu_{nm} = \overline{X}.F.\overline{Y}^T \quad \text{with} \quad \overline{X} = \left\{\left(x-\overline{x}\right)^n\right\}_{x=0,n=0}^{x=N-1,n=N-1} \quad \text{and} \quad \overline{Y} = \left\{\left(y-\overline{y}\right)^m\right\}_{y=0,m=0}^{x=M-1,m=M-1} \tag{43}$$

In a similar way, the computation of V_{nm} is speed up by using the following formulation:

$$V_{nm} = GM_{00}{}^{-\gamma} \sum_{i=0}^{n} \sum_{j=0}^{m} \left(A^i \odot B^j\right) \odot F \tag{44}$$

with $A = (x-\overline{x})\cos\theta + (y-\overline{y})\sin\theta$, $B = (y-\overline{y})\cos\theta - (x-\overline{x})\sin\theta$, $F = f(x,y)$,$x = 1:N$, $y = 1:M$, $\overline{x} = \frac{GM_{10}}{\,} GM_{00}$, $\overline{y} = \frac{GM_{01}}{\,} GM_{00}$ and the symbol $\odot$ denotes the Hadamard product [4].

The following recursive relations are developed to accelerate and stabilize the computation of HMIs. Indeed, $\omega(x)$ is computed as follows:

$$\omega(x) = \begin{cases} e^{\log\Gamma(N+\alpha)+\log\Gamma(\beta+1)-\log\Gamma(N)} & x = 0 \\ \dfrac{(\beta+x)(N-x)}{x(N+\alpha-x)}\omega(x-1) & x = 1,2,\ldots,N-1 \end{cases} \tag{45}$$

where $\log\Gamma(x)$ is the logarithmic gamma function [15]. This function is numerically stable for high values of the argument x. For example, $\log\Gamma(10^5) = 1.0513\mathrm{e}+06$ under Matlab.

The stable recursive computation of d_n^2 function is given by:

$$d_n^2 = \begin{cases} \dfrac{1}{(\alpha+\beta+1)} e^{\log\Gamma(\alpha+1)+\log\Gamma(\beta+1)+\log\Gamma(\alpha+\beta+N+1)-\log\Gamma(N)-\log\Gamma(\alpha+\beta+1)} & n = 0 \\ \dfrac{(\alpha+n)(\beta+n)(\alpha+\beta+n+N)(N-n)(\alpha+\beta+2n-1)}{n(\alpha+\beta+2n+1)(\alpha+\beta+n)} d_{n-1}^2 & n = 1,2,\ldots,N-1 \end{cases} \tag{46}$$

The fast and stable computation of $A_n^{(\beta)}$ and $B_{nk}^{(\alpha,\beta)}$ functions is defined as follows

$$A_n^{(\beta)} = \begin{cases} 1 & n = 0 \\ -\dfrac{(\beta+n)(N-n)}{n} A_{n-1}^{(\beta)} & n = 1,\ldots,N-1 \end{cases} \tag{47}$$

$$B_{nk}^{(\alpha,\beta)} = \begin{cases} 1 & k = 0 \\ -\dfrac{(n+\alpha+\beta+k)(n-k+1)}{(\beta+k)(N-k)k} B_{n(k-1)}^{(\alpha,\beta)} & k = 1,2,\ldots,n \end{cases} \tag{48}$$

The summary of the proposed method for fast and stable computation of HMIs is presented in algorithm 5.

Algorithm 5 The proposed method for fast and stable HMIs computation.

Inputs :	$\tilde{f}(x,y)$: normalized image of size $N \times M$; (n,m) : HMIs order ; α and β : HPs parameter values
Outputs:	HMI_{nm} : HMIs of order *(n, m)*
Step 1	Compute $A_n^{(\beta)}$ using Eq. (47)
Step 2	Compute $B_{nk}^{(\alpha,\beta)}$ using Eq. (48)
Step 3	Compute d_n^2 using Eq.(45)
Step 4	Compute $S(k,i)$ using Eq. (7)
Step 5	Compute GM_{nm} using Eq.(42)
Step 6	Compute V_{nm} using Eq. (44)
Step 7	Compute HMI_{nm} of (n,m) - th order using Eq. (41)

5 Experimental results and discussion

This section presents the simulation outcomes and the performed comparisons that validate the efficiency of the proposed algorithms in the field of large-size signal and image analysis. Initially, the numerical stability of the proposed algorithms for higher-order HPs computation

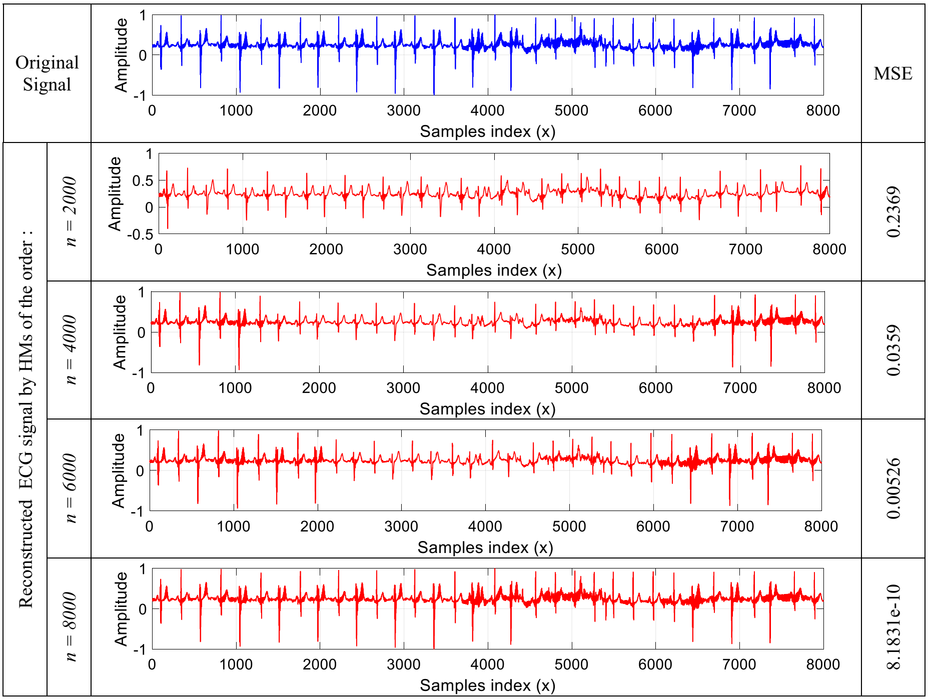


Fig. 5 Set of reconstructed ECG signals by HMs up to the order 8000 with the corresponding MSE values

is demonstrated through the reconstruction of large-size signals and images up to the last moment order. Then, the speed and numerical stability of the proposed method for HMIs computation is proved via the extraction of HMIs from large-size medical images.

5.1 Large-size signal reconstruction by higher-order HMs

In the first test, the efficiency of the proposed computation of high-order HPs with respect to the order n is demonstrated. For this purpose, HPs are computed by algorithm 1 for n, $x = 0:8000$ with $\alpha = \beta = 100$. Noting that the parameter values are selected empirically following the study carried out in [9]. This study shows that the parameters selection case $\alpha = \beta$ allows obtaining a minimal reconstruction error (MSE) in comparison to other cases ($\alpha \neq \beta$). Then, an "ECG" signal of size $N = 8000$ is selected from the

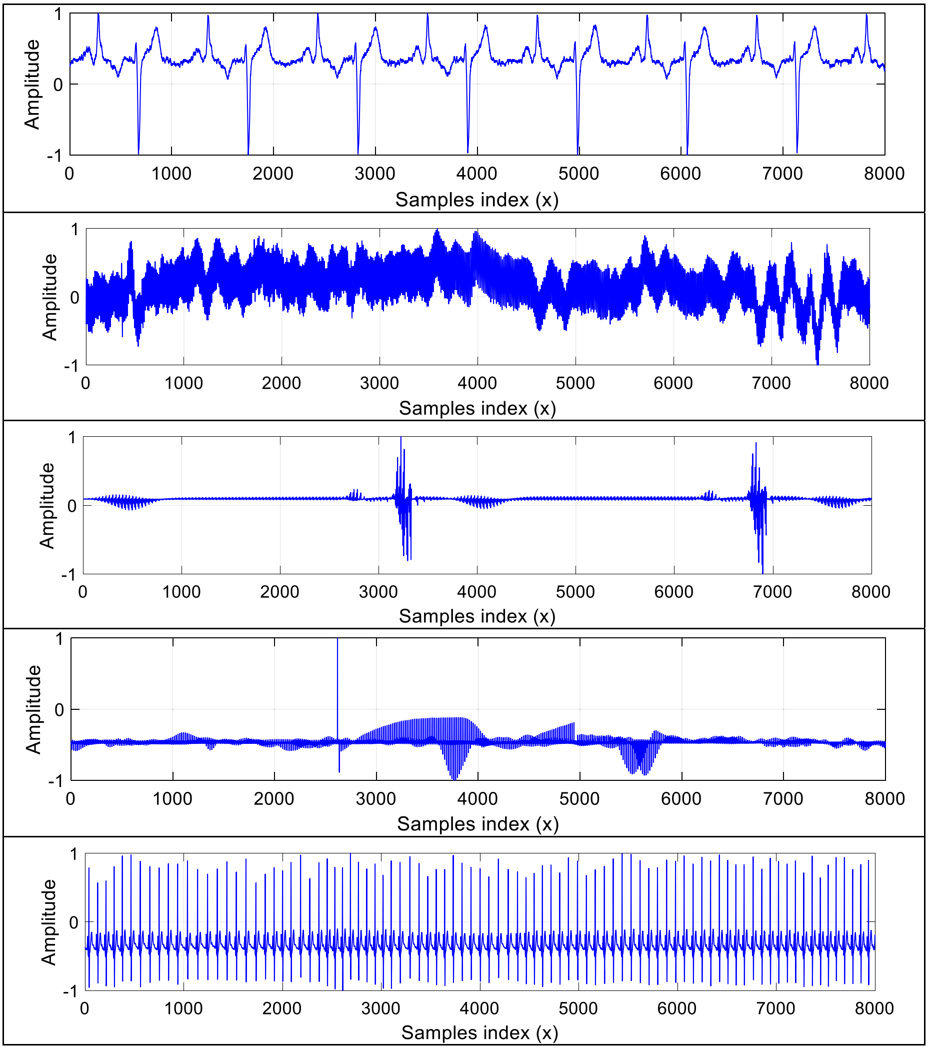


Fig. 6 Set of bio-signals selected from the database [25] that are used in the test

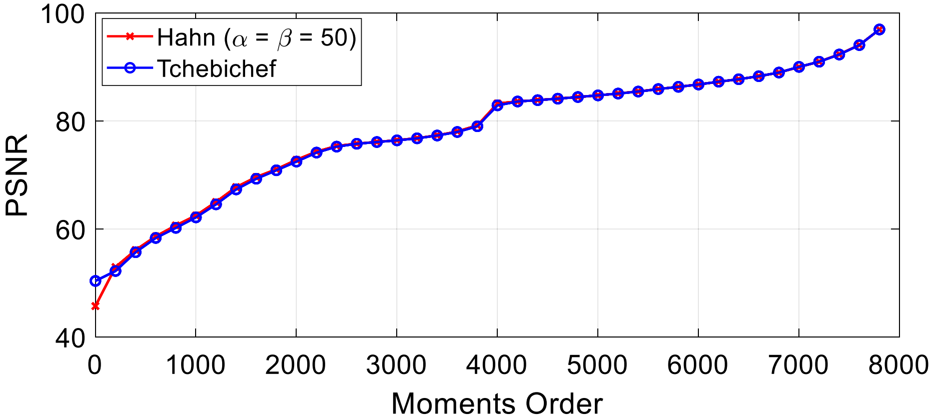


Fig. 7 Average of the PSNR values corresponding to the reconstructed bio-signals by HMs and TMs [5]

database [25], and reconstructed by HMs up to the last moment order ($n = 8000$). Figure 5 shows a set of reconstructed "ECG" signals with the corresponding MSE values. On one hand, the obtained results indicate that the MSE values decrease proportionally with the increase in HMs order until the last moment order. On the other hand, it is noted that the lower MSE value is reached at the last HMs order. These results prove the numerical stability of the proposed algorithm 1, and confirm the importance of high-order HMs in terms of precise signal reconstruction.

To further validate the efficiency of algorithm 1, five bio-signals of size $N = 8000$ (Fig. 6) are reconstructed by HMs (with HPs are computed by algorithm 1 for $\alpha = \beta = 100$). For comparison purposes, the test signal are also reconstructed by Tchebichef moments (TMs) [5]. Next, the average PSNR values corresponding to the reconstructed signals by both moment types are displayed in Fig. 7.

The PSNR values shown in Fig. 7 increase up to the last moment order, reflecting the efficiency and numerical stability of both TMs and HMs for the reconstruction of large-size signals. One should mention that the numerical stability of TMs is ensured via GSOP [5]. Whereas HMs numerical stability is provided by using the proposed method (stability condition). This makes HMs significantly faster to compute compared to TMs.

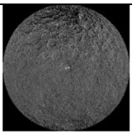
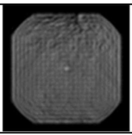
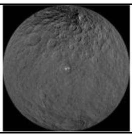
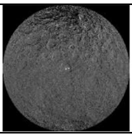
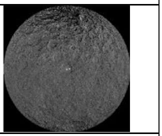
Original "PIA21906" image of size 4000×4000				
HMs order	100	1000	2000	4000
Reconstructed images				
MSE	0.0899	0.0163	0.0018	2.4562e-10
PSNR	58.5928	66.0184	75.4625	144.2282

Fig. 8 Set of reconstructed "PIA21906" images by HMs up to the order (4000, 4000) with MSE and PSNR values

5.2 Large-size image reconstruction by higher-order HMs

In this section, the performance of the proposed method for the computation of high-order HPs with respect to the variable x (algorithm 3) is illustrated. In the first test, the "PIA21906" test image of size 4000 × 4000 is selected from the database [31], and then reconstructed by HMs up to the last order (4000, 4000). Noting that the used HPs parameters values in all reconstruction tests of this section are $\alpha = \beta = 50$. Figure 8 shows a set of the reconstructed "PIA21906" images with the corresponding MSE and PSNR values. From this figure, it is noticeable that the visual similarity between the reconstructed image and the original one gradually improves until the last moment order (4000, 4000). It can also be noted that the best quality of the reconstructed image (lower MSE) is obtained at the last moment order. This is a clear indication on the efficiency of the proposed algorithm 3 for the stable reconstruction of large-sized images.

The second test presents a comparison in terms of reconstruction errors (MSE and PSNR) by using the moments of Tchebichef (TMs) [5], Krawtchouk (KMs) [47], Meixner (MMs)

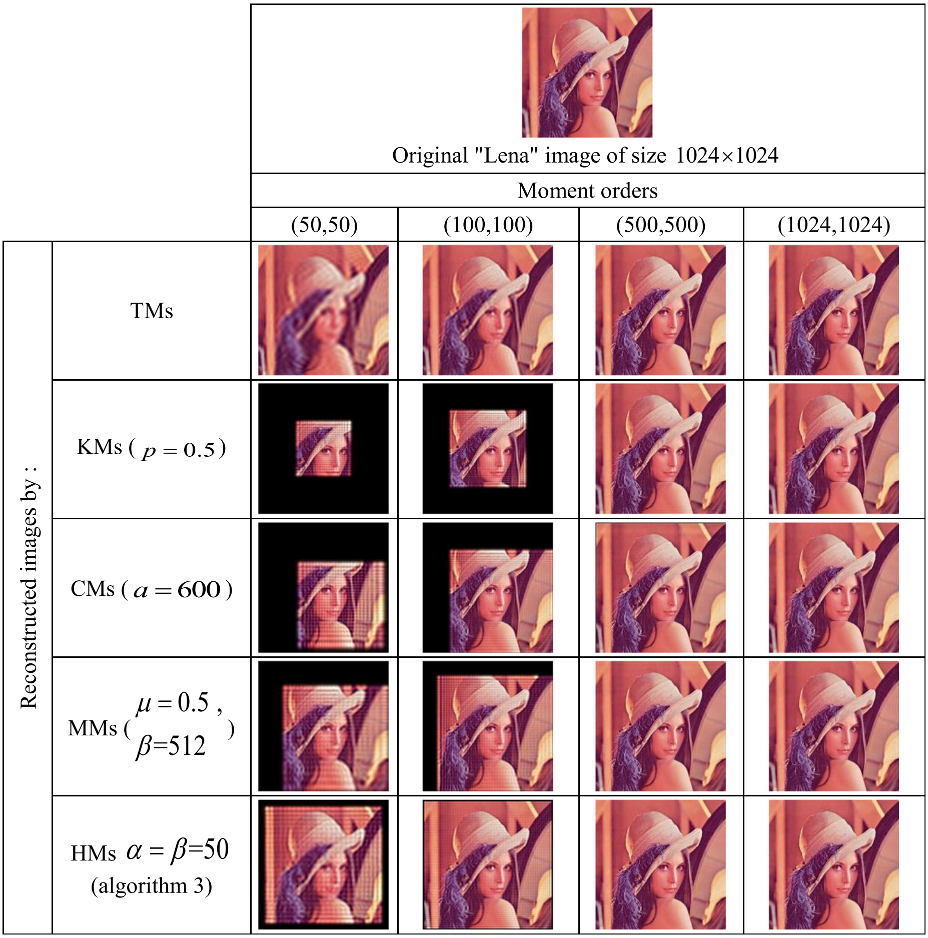

Original "Lena" image of size 1024×1024
Moment orders
(50,50)
(100,100)
(500,500)
(1024,1024)
Reconstructed images by :
TMs
KMs ($p = 0.5$)
CMs ($a = 600$)
MMs ($\mu = 0.5$, β=512)
HMs $\alpha = \beta$=50 (algorithm 3)

Fig. 9 Set of reconstructed "Lena" image by different moment types of order up to (1024, 1024)

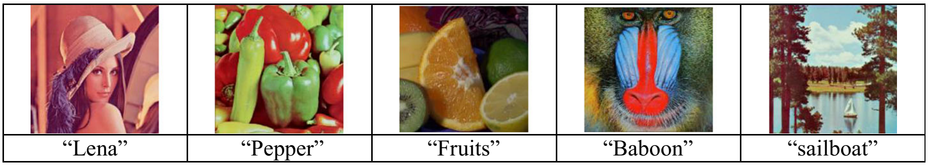


Fig. 10 Standard color images of size 1024 × 1024 used in the test

[12], Charlier (CMs) [12] and HMs (based on algorithm 3). Indeed, Fig. 9 shows the reconstructed "Lena" image by the aforementioned moments, and Fig. 11 shows the average MSE and PSNR values corresponding to the reconstructed test images shown in Fig. 10. The achieved results show on one hand that all the used moment types are stable because they allow the reconstruction of the test images until the last moment order (1024, 1024). On the other hand, it is apparent that the quality of the reconstructed images using high-order KMs is better (high PSNR) compared to other moments type, including HMs. However, KMs are unstable when the moment orders exceed (1075, 1075) [8]. For this reason, only images of size 1024 × 1024 are used in this test. Moreover, HMs are more general than TMs and KMs because the latter are special cases of HMs [22].

In the following test, the robustness of HMs against different types of noise is evaluated. For this purpose, a real medical image named "Boxer-fracture" of size 2048 × 2048 is selected from the database [32]. This image is contaminated by two noise types namely "Salt & pepper" and "Speckle" noise with density of 5% (Fig. 12). Then, these images are reconstructed by HMs, and the reconstruction error (PSNR) is displayed in Fig. 13. The achieved results show a slight decrease in the quality of the noisy reconstructed images compared to the quality of the noise-free reconstructed one, which validates the robustness of HMs in noisy conditions.

5.3 Fast and numerically stable computation of HMIs using large-size images

This section is preserved to demonstrate the effectiveness of the proposed method for fast and stable HMIs computation (algorithm 5).

The computation of HMIs via the proposed method depends on HPs local parameters (α and β). Therefore, the optimal selection of these parameters is crucial when computing HMIs.

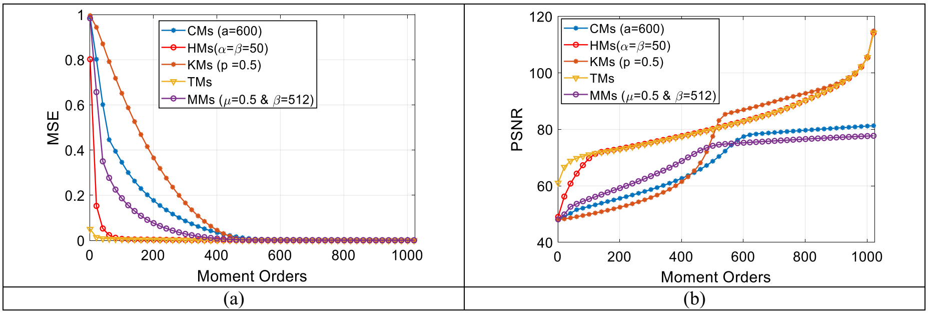


Fig. 11 The average of MSE and PSNR corresponding to the five reconstructed color images by different moment types

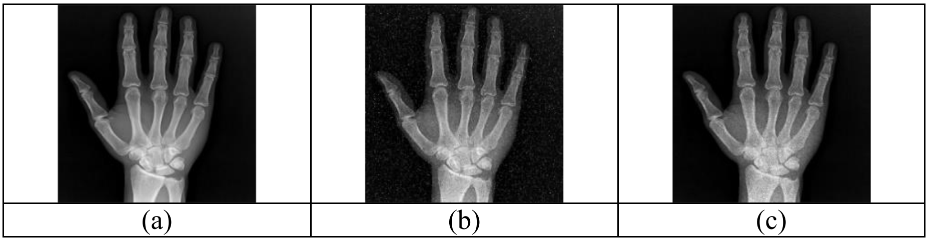


Fig. 12 (a): The original medical image. (b) and (c) the contaminated images by “Salt & pepper” and “Speckle” noise successively (with density of 5%)

In order to carry out this selection, one can use an empirical method, which is similar to the one described in [3]. Indeed, HMIs of a test image are computed for specific values of α and βparameters, then the test image is geometrically transformed by translation, rotation and scaling. Next, the relative error (*RE*) (Eq. (49)) is computed for each parameters case. Finally, the values of α and β that allow to obtain the lower *RE* values are therefore considered as the optimal parameters selection.

The *RE* criterion is defined as:

$$RE = \frac{\|V - V*\|}{\|V\|} \tag{49}$$

where ‖.‖ denotes the Euclidean norm, *V*denotes the invariant moments of the original image, and $V*$ represents the invariant moments of the geometrically transformed image.

To test the effectiveness of the empirical parameters selection method, one use a real X-ray medical image named “Covid 19” of size 2048 × 2048 (Fig. 14) that is selected from [32]. The test image is then translated by a vector varies from (−20, −20) to (+20, +20) with a step of (+4, +4), scaled by a factor scaling ranging from 0.5 to 2 with a step of 0.1, and rotated by a rotation angles ranging from 0° to 360° with a step of 10°. Then, HMIs of the original image and the geometrically transformed one are computed up to the order (5, 5) for four cases of α and βparameters: (A) $\alpha = 15$ and $\beta = 10$, (B) $\alpha = \beta = 10$, (C) and $\alpha = \beta = 5$, and (D) $\alpha = \beta = 0$.

The results of the performed test are presented in Fig. 15. From this figure, it can be noted that the cse $\alpha = \beta = 0$ leads to lower *RE* values, therefore this case will be considered as the

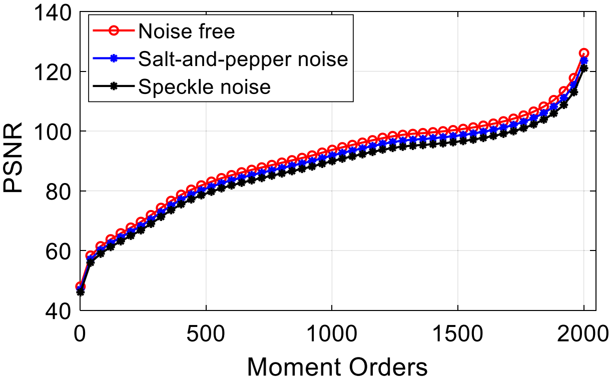
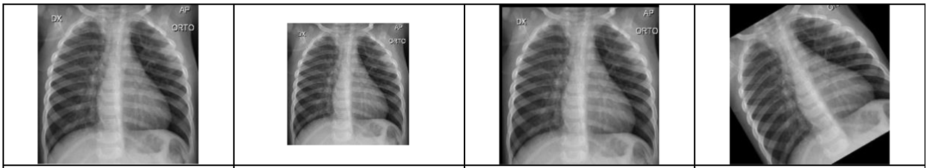


Fig. 13 PSNR corresponding to the reconstructed «Boxer-fracture “image with and without noise



(a)

(b) : scaling by a factor 0.75

(c) : translation by vector (+20, +20)

(d) : Rotation by angle 30°

Fig. 14 (a) "Covid 19" test image, (b), (c) and (d): geometrically transformed test image

optimal parameters selection in the rest of this work. However, the use of the empirical method for the optimal parameters selection is limited due to the few number of the studied cases. It is important to note that the optimal selection of local parameters can be made via population-based algorithms such as the Sine Cosine Algorithm (SCA) [10] or by similar algorithms (PSO, ABC, TLBO…).

In the following test, four grayscale medical images of size 2048 × 2048 are used (Fig. 16). These images that are selected from the database [32], are geometrically transformed, and then the corresponding average *RE* is displayed in Fig. 17. The outcome results indicate on one hand that the proposed computation of HMIs is stable for large-size images. On the other hand, the low *RE* values ($RE \simeq 10^{-10}$) indicate that the computed HMIs are highly accurate and robust against different geometrical transformations.

The following test is conducted to prove the fastness of the proposed HMIs computation method compared to the conventional one that is presented in [36]. Initially, the test image

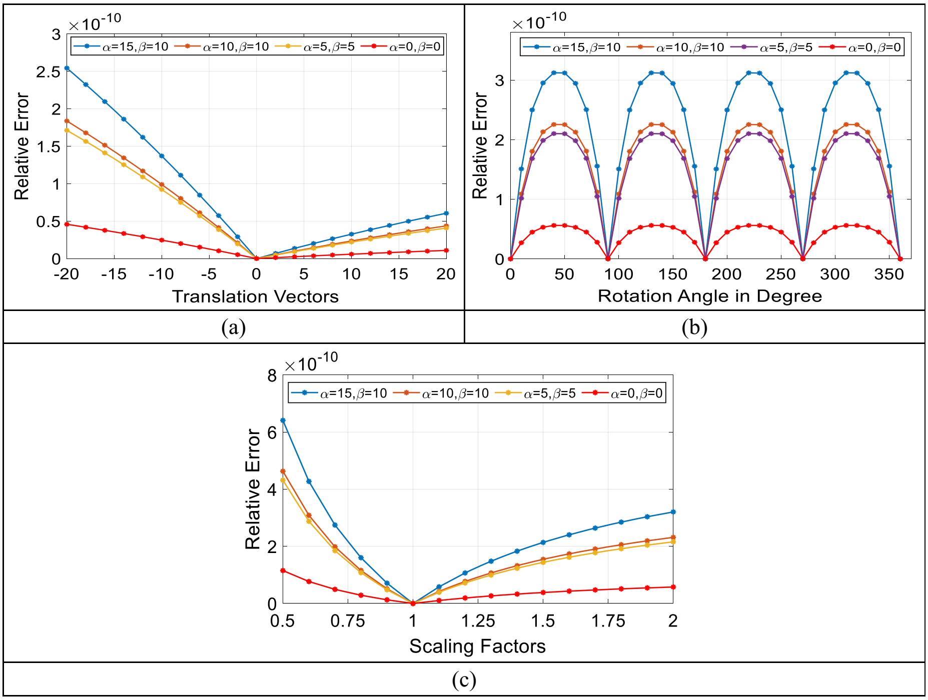


Fig. 15 Relative errors (*RE*) corresponding to "Covid 19" image transformed by (a): translation, (b): rotation, and (c): scaling

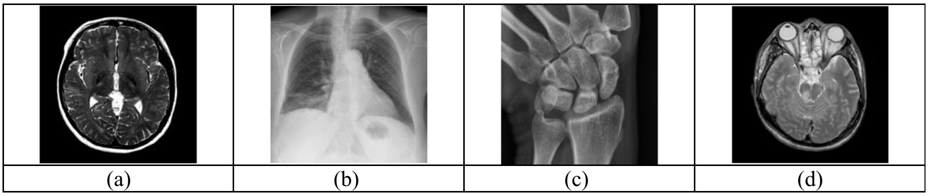


Fig. 16 Set of real medical images of size 2048 × 2048 used in the test

“Covid-19” is resized to 60 × 60, then HMIs are computed up to the order (20, 20) by using both methods. Noting that resizing the test image is undertaken since the conventional computation of HMIs is numerically unstable for images of size larger than 61 × 61 due to gamma and factorial functions that define the terms $A_n^{(\alpha,\beta)}$ and $B_{mk}^{(\alpha,\beta)}$. The time required to compute HMIs by both methods is computed 20 times, and then the average time is presented in Table 4.

The obtained results in Table 4 clearly demonstrate the considerable rapidity of the proposed method compared to the conventional one, especially for higher HMIs order.

Due to the limited paper length, only the advantages of the proposed method for HMIs computation have been illustrated. In future works, the proposed method (algorithm 5) can be successfully employed in various applications related images moments invariants such as pattern recognition, image classification and image zero – watermarking of large-size images.

It is important to point out that some recent methods for computing the invariant moments [3, 17, 20, 37, 39] are numerically unstable for large-size images (>1024 × 1024).For this reason, no comparison with these works is carried out. In addition, accelerating and stabilizing the computation of the previously mentioned methods might be a potentially interesting direction for further research.

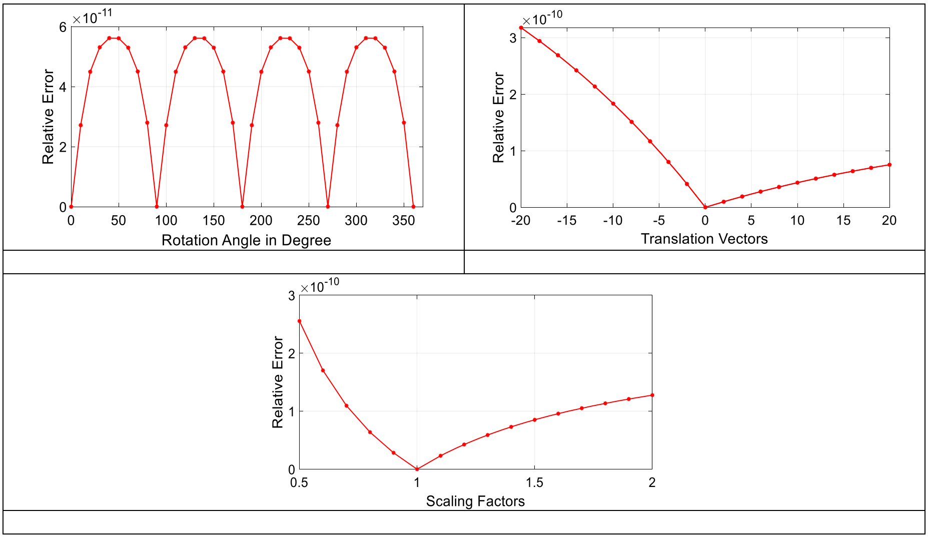


Fig. 17 Average of the REs corresponding to test images geometrically transformed by (a) rotation, (b) translation and (c) scaling

6 Conclusion

In this paper, new algorithms for the computation of high order HMs and HMIs are presented in the field of large-sized signal and image analysis. First, special attention is paid to ensuring the numerical stability and to speed up the computation of HPs with respect to the order n and with respect to the variable x. The proposed algorithms are fully independent from gamma and factorial functions that are the source of numerical instability (Underflow and Overflow). Secondly, a new numerical stability method is proposed to detect and eliminate unstable coefficients during recursive computation of HPs. This method is based on the distribution analysis of HPs values in the polynomial matrix. As a result, the computation of HPs becomes stable and fast for any order n, any value of the variable x and for any value of the local parameters. Also new algorithm is proposed for fast and stable HMIs computation. This algorithm is based on developed recursive relations and matrix formulations. The simulation and comparison result clearly validated the stability and efficiency of the proposed methods for the large-sized signal and image analysis. However, the empirical method used for the selection of HPs parameters is limited due to the few studied cases and therefore needs to be further improved. In perspective and future work, we will focus on other applications of the proposed algorithms such as compression, classification and zero - watermarking of large-size signals and 2D/3D images. In addition, the selection of HPs parameters will be conducted based on metaheuristic algorithms.

References

1. Abdulhussain SH, Al-Haddad SAR, Mahmmod BM, Jassim WA (2017) On computational aspects of Tchebichef polynomials for higher polynomial order. IEEE Access 5:2470–2478
2. Achraf, Daoui Hicham, Karmouni Mhamed, Sayyouri Hassan, Qjidaa 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
3. 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
4. Bocci C, Carlini E, Kileel J (2016) Hadamard products of linear spaces. J Algebra 448:595–617. https://doi.org/10.1016/j.jalgebra.2015.10.008
5. 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
6. Camacho-Bello C, Toxqui-Quitl C, Padilla-Vivanco A, Báez-Rojas JJ (2014) High-precision and fast computation of Jacobi–Fourier moments for image description. JOSA A 31:124–134
7. Comtet L (2012) Advanced combinatorics: the art of finite and infinite expansions. Springer Science & Business Media
8. Daoui A, Yamni M, Elogri O et al (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
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 39:4552–4577. https://doi.org/10.1007/s00034-020-01384-z
10. 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 107854:107854. https://doi.org/10.1016/j.sigpro.2020.107854
11. Daoui A, Yamni M, Karmouni H, et al (2020) Efficient reconstruction and compression of large size ECG signal by Tchebichef moments. In: 2020 international conference on intelligent systems and computer vision (ISCV). Pp 1–6

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 80:1641–1670. https://doi.org/10.1007/s11042-020-09739-z
13. Ernawan F, Kabir N, Zamli KZ (2017) An efficient image compression technique using Tchebichef bit allocation. Optik 148:106–119. https://doi.org/10.1016/j.ijleo.2017.08.007
14. Flusser J, Suk T, Zitova B (2016) 2D and 3D image analysis by moments. John Wiley & Sons
15. Hart JF (1978) Computer approximations. Krieger Publishing Co., Inc.
16. Hmimid A, Sayyouri M, Qjidaa H (2015) Fast computation of separable two-dimensional discrete invariant moments for image classification. Pattern Recogn 48:509–521. https://doi.org/10.1016/j.patcog.2014.08.020
17. 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
18. Hu M-K (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8:179–187
19. Jahid T, Karmouni H, Hmimid A, Sayyouri M, Qjidaa H (2019) Fast computation of Charlier moments and its inverses using Clenshaw's recurrence formula for image analysis. Multimed Tools Appl 78:12183–12201
20. Karakasis EG, Papakostas GA, Koulouriotis DE, Tourassis VD (2013) Generalized dual Hahn moment invariants. Pattern Recogn 46:1998–2014. https://doi.org/10.1016/j.patcog.2013.01.008
21. 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
22. Koekoek R, Lesky PA, Swarttouw RF (2010) Hypergeometric orthogonal polynomials and their q-analogues. Springer Science & Business Media
23. Kuo-Liang Chung Lung-Chun Chang (2003) A new predictive search area approach for fast block motion estimation. IEEE Trans Image Process 12(6) 648–652. https://doi.org/10.1109/TIP.2003.812756
24. Liu X, Han G, Wu J, Shao Z, Coatrieux G, Shu H (2017) Fractional Krawtchouk transform with an application to image watermarking. IEEE Trans Signal Process 65:1894–1908. https://doi.org/10.1109/TSP.2017.2652383
25. Moody GB, Mark RG (2001) The impact of the MIT-BIH arrhythmia database. IEEE Eng Med Biol Mag 20:45–50
26. Morales-Mendoza LJ, Gamboa-Rosales H, Shmaliy YS (2013) A new class of discrete orthogonal polynomials for blind fitting of finite data. Signal Process 93:1785–1793
27. Mukundan R (2004) Some computational aspects of discrete orthonormal moments. IEEE Trans Image Process 13:1055–1059
28. Mukundan R, Ong SH, Lee PA (2001) Image analysis by Tchebichef moments. IEEE Trans Image Process 10:1357–1364
29. Nikiforov AF, Suslov SK, Uvarov VB (1991) Classical orthogonal polynomials of a discrete variable. Springer-Verlag
30. Peleshko D, Peleshko M, Kustra N, Izonin I (2011) Analysis of invariant moments in tasks image processing. In: 2011 11th international conference the experience of designing and application of CAD Systems in Microelectronics (CADSM). Pp 263–264
31. Photojournal: NASA's Image Access Home Page. https://photojournal.jpl.nasa.gov/. Accessed 28 Dec 2020
32. Radiopaedia.org, the wiki-based collaborative Radiology resource. https://radiopaedia.org/. Accessed 31 Mar 2020
33. Rahman SMM, Howlader T, Hatzinakos D (2016) On the selection of 2D Krawtchouk moments for face recognition. Pattern Recogn 54:83–93. https://doi.org/10.1016/j.patcog.2016.01.003
34. Raj PA (2020) Charlier and Meixner moments and their application for texture and image De-noising problems. In: Information, Photonics and Communication. Springer, pp. 123–134
35. Rivero-Castillo D, Pijeira H, Assunçao P (2015) Edge detection based on Krawtchouk polynomials. J Comput Appl Math 284:244–250. https://doi.org/10.1016/j.cam.2014.11.017
36. Sayyouri M, Hmimid A, Qjidaa H (2013) Improving the performance of image classification by Hahn moment invariants. JOSA A 30:2381–2394. https://doi.org/10.1364/JOSAA.30.002381
37. Sayyouri M, Hmimid A, Qjidaa H (2015) A fast computation of novel set of Meixner invariant moments for image analysis. Circuits Syst Signal Process 34:875–900. https://doi.org/10.1007/s00034-014-9881-7
38. Singh C, Walia E, Pooja UR (2012) Analysis of algorithms for fast computation of pseudo Zernike moments and their numerical stability. Digit Signal Process 22:1031–1043. https://doi.org/10.1016/j.dsp.2012.06.009
39. Sit A, Kihara D (2014) Comparison of image patches using local moment invariants. IEEE Trans Image Process 23:2369–2379. https://doi.org/10.1109/TIP.2014.2315923
40. Teague MR (1980) Image analysis via the general theory of moments. JOSA 70:920–930

41. Tsougenis ED, Papakostas GA, Koulouriotis DE (2015) Image watermarking via separable moments. Multimed Tools Appl 74:3985–4012
42. Xiao B, Lu G, Zhang Y, Li W, Wang G (2016) Lossless image compression based on integer discrete Tchebichef transform. Neurocomputing 214:587–593. https://doi.org/10.1016/j.neucom.2016.06.050
43. Xiao B, Luo J, Bi X, Li W, Chen B (2020) Fractional discrete Tchebyshev moments and their applications in image encryption and watermarking. Inf Sci 516:545–559. https://doi.org/10.1016/j.ins.2019.12.044
44. Yamni M, Daoui A, El ogri O et al (2020) Fractional Charlier moments for image reconstruction and image watermarking. Signal Process 171:107509. https://doi.org/10.1016/j.sigpro.2020.107509
45. Yap P-T, Paramesran R, Ong S-H (2003) Image analysis by Krawtchouk moments. IEEE Trans Image Process 12:1367–1377. https://doi.org/10.1109/TIP.2003.818019
46. Yap P-T, Paramesran R, Ong S-H (2007) Image analysis using Hahn moments. IEEE Trans Pattern Anal Mach Intell 29:2057–2062
47. Zhang G, Luo Z, Fu B, Li B, Liao J, Fan X, Xi Z (2010) A symmetry and bi-recursive algorithm of accurately computing Krawtchouk moments. Pattern Recogn Lett 31:548–554
48. 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
49. Zhu H, Shu H, Liang J, Luo L, Coatrieux JL (2007) Image analysis by discrete orthogonal Racah moments. Signal Process 87:687–708. https://doi.org/10.1016/j.sigpro.2006.07.007
50. Zhu H, Liu M, Shu H, Zhang H, Luo L (2010) General form for obtaining discrete orthogonal moments. IET Image Process 4:335–352. https://doi.org/10.1049/iet-ipr.2009.0195

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

Affiliations

Achraf Daoui[1] • **Hicham Karmouni**[2] • **Mhamed Sayyouri**[1] • **Hassan Qjidaa**[2]

[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