electronics

MDPI

Article

Four-Term Recurrence for Fast Krawtchouk Moments Using Clenshaw Algorithm

Barmak Honarvar Shakibaei Asli [1,*] and Maryam Horri Rezaei [2]

[1] Centre for Life-Cycle Engineering and Management, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire MK43 0AL, UK
[2] Independent Researcher, Milton Keynes MK10 9AA, UK
* Correspondence: barmak@cranfield.ac.uk

Abstract: Krawtchouk polynomials (KPs) are discrete orthogonal polynomials associated with the Gauss hypergeometric functions. These polynomials and their generated moments in 1D or 2D formats play an important role in information and coding theories, signal and image processing tools, image watermarking, and pattern recognition. In this paper, we introduce a new four-term recurrence relation to compute KPs compared to their ordinary recursions (three-term) and analyse the proposed algorithm speed. Moreover, we use Clenshaw's technique to accelerate the computation procedure of the Krawtchouk moments (KMs) using a fast digital filter structure to generate a lattice network for KPs calculation. The proposed method confirms the stability of KPs computation for higher orders and their signal reconstruction capabilities as well. The results show that the KMs calculation using the proposed combined method based on a four-term recursion and Clenshaw's technique is reliable and fast compared to the existing recursions and fast KMs algorithms.

Keywords: Krawtchouk moments; four-term recurrence; digital filter; Z-transform; Clenshaw formula

Citation: Honarvar Shakibaei Asli, B.; Rezaei, M.H. Four-Term Recurrence for Fast Krawtchouk Moments Using Clenshaw Algorithm. *Electronics* **2023**, *12*, 1834. https://doi.org/10.3390/electronics12081834

Academic Editor: Catalin Stoean

Received: 7 March 2023
Revised: 27 March 2023
Accepted: 10 April 2023
Published: 12 April 2023

1. Introduction

Image moments are projections of an image function onto a polynomial basis or kernel functions which can be orthogonal or non-orthogonal. Geometric moments (GMs) and complex moments (CMs) are examples of non-orthogonal moments while orthogonal moments are based on an orthogonal kernel basis such as Legendre, Hermite, Tchebichef, and Krawtchouck polynomials. Apart from these orthogonality properties, we can divide moment kernels into continuous and discrete forms. This segmentation could be based on the polynomials' nature: analogue or digital. For example, Legendre polynomials are defined as a continuous system over the interval $[-1, 1]$. On the other hand, Tchebichef polynomials are examples of a discrete system over the interval $[0, N-1]$ by having a degree at most N forms a natural reference for the discrete orthogonal polynomials [1].

Moments are applied as feature descriptors in various areas such as image processing and computer vision [2]. These feature extractions among moment transformations could be applied for a wide range of applications such as object recognition [3–5], edge detection [6,7], coding [8], classification [9,10], and image reconstruction [11–13]. Moment kernels depend on the polynomial basis and might be represented in a continuous or discrete model. Legendre [14], Zernike [15,16], Gaussian–Hermite [17], pseudo-Zernike [18,19], and Fourier–Mellin [20,21] moments are some examples of continuous orthogonal moments. These continuous moments are not suitable for digital image analysis due to the digital nature of image pixel values and they need to be discretised along the image dimensions. To overcome this problem raised by continuous moments, many researchers have begun to apply discrete polynomials as the basis of discrete moments which are fitted for digital image studies. Examples of discrete orthogonal moments include Tchebichef [22,23], Krawtchouk [24,25], Charlier [26,27], Meixner [28,29], Hahn/Dual-Hahn [30,31], and Racah [32].

Since most orthogonal polynomials involve hypergeometric functions, the computation of these polynomials and their corresponding moments are time-consuming. Several fast computational methods and techniques are proposed to accelerate this process, which is usually focusing on the calculation of: (1) the moment kernels (orthogonal polynomials) and (2) the summation of this kernel over the 1D, 2D, or 3D signals. Regarding the first approach, using a recursive relationship to compute the polynomial kernel is suggested instead of the direct computation of the polynomials by using the hypergeometric functions. Moreover, for some polynomials, their symmetry property would make this computation easy, fast, and effective [12,33–35]. Concerning the computation of summation over the multidimensional signal, the reduction of the number of signal data could decrease the moment's computation time where the kernel functions are evaluated. One approach is to consider the signal as a series of intensity slices and blocks to calculate the moment functions from the represented blocks instead of the whole signal data [36,37]. Aside from these two approaches, there is a combined technique to compute the moment functions in terms of recursive calculation of the orthogonal polynomials and their summations over the signal data values (i.e., image pixels) using Clenshaw's algorithm.

The Clenshaw recurrence formula evaluates a sum of multiplied indexed coefficients by functions which obey a recurrence relation. This algorithm is extremely useful for the fast computation of polynomials [38], the fast sparse spectral method for nonlinear integro-differential equations with general kernels [39], the fast technique for calculating geoid undulation [40], and evaluation of Chebyshev polynomials on intervals to find its roots [41].

Some considerable studies are concentrated on this combined method to compute the forward and inverse moment transforms using Clenshaw's recursive formula [29,42–44]. Among these studies, many pieces of research have been focused on the computation of KMs and their corresponding KPs using fast methods and effective algorithms (see [35,45,46]). Most of these research works apply a three-term recurrence relation to compute KPs [45,47]. High dependency of the coefficients for each term of the three-term recurrence relation on variables x, n, p, and N became a matter of concern in computing KPs in both recursions with respect to x or n (where x is the KPs discrete independent variable, n is the polynomial degree, p is the localisation parameter, and N is the number of discrete points). Using a combined recursive relation to generate a four-term recurrence equation in both n and x directions is one of this research targets to reduce the dependencies of the KPs coefficients for each term of recursion. However, due to the numerical instability of the hypergeometric function used in KPs, the KMs calculation becomes erratic when the order of KMs increases. The proposed recursion in the n direction by Yap et al. [24] fails when the polynomial-size (N) increases. Therefore, the propagation error will be developed. To overcome this fluctuation, Jassim W.A. et al. [48] proposed another three-term recurrence in the x direction by segmenting the KP plane into two segments. Their results show that the implemented algorithm outperforms the Yap recursion.

In this study, we present a new algorithm to compute KMs using a four-term recurrence relation which is combined with Clenshaw's formula to accelerate the computation speed. Unlike the existing method in [47] that applied another four-term recurrence to compute KPs in only a limited range of x and n, we implement a unique four-term recursion to compute KPs in a full range of KPs degree and variable. Moreover, the complexity of the proposed four-term recurrence relation is less than the proposed algorithm in [47]. The proposed technique is also robust in terms of computation of the KMs for all values of the parameter $p \in (0, 1)$.

This paper is organised as follows. Section 2 presents the mathematical definition of KPs and their corresponding moments as well as the recurrence relations with respect to n and x directions. The new four-term recursion to generate the KPs is introduced in Section 3. Section 4 applies Clenshaw's formula to compute KMs using the $\mathscr{Z}$-transform properties. The validation of the theoretical framework presented in the previous sections

is given in Section 5 through experimental study. Concluding remarks are discussed in Section 6.

2. Krawtchouk Polynomials (KPs) and Moments (KMs): Mathematical Framework

Krawtchouk Polynomials (KPs) were introduced by a Soviet–Ukrainian mathematician, Mikhail P. Kravchuk in 1929 [49] and later, in 1934, Jozef Meixner generalised these polynomials into a form of Meixner polynomials of the first kind [50]. KPs were used in the field of image processing by Yap et al. [24]. In this section, we recall the KPs with their normalised and weighted forms as well as their recurrence relations in both n and x directions. We then define the 1D and 2D KMs for signal and image analysis.

2.1. Krawtchouk Polynomials

The nth degree discrete classical KP is defined as

$$K_n(x;p,N-1) = {}_2F_1\left(-n,-x;-N+1;\frac{1}{p}\right) \tag{1}$$

where $x, n = 0,1,2,...,N-1, N > 0, p \in (0,1)$ is the localisation parameter and ${}_2F_1(a,b,c;z)$ is the Gauss hypergeometric series and is given by:

$${}_2F_1(a,b,c;z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \tag{2}$$

and $(a)_k$ is the Pochhammer symbol given by

$$(a)_k = a(a+1)\cdots(a+k-1), \quad k \geq 1 \quad \text{and} \quad (a)_0 = 1. \tag{3}$$

For the sake of notation simplicity, we consider ${}_2F_1(a,b,c;z)$ as $F(a,b,c;z)$ in the rest of this paper. The discrete orthogonal KPs satisfy the following orthogonality property:

$$\sum_{x=0}^{N-1} w(x;p,N-1) K_n(x;p,N-1) K_m(x;p,N-1) = \rho(n;p,N-1)\delta_{nm} \tag{4}$$

where δ_{nm} is the Kronecker function, $w(x;p,N-1)$ is the weight function and $\rho(n;p,N-1)$ is the squared norm, respectively defined as

$$w(x;p,N-1) = \binom{N-1}{x} p^x (1-p)^{N-1-x}, \tag{5a}$$

$$\rho(n;p,N-1) = \left(\frac{p-1}{p}\right)^n \frac{n!}{(1-N)_n}. \tag{5b}$$

Based on the above definitions, the weighted and normalised KPs, $\overline{K}_n$, can be represented as

$$\overline{K}_n(x;p,N-1) = K_n(x;p,N-1)\sqrt{\frac{w(x_i p,N-1)}{\rho(n;p,N-1)}} \tag{6}$$

and the KPs orthogonality condition described in (4) might be reduced to the following form now:

$$\sum_{x=0}^{N-1} \overline{K}_n(x;p,N-1)\overline{K}_m(x;p,N-1) = \delta_{nm}. \tag{7}$$

There is also the following symmetry relation for KPs as [51]

$$\overline{K}_n(x;p,N-1) = \overline{K}_x(n;p,N-1). \tag{8}$$

This means that it is always possible to swap the n and x places in future analysis. Figure 1 illustrates the KPs graphs with respect to the localisation parameter, p, for $N = 10$ and $x = 0, 1, 2$ and 3. This figure shows by increasing the value of x, the amplitude of the KPs increases as well. For example, the maximum range of the KPs for $x = 0$ is 1 (see Figure 1a), while this range grows to 16 when $x = 3$ (see Figure 1d). On the other hand, by increasing the number of the KPs orders (n) from 0 to 9, the pick point of the KPs moves from left to right by the localisation parameter's increment from 0 to 1. A 3D representation of the KPs is shown in Figure 2 with $N = 256$ and various values of parameter p.

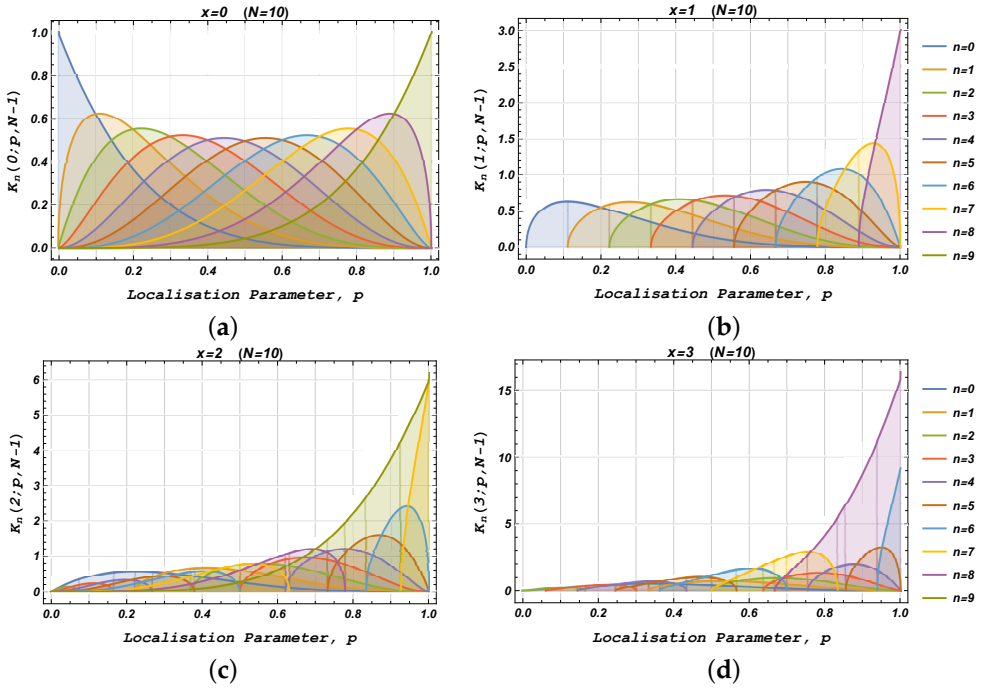


Figure 1. The KPs graphs with respect to the localisation parameter, p, for $N = 10$ and different values of x: (**a**) $x = 0$, (**b**) $x = 1$, (**c**) $x = 2$, (**d**) $x = 3$.

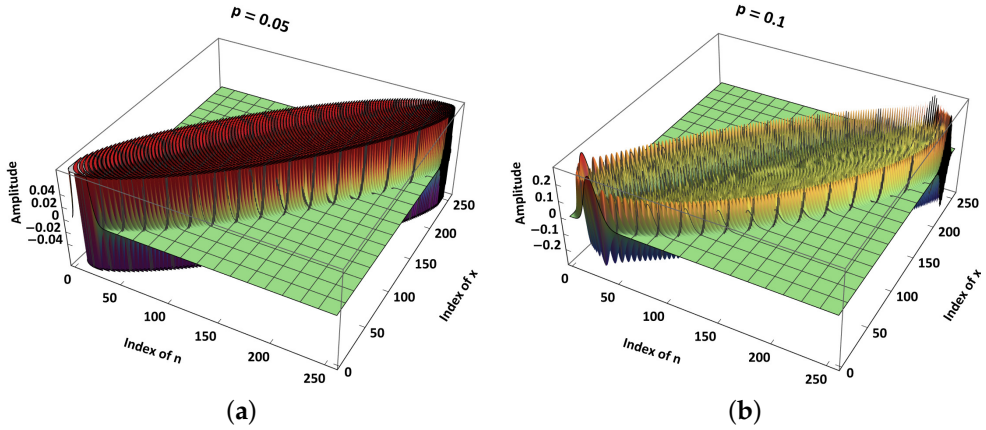


Figure 2. *Cont.*

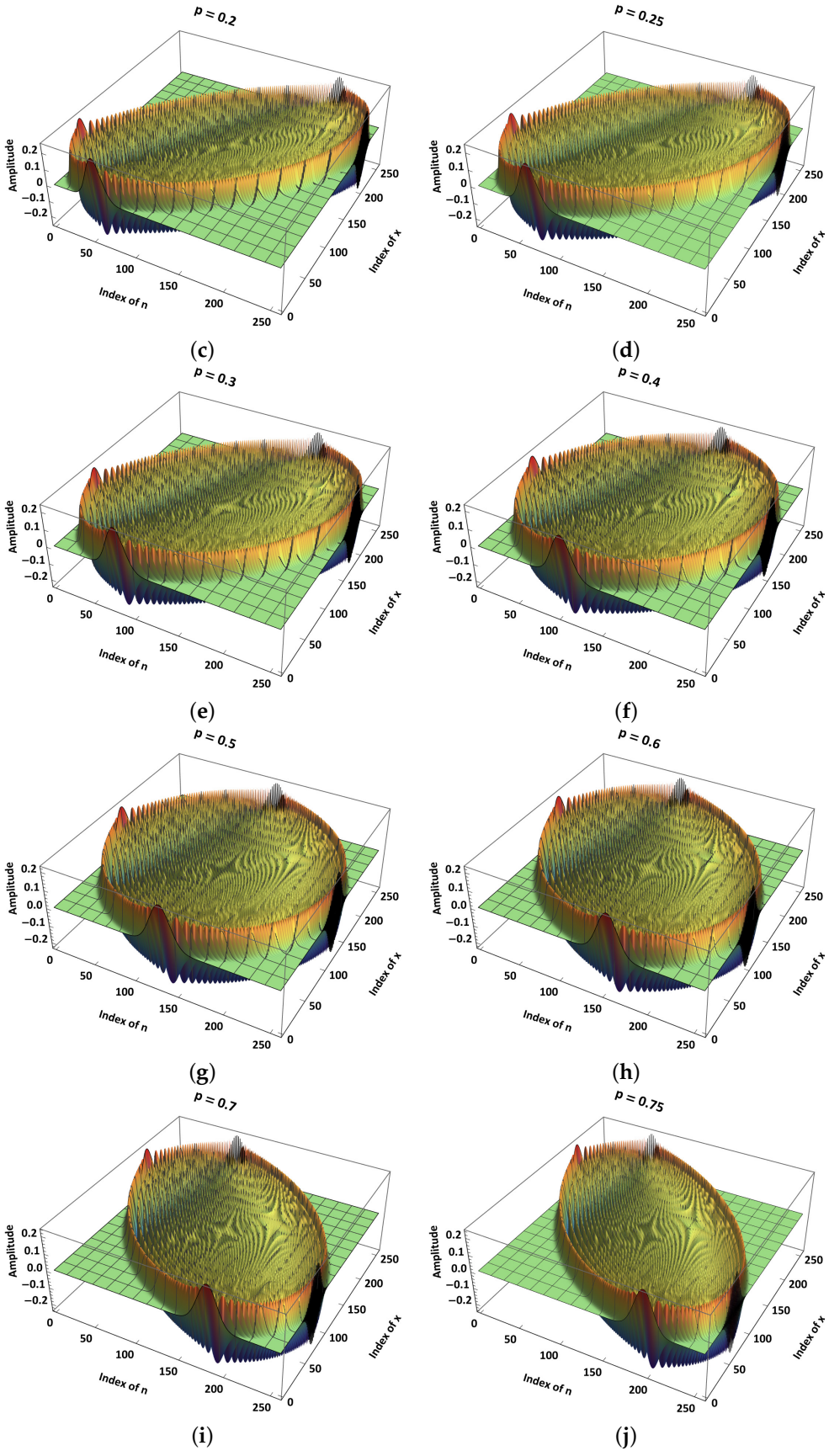
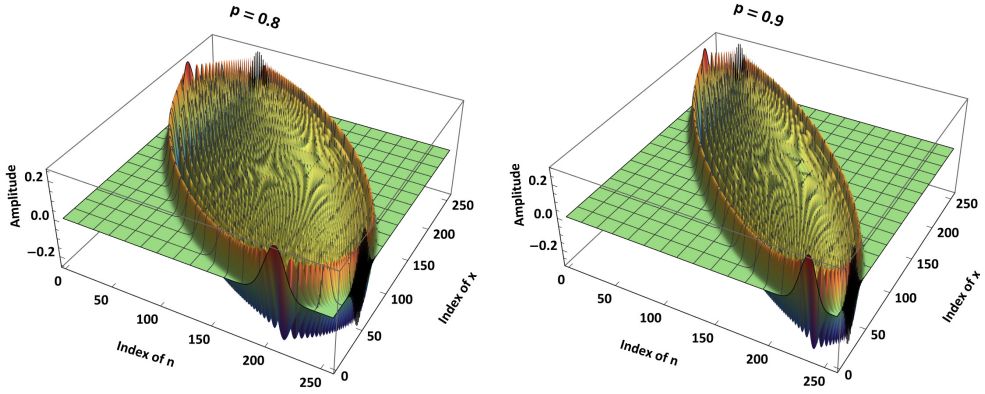


Figure 2. *Cont.*



(**k**) (**l**)

Figure 2. 3D plot of the KPs computed for N = 256 and various values of p: (**a**) $p = 0.05$, (**b**) $p = 0.1$, (**c**) $p = 0.2$, (**d**) $p = 0.25$, (**e**) $p = 0.3$, (**f**) $p = 0.4$, (**g**) $p = 0.5$, (**h**) $p = 0.6$, (**i**) $p = 0.7$, (**j**) $p = 0.75$, (**k**) $p = 0.8$, (**l**) $p = 0.9$.

2.2. Recurrence Relations of the KPs

There are several types of recurrence relations for generating KPs in the literature. Due to the hypergeometric function term in KPs, they depend on three discrete variables: n, x, and N. Since the localisation parameter, p, is a real number between zero and one, it is possible to represent KPs in a recursion form in terms of these discrete variables only [52]. In this paper, we do not focus on the recursion formula in terms of N and recall two regular three-term recurrence relations of the KPs with respect to variables n and x. We then propose a new four-term recursion of the KPs with respect to both variables n and x.

2.2.1. Recursive Relation with Respect to the Order n

The discrete orthogonal KPs satisfy the recurrence relation with respect to order n according to the following relation [46]:

$$\overline{K}_n(x) = A_n\Big[p(N-2n+1)+n-x-1\Big]\overline{K}_{n-1}(x) - B_n\overline{K}_{n-2}(x), \tag{9}$$

where

$$\left\{\begin{array}{l} A_n = \dfrac{1}{\sqrt{p(1-p)n(N-n)}}, \\ B_n = \sqrt{\dfrac{(n-1)(N-n+1)}{n(N-n)}}, \\ \overline{K}_0(x) = \sqrt{w(x;p,N-1)}, \\ \overline{K}_1(x) = \sqrt{w(x;p,N-1)}\,\dfrac{(N-1)p-x}{\sqrt{p(1-p)(N-1)}}. \end{array}\right\} \text{ Initial conditions} \tag{10}$$

As can be seen from Equation (10), the last two KPs of the zeroth and first orders are the initial conditions for calculating the KPs using three-term recursion in Equation (9).

2.2.2. Recursive Relation with Respect to the Independent Variable x

Many researchers applied another three-term recurrence relation for the computation of KPs with respect to the independent discrete variable of x [44,45,47,53,54]. Their implementation of this recursion with respect to x is not unique in most of this literature. Unlike the mentioned articles, here we use the symmetry property of the KPs represented in (8) to express the three-term recursive relation of the KPs with respect to x which is following the same three-term recursion with respect to n by swapping the variables x and n in both the

main formula and initial conditions (as mentioned earlier). Equations (11) and (12) define this recursion formula.

$$\overline{K}_n(x) = C_x\Big[p(N-2x+1)+x-n-1\Big]\overline{K}_n(x-1) - D_x\overline{K}_n(x-2), \tag{11}$$

where

$$\left\{\begin{array}{l} C_x = \dfrac{1}{\sqrt{p(1-p)x(N-x)}}, \\ D_x = \sqrt{\dfrac{(x-1)(N-x+1)}{x(N-x)}}, \\ \overline{K}_n(0) = \sqrt{\dfrac{(1-p)^{N-1}}{\rho(n;p,N-1)}}, \\ \overline{K}_n(1) = \sqrt{\dfrac{(1-p)^{N-1}}{\rho(n;p,N-1)}}\dfrac{(N-1)p-n}{\sqrt{p(1-p)(N-1)}}. \end{array}\right\} \text{ Initial conditions} \tag{12}$$

As can be seen from the above equations, even the coefficients A_n and B_n are symmetrical to the coefficients C_x and D_x.

2.3. Krawtchouk Moments (KMs)

Given a set of weighted and normalised KPs, $\overline{K}_n(x)$, with weight function $w(x)$ and norm $\rho(n)$, the orthogonal KM of order $n < N$, Q_n, of any bounded 1D signal, $f(x), x = 0, 1, ..., N-1$, is defined as [24,25]

$$Q_n = \sum_{x=0}^{N-1} f(x)\overline{K}_n(x). \tag{13}$$

If only the moments of order up to N_{max} are considered, then the recovered signal can be approximated by

$$\widetilde{f}(x) = \sum_{n=0}^{N_{max}} Q_n\overline{K}_n(x). \tag{14}$$

While the $(n+m)$th order Krawtchouk moments, Q_{nm}, for a 2D image $f(x,y)$ with a support of $N \times M$ pixels can be defined as:

$$Q_{nm} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\overline{K}_n(x)\overline{K}_m(y). \tag{15}$$

If only the moments of order up to (N_{max}, M_{max}) are computed, then the reconstructed image can be approximated by

$$\widetilde{f}(x,y) = \sum_{n=0}^{N_{max}}\sum_{m=0}^{M_{max}} Q_{nm}\overline{K}_n(x)\overline{K}_m(y). \tag{16}$$

The moments as scalar quantities could be considered capable and superior data descriptors. The signal representation by using moment features without redundancy is the most important role of these functions. The KMs of the lower orders contain the most energy of the signal, whereas the KMs of the higher orders illustrate the high-frequency components of the signal.

3. Main Result

Most of the orthogonal polynomials could be computed by three-term recurrence relations in terms of the polynomials' degree. In the case of the discrete orthogonal polynomials, there is another three-term recursion with respect to the independent variable x, as

we showed both recursions for the KPs earlier. Since many of these orthogonal polynomials can be described by hypergeometric functions, we can find some higher-order term recurrence relations to compute them [55]. Four-term recurrence relations are easy to compute due to their low dependencies on the polynomial order or independent variable. Therefore, they have less complexity than three-term recurrence relations [16,56]. Here, we propose a new four-term recurrence relation to generating the KPs with respect to both order (n) and independent variable (x). The most significant advantage of this four-term recursion is its order. The existing recursions described in Equations (9) and (11) are second-order recursions while the proposed method is in a first-order recursion form.

We start this section with a theorem about the hypergeometric function $F(a, b, c; z)$. We then prove the main result of generating the KPs with the help of this theorem and its following lemma.

Theorem 1. *The Gauss hypergeometric series satisfy the four-term recurrence relation formula:*

$$\begin{aligned} aF(a+1,b,c;z) + a(z-1)F(a+1,b+1,c;z) + (a-c+1)F(a,b+1,c;z) \\ = (a-c+1)F(a,b,c;z). \end{aligned} \tag{17}$$

Proof of Theorem 1. To prove the theorem, we need to consider some identities of the Pochhammer symbol as follows:

$$\begin{cases} \dfrac{(a+1)_k}{(a)_k} = \dfrac{a+k}{a} \\ \dfrac{(a-1)_k}{(a)_k} = \dfrac{a-1}{a+k-1} \\ \dfrac{(a)_{k+1}}{(a)_k} = a+k \\ \dfrac{(a-1)_k}{(a)_{k-1}} = a-1. \end{cases} \tag{18}$$

We then start with Equation (17) to rewrite the three hypergeometric series in the right-hand side (RHS) of this equation in the form of a summation of the Pochhammer symbols, power function z^k, and the factorial term in the denominator. Due to the similarity of both sides of (17), we are able to remove the summations and the power functions to simplify the procedure of this proof. Using the hypergeometric function definition in (2) and ignoring the summation, factorial terms, and exponential term, the RHS of (17) can be written as:

$$\begin{aligned} & a\sum_{k=0}^{\infty}\frac{(a+1)_k(b)_k}{(c)_k}\frac{z^k}{k!} + a\sum_{k=0}^{\infty}\frac{(a+1)_k(b+1)_k}{(c)_k}\frac{z^{k+1}}{k!} \\ & - a\sum_{k=0}^{\infty}\frac{(a+1)_k(b+1)_k}{(c)_k}\frac{z^k}{k!} + (a-c+1)\sum_{k=0}^{\infty}\frac{(a)_k(b+1)_k}{(c)_k}\frac{z^k}{k!} \end{aligned} \tag{19}$$

The above expression has four terms. By using Appendix A, we can simplify the second term that contains exponential terms of z^{k+1} to its normal form z^k to achieve the hypergeometric function's definition.

$$
\begin{aligned}
&a\sum_{k=0}^{\infty}\frac{(a)_k(b)_k(a+k)}{a(c)_k}\frac{z^k}{k!}+a\sum_{k=0}^{\infty}\frac{(a)_k(b)_k}{ab}\frac{(c+k-1)k}{(c)_k}\frac{z^k}{k!} \\
&-a\sum_{k=0}^{\infty}\frac{(a)_k(b)_k(a+k)(b+k)}{ab(c)_k}\frac{z^k}{k!}+(a-c+1)\sum_{k=0}^{\infty}\frac{(a)_k(b)_k(b+k)}{b(c)_k}\frac{z^k}{k!} \\
&=\sum_{k=0}^{\infty}\frac{(a)_k(b)_k}{(c)_k}\frac{z^k}{k!}\left\{a+k+\frac{(c+k-1)k}{b}-\frac{(a+k)(b+k)}{b}+\frac{(a-c+1)(b+k)}{b}\right\} \\
&=\sum_{k=0}^{\infty}\frac{(a)_k(b)_k}{(c)_k}\frac{z^k}{k!}(a-c+1)=(a-c+1)F(a,b,c,;z).
\end{aligned}
\tag{20}
$$

The proof of Theorem 1 is now completed.
□

Lemma 2. *The KPs satisfy the first-order four-term recurrence relation formula as follows:*

$$K_n(x)=\frac{n}{n-N}\Big[K_{n-1}(x)+\lambda K_{n-1}(x-1)\Big]+K_n(x-1), \tag{21}$$

where $\lambda=\frac{1}{p}-1$.

Proof of Lemma 2. The proof is straightforward and can be reached by substituting $a=-n$, $b=-x$, $c=1-N$ and $z=\frac{1}{p}$ in the derived recurrence relation of Theorem 1. □

Note that the initial values for computation of the KPs using the proposed four-term recursion are: $K_n(0)=1$ and $K_0(x)=1$. Figure 3 shows an algorithm to compute Krawtchouk polynomials, $K_n(x)$, using its previous three-term values. This lattice network uses a joint block containing a simple digital filter which involves two adders and two multipliers. This joint block has three inputs as $K_n(x-1)$, $K_{n-1}(x-1)$, and $K_{n-1}(x)$ with a single output as $K_n(x)$.

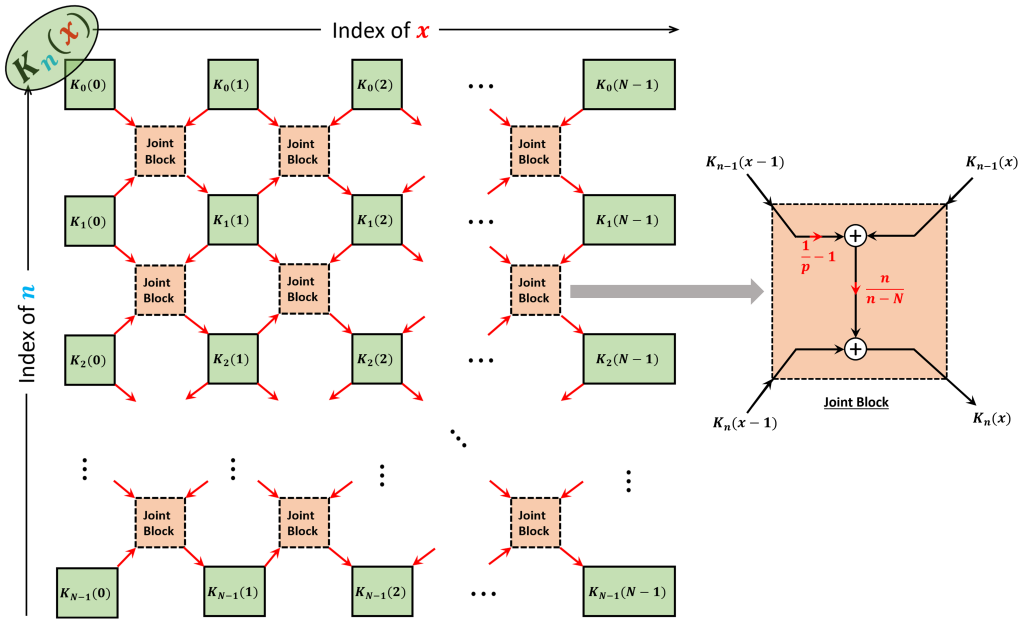


Figure 3. Lattice network for calculating Krawtchouck polynomials using the proposed four-term recurrence.

We would point out that Equation (21) can be used in its symmetric version using (8) as

$$K_n(x)=\frac{x}{x-N}\Big[K_n(x-1)+\lambda K_{n-1}(x-1)\Big]+K_{n-1}(x). \tag{22}$$

Lemma 3. *The weighted Krawtchouk polynomials satisfy the first-order four-term recurrence relation formula as:*

$$\overline{K}_n(x) = \alpha(x)\overline{K}_n(x-1) - \beta(n)\overline{K}_{n-1}(x) - \gamma(n,x)\overline{K}_{n-1}(x-1), \tag{23}$$

where $\lambda = \frac{1}{p} - 1$, $\alpha(x) = \sqrt{\frac{N-x}{\lambda x}}$, $\beta(n) = \sqrt{\frac{n}{\lambda(N-n)}}$ *and* $\gamma(n,x) = \sqrt{\frac{n(N-x)}{x(N-n)}}$. *The initial conditions for the weighted Krawtchouk polynomials are:* $\overline{K}_0(x) = \sqrt{w(x)}$ *and* $\overline{K}_n(0) = \sqrt{\frac{(1-p)^{N-1}}{\rho(n)}}$.

Proof of Lemma 3. The proof is straightforward based on (22) by applying the weight and norm functions on that. □

4. Clenshaw Algorithm for Computing KMs in $\mathscr{Z}$-Domain

By taking the $\mathscr{Z}$-transform of Equation (22) with respect to the order of the Krawtchouk polynomials (n), we have:

$$K(z,x) = \frac{x}{x-N}\Big[K(z,x-1) + \lambda z^{-1}K(z,x-1)\Big] + z^{-1}K(z,x), \tag{24}$$

where $K(z,x) = \mathscr{Z}\{K_n(x)\}$. Equation (24) can be simplified as follows:

$$K(z,x+1) = b(z,x)K(z,x), \tag{25}$$

where $b(z,x) = \left(\frac{x+1}{x+1-N}\right)\frac{z+\lambda}{z-1}$.

Now, to compute the 1D KMs, we start with Equation (13), which can be rewritten as follows (we use the short form of Equation (5) as $w(x)$ and $\rho(n)$ instead of repeating the variables p and $N-1$):

$$Q_n = \sum_{x=0}^{N-1} f(x)\sqrt{\frac{w(x)}{\rho(n)}}K_n(x) = \frac{P_n}{\sqrt{\rho(n)}}, \tag{26}$$

where:

$$g(x) = f(x)\sqrt{w(x)}, \tag{27a}$$

$$P_n = \sum_{x=0}^{N-1} g(x)K_n(x). \tag{27b}$$

To connect the Equation (27b) to the derived $\mathscr{Z}$-transform of the KPs in (24), we take the $\mathscr{Z}$-transform of Equation (27b) with respect to the order of the Krawtchouk polynomials (n):

$$P(z) = \sum_{x=0}^{N-1} g(x)K(z,x), \quad \text{where: } K(z,x+1) = b(z,x)K(z,x) \text{ and } P(z) = \mathscr{Z}\{P_n\}. \tag{28}$$

As the proposed method, we apply Clenshaw's recurrence algorithm to the above equation. To perform the summation in (28) for given sequences of $g(0), g(1), ..., g(N-1)$, we compute the values $Y_X(z)$ by the "reverse" recurrence formula:

$$Y_N(z) = Y_{N+1}(z) = 0 \tag{29a}$$

$$Y_x(z) = g(x) + b(z,x)Y_{x+1}(z). \tag{29b}$$

Figure 4 shows the digital filter structure for generating $Y_x(z)$ using the recurrence equation described in (29). Note that in this figure, to describe $Y_x(z)$ in terms of $Y_{x-1}(z)$ and the input $g(x)$, we need to exchange x to $(x-1)$ in Equation (29b).

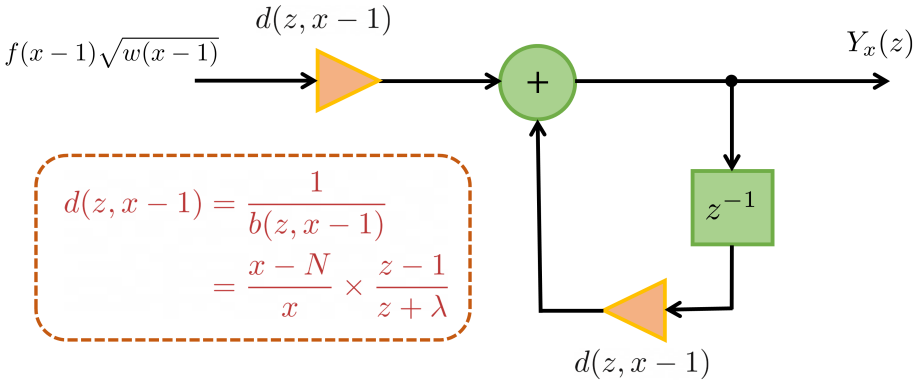


Figure 4. 1D digital filter structure to generate $Y_x(z)$ based on Equation (29).

By expanding Equation (28) and using Equation (29), it is possible to express $P(z)$ as follows (this procedure is part of Clenshaw's algorithm):

$$P(z) = K(z,0)g(0) + K(z,1)Y_1(z). \tag{30}$$

The advantage of the proposed method in the $\mathscr{Z}$-domain is that the computation of $P(z)$ does not contain the term $Y_2(z)$ as normal Clenshaw's technique and this reduces the computation costs for generating the KMs. We now calculate the values of $K(z,0)$ and $K(z,1)$ in Equation (30) using (25). These values are: $K(z,0) = \frac{z}{z-1}$ as $K_n(0) = 1$ and $K(z,1) = b(z,0)K(z,0)$ with $b(z,0) = \frac{1}{N-1}\frac{z+\lambda}{z-1}$. Equation (30) can be simplified as

$$P(z) = \frac{z}{z-1}\Big[g(0) + \frac{1}{N-1}\frac{z+\lambda}{z-1}Y_1(z)\Big]. \tag{31}$$

By applying the inverse $\mathscr{Z}$-transform on Equation (31), replacing $g(0) = f(0)\sqrt{w(0)} = f(0)\sqrt{(1-p)^{N-1}}$, and taking the KMs definition of (26) into account, we have:

$$Q_n = \frac{1}{\sqrt{\rho(n)}}\Big[f(0)\sqrt{(1-p)^{N-1}} + \frac{1}{N-1}\, y_1(n) * (1+\frac{n}{p})\Big], \tag{32}$$

where the $*$ symbol denotes the 1D convolution and $y_1(n) = \mathscr{Z}^{-1}\{Y_1(z)\}$. With the help of the discrete convolution rules (see Appendix B), we can simplify the final KMs' computation in (32) as follows:

$$Q_n = \frac{1}{\sqrt{\rho(n)}}\Big\{f(0)\sqrt{(1-p)^{N-1}} + \frac{1}{N-1}\Big[\sum_{k=0}^{n} y_1(k) + \frac{1}{p}\sum_{k=0}^{n}(n-k)y_1(k)\Big]\Big\}. \tag{33}$$

Equation (33) describes new and fast computation algorithms of KMs based on Clenshaw's method and the concept of $\mathscr{Z}$-transform. As can be seen from this equation, to compute KMs, we need to calculate two simple summations of signal $y_1(n)$ to reach the desired order of KMs (nth order). To find the value of signal $y_1(n)$, we can use the inverse $\mathscr{Z}$-transform of $Y_1(z)$ formulised in the recurrence relations of $Y_x(z)$ in (29b):

$$Y_1(z) = \sum_{i=1}^{N-1} g(i) \prod_{k=1}^{i-1} b(z,k). \tag{34}$$

Substituting $b(z,k)$ from (25) into the above product and simplifying it will lead us to:

$$\begin{aligned} Y_1(z) &= \sum_{i=1}^{N-1} g(i) \prod_{k=1}^{i-1} \frac{k+1}{k+1-N}\frac{z+\lambda}{z-1} \\ &= \sum_{i=1}^{N-1} g(i) \prod_{k=1}^{i-1} \Big(\frac{z+\lambda}{z-1}\Big)^{i-1} \frac{\Gamma(i+1)}{(2-N)_{i-1}}, \end{aligned} \tag{35}$$

where $\Gamma(.)$ represents the Gamma function. Now, by taking the inverse $\mathscr{Z}$-transform of (35), we can find the value of signal $y_1(n)$ as follows:

$$y_1(n) = \sum_{i=1}^{N-1} g(i) \frac{\Gamma(i+1)}{(2-N)_{i-1}} r_i(n), \tag{36}$$

where $r_i(n) = \mathscr{Z}^{-1}\left\{\left(\frac{z+\lambda}{z-1}\right)^{i-1}\right\}$. The evaluated discrete signals as $r_i(n)$ can be easily calculated using the inverse $\mathscr{Z}$-transform. Figure 5 shows the calculated $r_i(n)$ for $i = 1, 2, 3, 4, 5$ and 6 assuming that $p = \frac{1}{2}$ $(\lambda = 1)$.

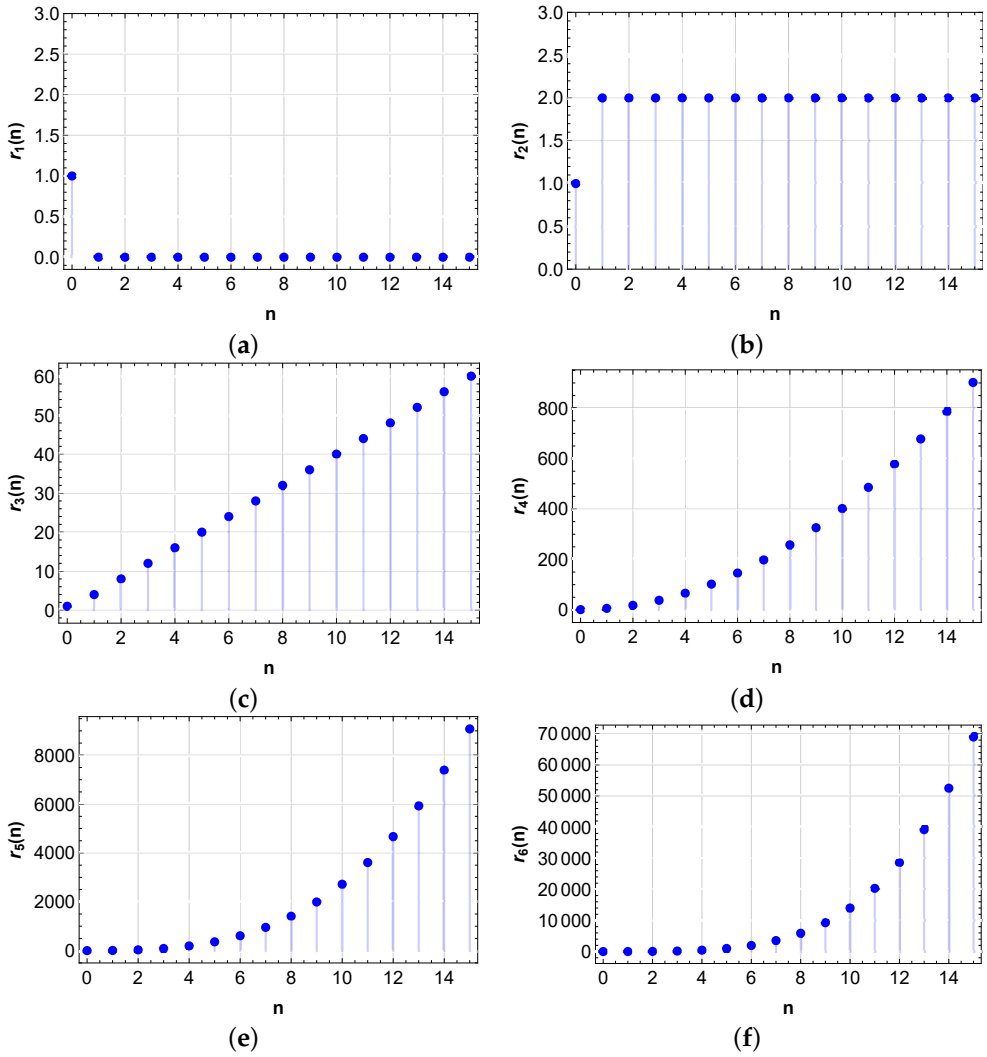


Figure 5. (**a**–**f**) Plots of signal $r_i(n)$ for $i = 1, 2, 3, 4, 5$, and 6 assuming that $p = \frac{1}{2}$ $(\lambda = 1)$.

5. Experimental Results

In this section, we illustrate the performance of the proposed method in terms of the computational time of the KMs. We first show the simple error analysis of the KPs computation with respect to the three mentioned recursive algorithms (with respect to the order n, to the independent variable x, and to both order and variable n, x). We then perform the signal reconstruction of the KMs with different values of the localisation parameter p. Finally, we compare the computational time for achieving KMs using the three mentioned recurrence techniques to show the superiority of the proposed four-term recursion for KPs/KMs.

5.1. Error Analysis of the KPs Recursive Computation

This subsection aims to focus on the stability of the three recursive formulas for computing the KPs. The stability of Equations (9) and (11) is the same since the KPs have symmetry properties in terms of the order n and the independent variable x. Therefore, the error analysis will be performed for Equations (9) and (11) and the proposed recursive algorithm in (23). Let us consider the coefficient of $\overline{K}_{n-1}(x)$ in (9) as $A_{n,x} = A_n\Big[p(N-2n+1)+n-x-1\Big]$ and the coefficient of $\overline{K}_n(x-1)$ in (11) as $C_{x,n} = C_x\Big[p(N-2x+1)+x-n-1\Big]$. Figure 6a illustrates the value of $|A_{n,x}|$ (or $|C_{x,n}|$) varying with n and x. We assumed $p = 0.5$ and $N = 800$. The figure shows the variation of this coefficient for starting x with 0 and stopping at 800 and using an increment of 100. The plot also shows that $|A_{n,x}|$ ranges from its minimum value, $\frac{p(3-N)-1}{\sqrt{p(1-p)(N-1)}}$ to its maximum value, $\sqrt{\frac{p(N-1)}{1-p}}$. Figure 6b shows the variation of $|B_n|$ (or $|D_x|$) with respect to n (or x). From this plot, we can recognise that this coefficient is approximately 1. Hence, $|A_{n,x}|$ (or $|C_{x,n}|$) has a greater effect on the propagation of the KPs error than $|B_n|$ (or $|D_x|$) does by using these recursion methods. Moreover, this figure mentions that the n (or x)-recursive computation of the edge points would generate larger errors than the centre points.

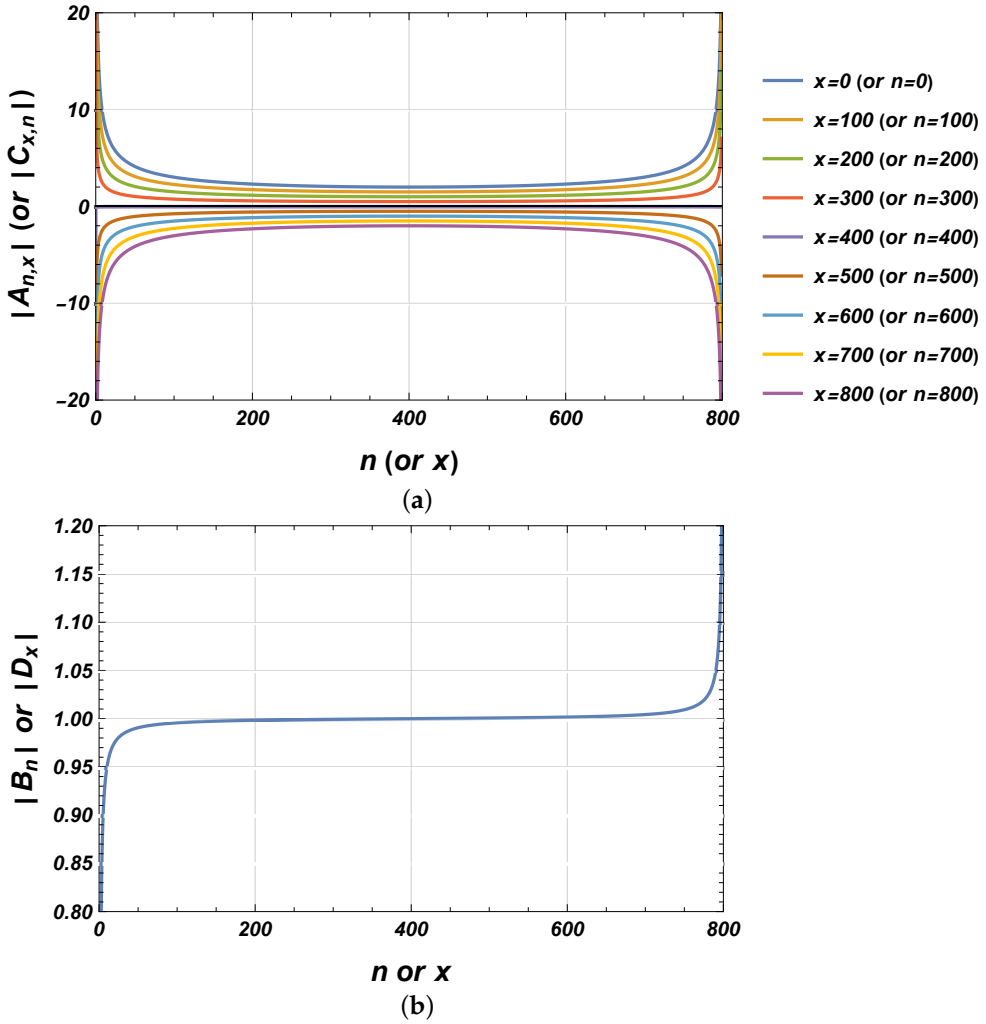


Figure 6. The plots of the KPs' coefficients in (9) or (11): (**a**) plot of $|A_{n,x}|$ (or $|C_{x,n}|$) with respect to n (or x), and (**b**) plot of $|B_n|$ (or $|D_x|$) with respect to n (or x). Assuming $p = 0.5$ and $N = 800$.

Figure 7 shows the stability of the proposed recursive algorithm of the KPs for its four-term recurrence coefficients. We assumed that $p = 0.5$ and $N = 800$. Figure 7a indicates that the y-axis is the asymptote of $\alpha(x)$ and has a minimum value of zero at the endpoint (N). Part (b) of this figure shows the variation of $\beta(n)$ with respect to n. The graph has a minimum of 0 at 0 and a maximum of $\sqrt{\frac{N}{\lambda}}$ at $N-1$. Therefore, both coefficients are stable. Figure 7c illustrates the changes of $\gamma(n,x)$ in terms of n with different values of x. All the graphs have a minimum value of 0 at 0 and a maximum value of $\sqrt{\frac{(N-1)(n-x)}{x}}$ depending on the selected magnitude of $0 \leq x \leq N-1$. An instability of $\gamma(n,x)$ for $x = 0$ can be recognised.

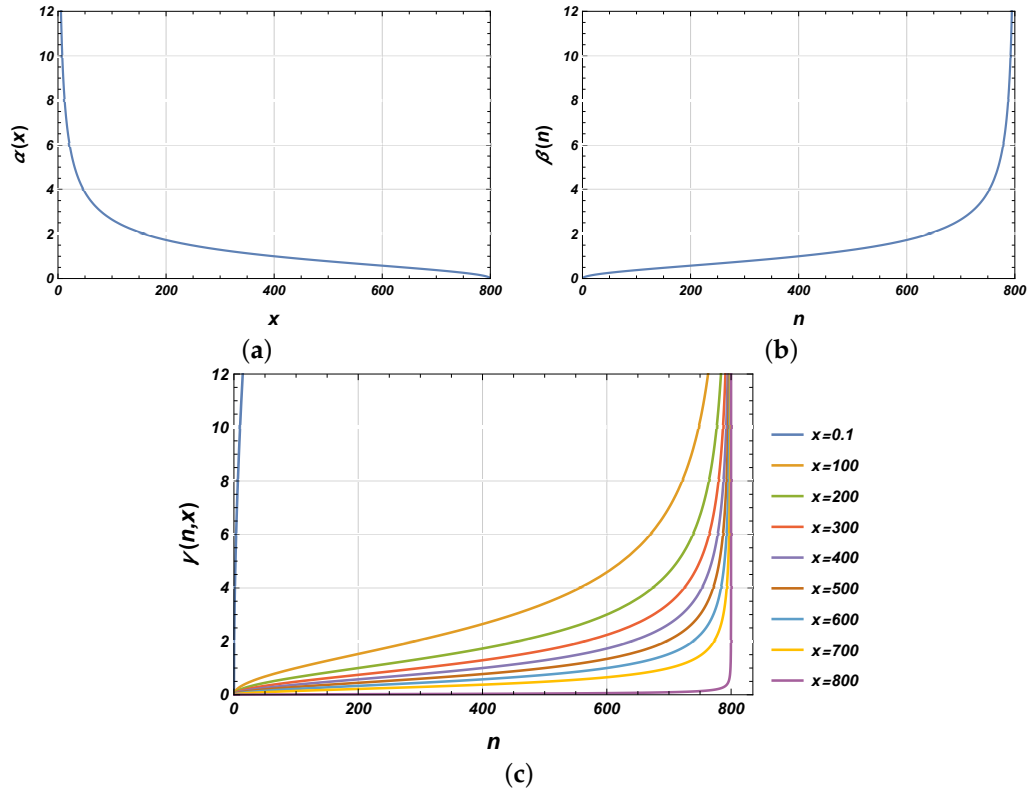


Figure 7. The plots of the KPs' coefficients in (23): (**a**) plot of $\alpha(x)$ with respect to x, (**b**) plot of $\beta(n)$ with respect to n, and (**c**) plot of $\gamma(n,x)$ with respect to n with different values of x. Assuming $p = 0.5$ and $N = 800$.

5.2. Signal Reconstruction

We performed the experiments on four real 1D signals, which were extracted as line segments from the commonly available test image "Boat". To ensure the diversity of the data, the signals were taken from different parts of the image, see Figure 8. All extracted signals have a length of 128 samples (see Figure 9 for the signal plots). The first signal (red) is extracted from row 100, columns 281 to 409, the second signal (blue) is extracted from row 256, columns 1 to 128, the third signal (green) is extracted from row 310, columns 201 to 329, and the fourth signal (orange) is extracted from row 410, columns 311 to 439.

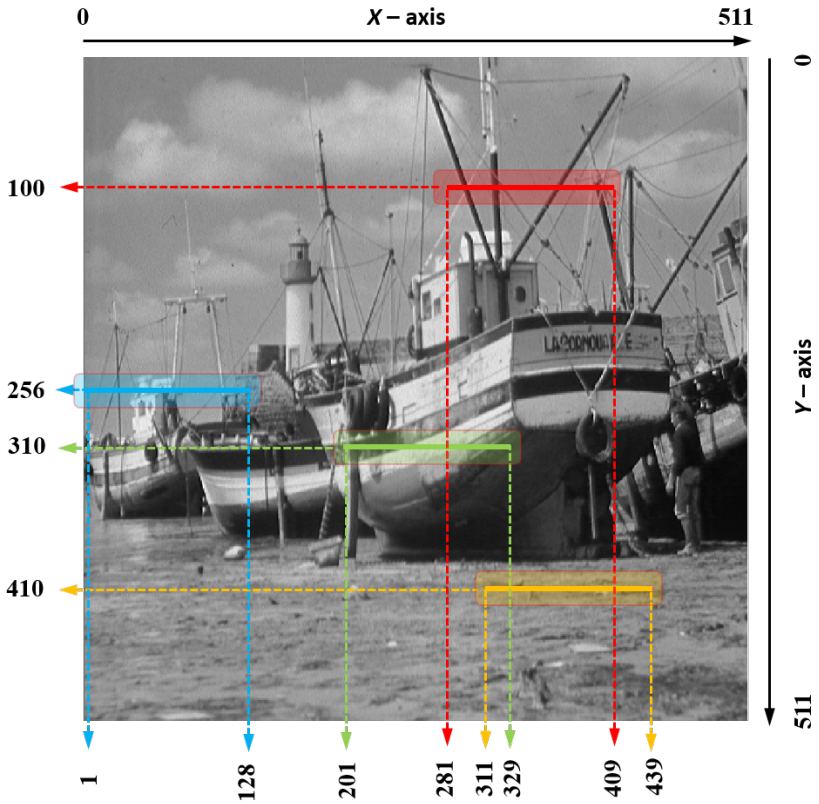


Figure 8. Test grayscale image with the size of 512 × 512 (Boat): extracted 1D signals with the length of 128 highlighted by red, blue, green, and orange colours.

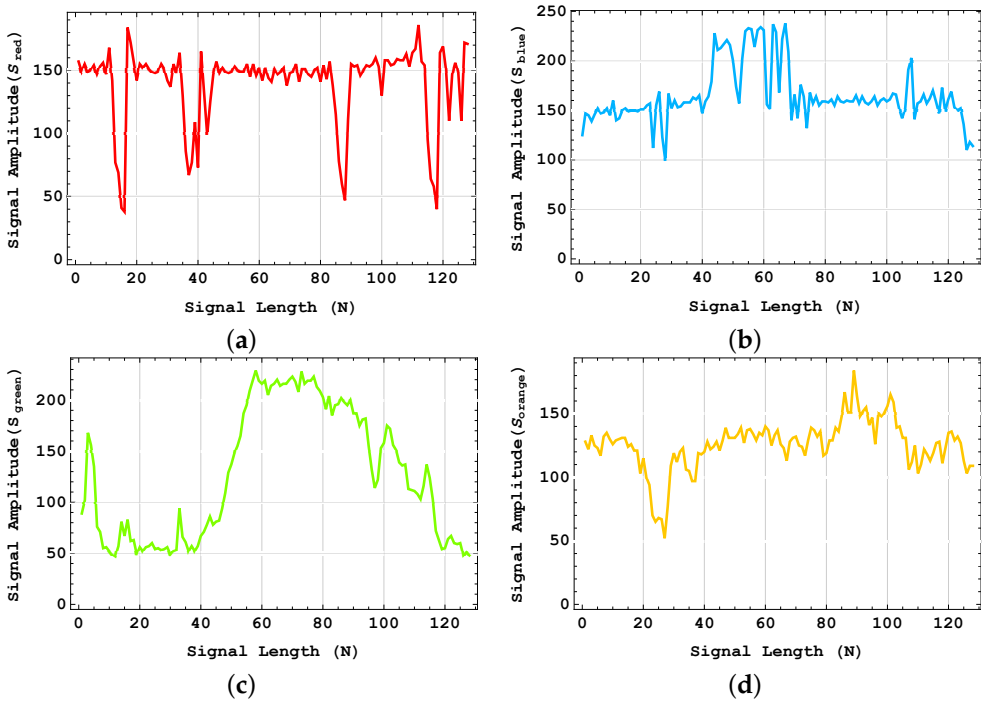


Figure 9. The plots of four test signals extracted from the "Boat" image: (**a**) row 100, (**b**) row 256, (**c**) row 310, and (**d**) row 410.

In this experiment, we calculated the KMs based on the proposed recursive equation and performed partial reconstruction from KMs of orders up to 10, 50, 90, and 125 (for the 128th order the reconstruction becomes complete). The pseudo-code implementation of the KMs and its inverse transform is given in Algorithm 1.

Algorithm 1: Algorithm in pseudo-code for implementing the new four-term recurrence of the 1D Krawtchouk moment computation.

Input : Signal $f(x)$; $0 \leq x \leq N-1$
Output: Computation of Krawtchouk moments based on Clenshaw's summation using a four-term recurrence

```
Compute Krawtchouk polynomials using four-term recursion from (23)
Applying Clenshaw formula for the computation of Krawtchouk moments
from (33)
Computation of y_1(n) using (36) for x ← 0 to N − 1 do
    for n ← 0 to N − 1 do
        compute Q_n from Equation (13)
    end
end
for n ← 0 to N_max do
    for x ← 0 to N − 1 do
        compute f̃(x) from Equation (14)
    end
end
```

Figures 10–13 illustrate the reconstruction process for the signals shown in Figure 9. We perform the experiments for different values of the localisation parameter p. The values of p for red, blue, green, and orange signals are 0.25, 0.5, 0.75, and 0.95, respectively. It can be observed from the reconstructed signals that the convergence of the recovered signal depends on the value of p. For example, in Figure 10b, the black signal almost covers the first region of the original red signal (p = 0.25) while in Figure 12b, the black signal covers the final segment of the original green signal (p = 0.95). For both mentioned cases, the maximum reconstruction order is 50. It can be seen from the results that the reconstruction always converges to the original signal. However, the partial reconstructions from the proposed recursion, especially from orders less than 10, are different.

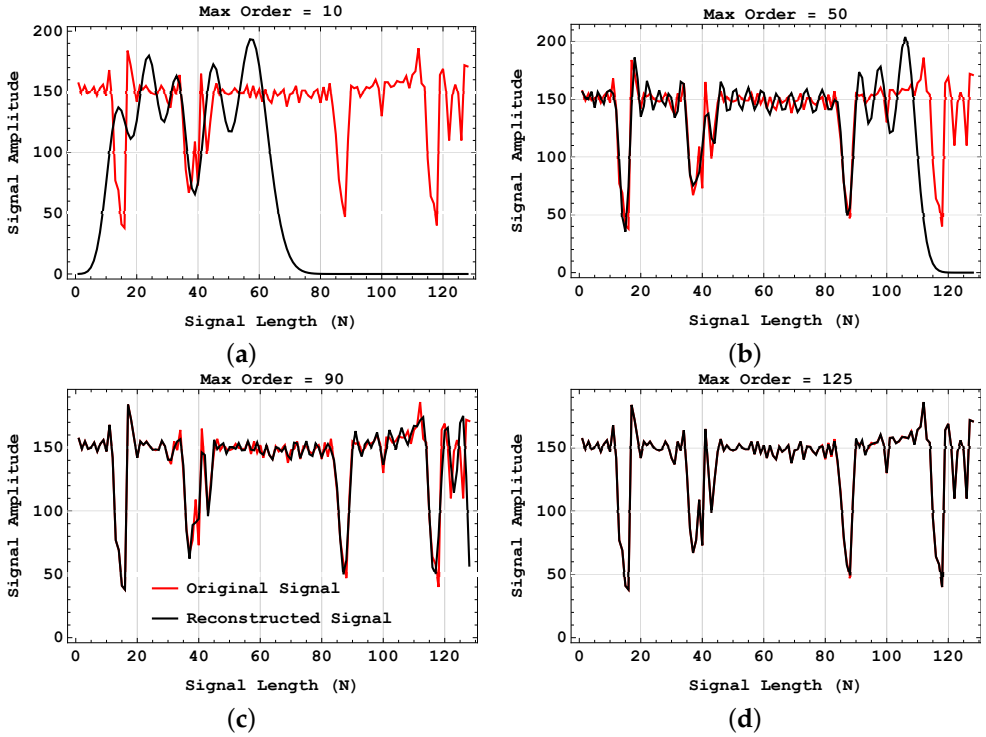


Figure 10. Signal reconstruction of the red signal in Figure 9a with different maximum orders: (**a**) 10, (**b**) 50, (**c**) 90, and (**d**) 125. $p = 0.25$.

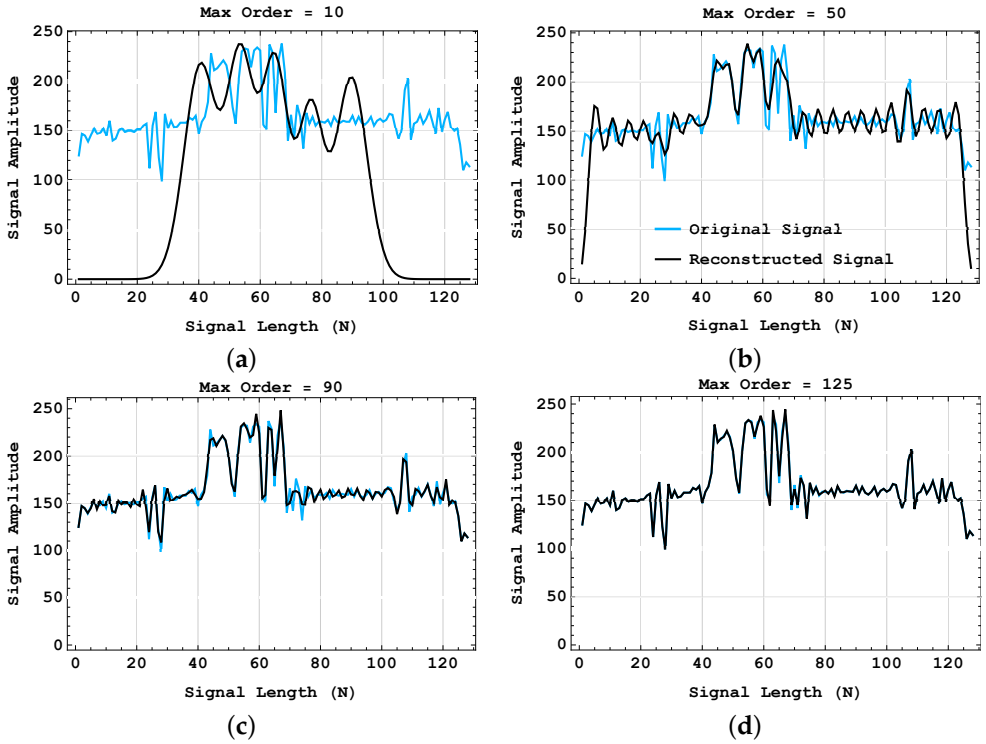


Figure 11. Signal reconstruction of the blue signal in Figure 9b with different maximum orders: (**a**) 10, (**b**) 50, (**c**) 90, and (**d**) 125. $p = 0.5$.

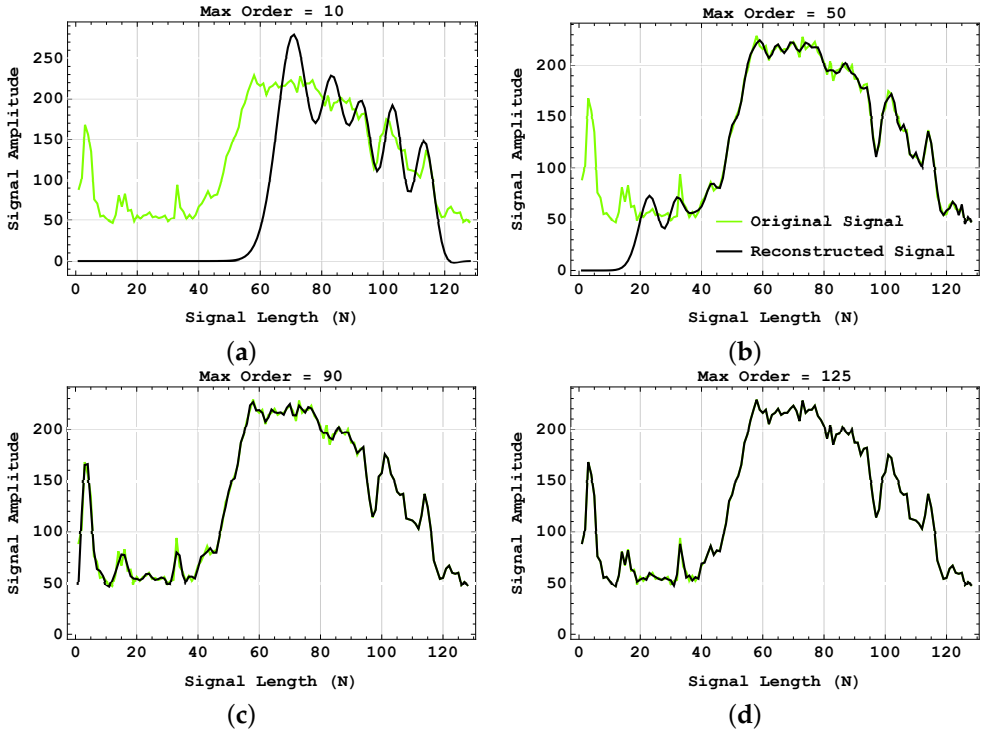


Figure 12. Signal reconstruction of the green signal in Figure 9c with different maximum orders: (**a**) 10, (**b**) 50, (**c**) 90, and (**d**) 125. $p = 0.75$.

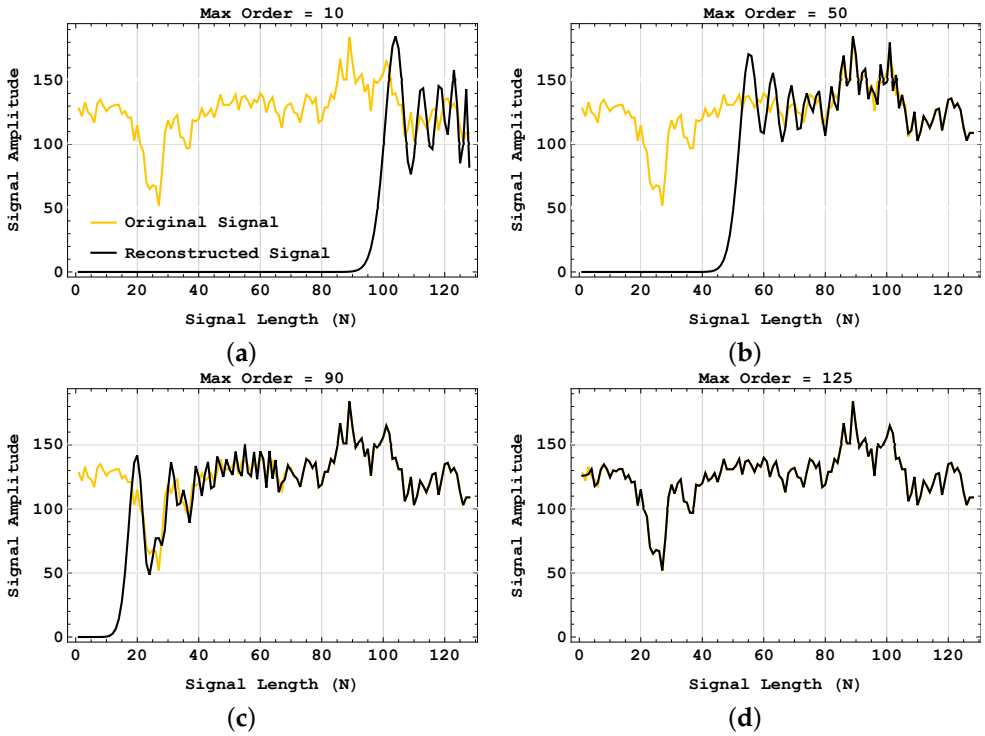


Figure 13. Signal reconstruction of the orange signal in Figure 9d with different maximum orders: (**a**) 10, (**b**) 50, (**c**) 90, and (**d**) 125. $p = 0.95$.

5.3. Computational Time

Moment computation is always time-consuming and fast computational algorithms may help greatly. Here, we compare the proposed method's computational time with the two existing and common recursions. Two experiments will be performed: first for computing the KPs using the recursive methods, and the second experiment to calculate the computation time of the KMs using the proposed method and the existing recursions.

5.3.1. Computational Time of KPs

To evaluate the performance of the proposed four-term recurrence relation of the KPs, we run an experiment to show the computation cost of three different methods. A comparison of computation time is performed for several polynomials size $(N = 500, 1000, 2000, ..., 10,000)$. All execution time is accomplished using the Krawtchouk localisation parameter $p = 0.5$. The required computation time for each recursive method to generate the KPs of size N is shown in Figure 14. We run the CPU elapsed times 10 times for each recursive algorithm and illustrate the average executed time in this figure. It can be noticed from the figure that the computational time for the three-term recursions is almost identical due to the symmetry property of the KPs (Equation (8)). The proposed four-term recursion algorithm to calculate the KPs up to the size of 10,000 performs a faster approach than the existing methods. To confirm the computed time using the proposed algorithm, we can measure the improvement ratio of the execution time. For example, the improvements of the proposed method compared to both existing methods for $N = 6000, 7000$ and 8000 are 89.899%, 93.525% and 95.656%, respectively.

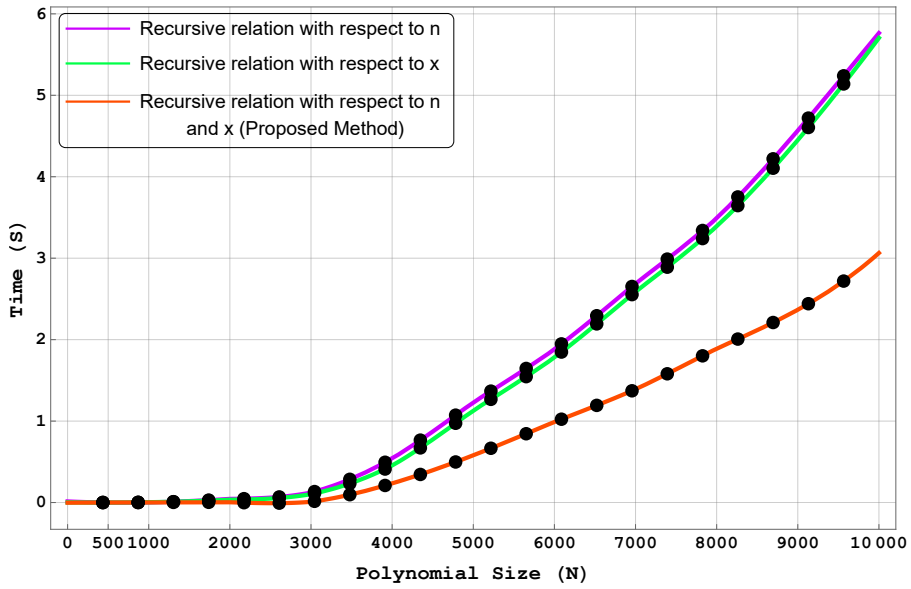


Figure 14. Computation time of the KPs using the proposed four-term recursion and existing three-term recursions. The minimum and maximum KPs sizes are 500 and 10,000, respectively. Assuming $p = 0.5$.

5.3.2. Computational Time of KMs

As the computation of almost entire image orthogonal moments is time-consuming due to the presence of hypergeometric series, falling and rising factorials, the Pochhammer function and the gamma function, recursive algorithms are usually adopted to calculate the polynomial coefficients or kernels.

In this subsection, we compute and compare the execution time to generate the KMs using three recursive methods: recursive relation with respect to n, x, and both n and x, which is based on the proposed algorithm derived from the Clenshaw method (Algorithm 1). In this experiment, the performance of the proposed algorithm is demonstrated with the sample word 'Kravchuk' which is collected from a female subject of Ukrainian background

using Google Translate. The number of samples for this speech signal is 23,000. Figure 15a shows this speech signal. To simplify the experiment, we focus on two syllables of the word 'Kravchuk'. Figure 15b,c indicate the left side (first syllable: **'Krav'**) which contains the samples from 3000 to 9000 and the right side (second syllable: **'chuk'**) of this test word that contains the samples from 11,000 to 17,000. Both selected syllables have the same length of 6000.

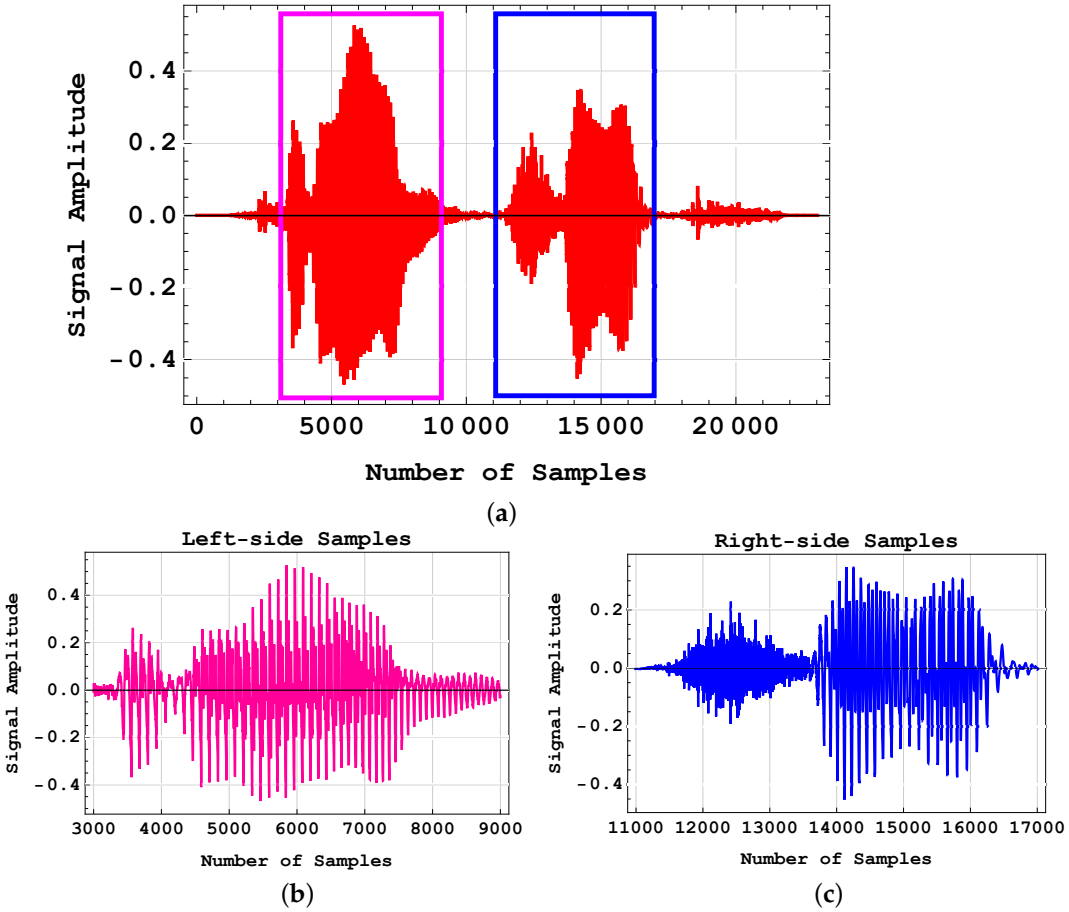


Figure 15. Test speech signal: (**a**) the signal representation of the word 'Kravchuk' with 23000 samples, (**b**) the first syllable of this word ('Krav'), and (**c**) the second syllable of this word ('chuk').

We calculate the KMs of the above signals using the following fast algorithms:

- Direct recursive relation with respect to n or x (Method 1)
- Clenshaw's recurrence formula [43] (Method 2)
- Digital filter direct computation [25] (Method 3)
- The proposed algorithm based on four-term recursive relation using Clenshaw's formula (Proposed method)

For the first signal (left side), we set $p = 0.5$ and in the second signal (right side), we assume $p = 0.75$. Since both signals have the same length of 6000, we compute the KMs up to order 5999 and calculate the elapsed CPU time for them. We compare the performance of the proposed method with the aforementioned existing algorithms (Methods 1, 2, and 3). Figures 16 and 17 show the average elapsed CPU time of the KMs computation.

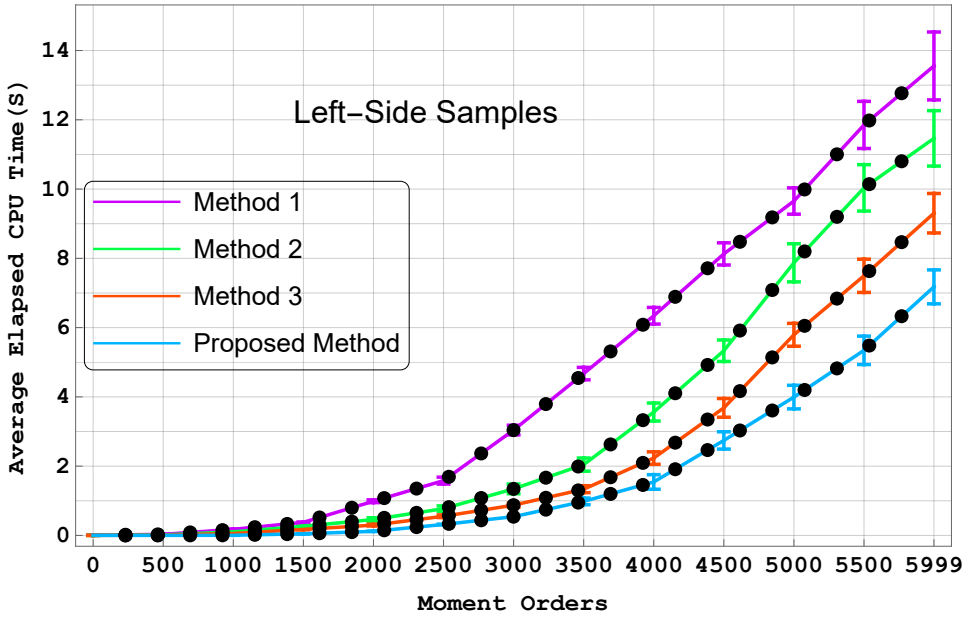


Figure 16. Average elapsed CPU times in seconds with error bar: Computation of full set of KMs for left-side samples of the speech signal of size 3000 shown in Figure 15b. Assuming $p = 0.5$.

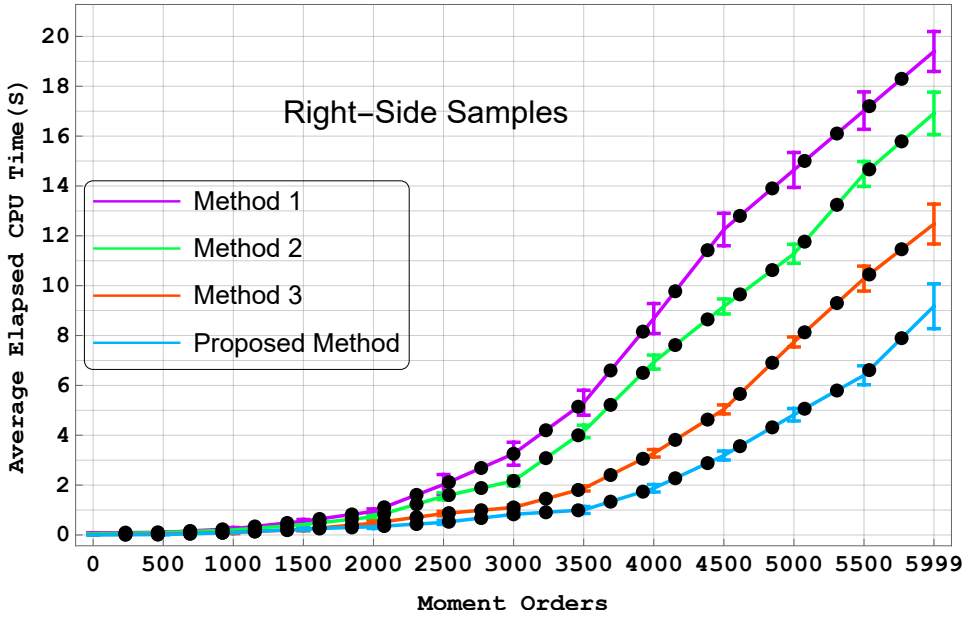


Figure 17. Average elapsed CPU times in seconds with error bar: Computation of full set of KMs for right-side samples of the speech signal of size 3000 shown in Figure 15c. Assuming $p = 0.75$.

The result indicates that the proposed four-term recursive method based on Clenshaw's algorithm has a better performance than the recurrence relation with respect to n or x in terms of the computation time of the KMs. Method 3 proposed by [25] is a fast digital filter-based approach which is faster than the three-term recursive algorithms. As can be seen from Figure 16, method 3 is 25.6% faster than the average time of Methods 1 and 2 for the maximum order of the KMs (5999), while the proposed method is 22.85% faster than Method 3 for the same order of KMs. For the right side sample of the test speech signal, our method is 49.46% faster than Methods 1 and 2 for the maximum order of the KMs (5999), while the proposed algorithm is 26.44% faster than Method 3 for the same KMs' order. In all these cases the results are consistent with the first experiment, regardless of the signal size. The computation of the KMs using the proposed method is significantly faster

than the other existing methods. In both figures, we use the error bar for each technique to represent the uncertainty or variation of the average time for each moment order.

6. Conclusions and Future Work

The contribution of this paper lies in the efficient four-term recursive algorithm for the calculation of discrete KPs. We combine this new recursion with Clenshaw's method using $\mathscr{Z}$-transform properties to find an accelerated technique to compute KMs which are important for image analysis. We have proposed a lattice network to compute KPs using only two initial values of them. The proposed recursion has less complexity than the existing three-term recursions. A couple of experiments were conducted to validate the theoretical formula. The first experiment performed the error analysis of the KPs using recursive computation. The stability and solidity of the four-term recursive formula for computing the KPs as the KMs kernel helped us to merge this approach with Clenshaw's formula to accelerate the computation time of the KMs. The second experiment showed the capability of the proposed algorithm for signal reconstruction. Finally, a computational time test was performed to evaluate the elapsed CPU time of the KPs and KMs calculation for a large size of polynomials and a full set of moment orders using a speech signal with a number of samples of 23,000. In terms of speed, which was our main concern, we showed that the proposed method performed significantly faster than the reference recursive methods.

This paper studied only 1D KPs and moments. Extension of the fast computation of KMs for image analysis as 2D signals will be considered in future work by using 2D $\mathscr{Z}$-transform concepts. We can define a 2D lattice network that will be added to compute two sets of KPs with different degrees in parallel. By doing so, we can study the performance of the new proposed algorithm in all traditional tasks of 2D and 3D image analysis, where KMs have been commonly applied in object description and recognition, watermarking, image registration, and many others.

This research could pave the way for fast computation of several orthogonal moment functions since most of these functions are based on the hypergeometric series. It is possible to find the higher-order recursions which are independent of the polynomial parameters/variables and accelerate the calculation of moment functions not only for 1D but also for 2D and 3D signal analysis. Moreover, the design of digital filter architectures for moment functions by means of the $\mathscr{Z}$-transform idea can be considered as one the interesting and challenging problems for future work.

Author Contributions: Conceptualization, B.H.S.A.; methodology, B.H.S.A.; resources, B.H.S.A. and M.H.R.; writing—original draft preparation, B.H.S.A. and M.H.R.; writing—review and editing, B.H.S.A. and M.H.R.; visualization, B.H.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Simplifying the summation contains the exponential term of z^{k+1}:

$$a \sum_{k=0}^{\infty} \frac{(a+1)_k (b+1)_k}{(c)_k} \frac{z^{k+1}}{k!}. \tag{A1}$$

Let $m = k + 1$. We then rewrite (A1) as

$$a \sum_{m=1}^{\infty} \frac{(a+1)_{m-1}(b+1)_{m-1}}{(c)_{m-1}} \frac{z^m}{(m-1)!}, \tag{A2}$$

By using the last identity in (18), we have

$$a \sum_{m=1}^{\infty} \frac{(a)_m (b)_m}{ab} \frac{c-1}{(c-1)_m} \frac{m z^m}{m!}, \tag{A3}$$

Finally, the second identity in (18) can be applied to variable c as follows (notice that because of the presence of index m in the numerator, it is possible to start the lower band of summation from 0):

$$a \sum_{m=0}^{\infty} \frac{(a)_m (b)_m}{ab} \frac{(c+m-1)m}{(c)_m} \frac{z^m}{m!}. \tag{A4}$$

Appendix B

Convolution identities based on the unit-step function:

$$x(n) * u(n) = \sum_{k=-\infty}^{n} x(k) \tag{A5}$$

$$x(n) * nu(n) = \sum_{k=-\infty}^{n} (n-k)x(k) \tag{A6}$$

References

1. Gogin, N.; Hirvensalo, M. On the generating function of discrete Chebyshev polynomials. *J. Math. Sci.* **2016**, *448*, 124–134. [CrossRef]
2. Flusser, J.; Suk, T.; Zitová, B. *2D and 3D Image Analysis by Moments*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
3. Belkasim, S.; Shridhar, M.; Ahmadi, M. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognit.* **1991**, *24*, 1117–1138. [CrossRef]
4. Flusser, J.; Suk, T. Pattern recognition by affine moment invariants. *Pattern Recognit.* **1993**, *26*, 167–174. [CrossRef]
5. Asli, B.H.S.; Flusser, J.; Zhao, Y.; Erkoyuncu, J.A. Filter-generating system of Zernike polynomials. *Automatica* **2019**, *108*, 108498. [CrossRef]
6. Hsu, H.S.; Tsai, W.H. Moment-preserving edge detection and its application to image data compression. *Opt. Eng.* **1993**, *32*, 1596–1608. [CrossRef]
7. Asmara, R.A.; Mentari, M.; Putri, N.S.H.; Handayani, A.N. Identification of Toga Plants Based on Leaf Image Using the Invariant Moment and Edge Detection Features. In Proceedings of the IEEE 2020 4th International Conference on Vocational Education and Training (ICOVET), Malang, Indonesia, 19 September 2020; pp. 75–80.
8. Xiao, B.; Lu, G.; Zhang, Y.; Li, W.; Wang, G. Lossless image compression based on integer Discrete Tchebichef Transform. *Neurocomputing* **2016**, *214*, 587–593. [CrossRef]
9. Benouini, R.; Batioua, I.; Zenkouar, K.; Zahi, A.; El Fadili, H.; Qjidaa, H. Fast and accurate computation of Racah moment invariants for image classification. *Pattern Recognit.* **2019**, *91*, 100–110. [CrossRef]
10. Sayyouri, M.; Hmimid, A.; Qjidaa, H. Improving the performance of image classification by Hahn moment invariants. *JOSA A* **2013**, *30*, 2381–2394. [CrossRef] [PubMed]
11. Honarvar, B.; Paramesran, R.; Lim, C.L. Image reconstruction from a complete set of geometric and complex moments. *Signal Process.* **2014**, *98*, 224–232. [CrossRef]
12. Asli, B.H.S.; Paramesran, R.; Lim, C.L. The fast recursive computation of Tchebichef moment and its inverse transform based on Z-transform. *Digit. Signal Process.* **2013**, *23*, 1738–1746. [CrossRef]
13. Deng, A.W.; Gwo, C.Y. Efficient computations for generalized Zernike moments and image recovery. *Appl. Math. Comput.* **2018**, *339*, 308–322. [CrossRef]
14. Hosny, K.M.; Khalid, A.M.; Mohamed, E.R. Efficient compression of volumetric medical images using Legendre moments and differential evolution. *Soft Comput.* **2020**, *24*, 409–427. [CrossRef]
15. Liao, S.X.; Pawlak, M. On the accuracy of Zernike moments for image analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1358–1364. [CrossRef]
16. Shakibaei, B.H.; Paramesran, R. Recursive formula to compute Zernike radial polynomials. *Opt. Lett.* **2013**, *38*, 2487–2489. [CrossRef]

17. Yang, B.; Dai, M. Image analysis by Gaussian–Hermite moments. *Signal Process.* **2011**, *91*, 2290–2303. [CrossRef]
18. Chong, C.W.; Raveendran, P.; Mukundan, R. The scale invariants of pseudo-Zernike moments. *Pattern Anal. Appl.* **2003**, *6*, 176–184. [CrossRef]
19. Larbi, G. Two-step text detection framework in natural scenes based on Pseudo-Zernike moments and CNN. *Multimed. Tools Appl.* **2022**, 82, 10595–10616. [CrossRef]
20. Wang, X.y.; Wang, L.; Tian, J.l.; Niu, P.p.; Yang, H.y. Color image zero-watermarking using accurate quaternion generalized orthogonal fourier–mellin moments. *J. Math. Imaging Vis.* **2021**, *63*, 708–734. [CrossRef]
21. Shakibaei, B.H.; Paramesran, R. Fourier-Mellin expansion coefficients of scaled pupils. *Chin. Opt. Lett.* **2013**, *11*, 080101. [CrossRef]
22. Mukundan, R.; Ong, S.; Lee, P.A. Image analysis by Tchebichef moments. *IEEE Trans. Image Process.* **2001**, *10*, 1357–1364. [CrossRef]
23. Chang, K.H.; Paramesran, R.; Asli, B.H.S.; Lim, C.L. Efficient hardware accelerators for the computation of Tchebichef moments. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *22*, 414–425. [CrossRef]
24. Yap, P.T.; Paramesran, R.; Ong, S.H. Image analysis by Krawtchouk moments. *IEEE Trans. Image Process.* **2003**, *12*, 1367–1377.
25. Asli, B.H.S.; Flusser, J. Fast computation of Krawtchouk moments. *Inf. Sci.* **2014**, *288*, 73–86. [CrossRef]
26. Daoui, A.; Yamni, M.; Karmouni, H.; Sayyouri, M.; Qjidaa, H. Stable computation of higher order Charlier moments for signal and image reconstruction. *Inf. Sci.* **2020**, *521*, 251–276. [CrossRef]
27. Yamni, M.; Daoui, A.; Karmouni, H.; Sayyouri, M.; Qjidaa, H.; Maaroufi, M.; Alami, B. Fast and Accurate Computation of 3D Charlier Moment Invariants for 3D Image Classification. *Circuits, Syst. Signal Process.* **2021**, *40*, 6193–6223. [CrossRef]
28. Jahid, T.; Hmimid, A.; Karmouni, H.; Sayyouri, M.; Qjidaa, H.; Rezzouk, A. Image analysis by Meixner moments and a digital filter. *Multimed. Tools Appl.* **2018**, *77*, 19811–19831. [CrossRef]
29. Karmouni, H.; Jahid, T.; Hmimid, A.; Sayyouri, M.; Qjidaa, H. Fast computation of inverse Meixner moments transform using Clenshaw's formula. *Multimed. Tools Appl.* **2019**, *78*, 31245–31265. [CrossRef]
30. Spiliotis, I.M.; Karampasis, N.D.; Boutalis, Y.S. Fast computation of Hahn moments on gray images using block representation. *J. Electron. Imaging* **2020**, *29*, 013020. [CrossRef]
31. Zhu, H.; Shu, H.; Zhou, J.; Luo, L.; Coatrieux, J.L. Image analysis by discrete orthogonal dual Hahn moments. *Pattern Recognit. Lett.* **2007**, *28*, 1688–1704. [CrossRef]
32. Zhu, H.; Shu, H.; Liang, J.; Luo, L.; Coatrieux, J.L. Image analysis by discrete orthogonal Racah moments. *Signal Process.* **2007**, *87*, 687–708. [CrossRef]
33. Abdulhussain, S.H.; Mahmmod, B.M. Fast and efficient recursive algorithm of Meixner polynomials. *J.-Real-Time Image Process.* **2021**, *18*, 2225–2237. [CrossRef]
34. Koehl, P. Fast recursive computation of 3d geometric moments from surface meshes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2158–2163. [CrossRef]
35. Abdulhussain, S.H.; Ramli, A.R.; Al-Haddad, S.A.R.; Mahmmod, B.M.; Jassim, W.A. Fast recursive computation of Krawtchouk polynomials. *J. Math. Imaging Vis.* **2018**, *60*, 285–303. [CrossRef]
36. Karmouni, H.; Hmimid, A.; Jahid, T.; Sayyouri, M.; Qjidaa, H.; Rezzouk, A. Fast and stable computation of the Charlier moments and their inverses using digital filters and image block representation. *Circuits Syst. Signal Process.* **2018**, *37*, 4015–4033. [CrossRef]
37. Spiliotis, I.M.; Mertzios, B.G. Real-time computation of two-dimensional moments on binary images using image block representation. *IEEE Trans. Image Process.* **1998**, *7*, 1609–1615. [CrossRef]
38. Gutleb, T.S.; Olver, S. A sparse spectral method for Volterra integral equations using orthogonal polynomials on the triangle. *SIAM J. Numer. Anal.* **2020**, *58*, 1993–2018. [CrossRef]
39. Gutleb, T.S. A fast sparse spectral method for nonlinear integro-differential Volterra equations with general kernels. *Adv. Comput. Math.* **2021**, *47*, 42. [CrossRef]
40. Fan, Q.Q.; Zhang, H.W.; Lei, W.W. Fast algorithm for calculating geoid undulation based on Clenshaw technique. *Prog. Geophys.* **2020**, *35*, 823–828.
41. Ledoux, V.; Moroz, G. Evaluation of Chebyshev polynomials on intervals and application to root finding. In Proceedings of the Mathematical Aspects of Computer and Information Sciences: 8th International Conference, MACIS 2019, Gebze, Turkey, 13–15 November 2019; Revised Selected Papers 8; Springer: Cham, Switzerland, 2020; pp. 35–41.
42. Raj, P.A.; Venkataramana, A. Fast computation of inverse Krawtchouk moment transform using Clenshaw's recurrence formula. In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 20 November 2007; Volume 4, pp. 4–37.
43. Jahid, T.; Karmouni, H.; Hmimid, A.; Sayyouri, M.; Qjidaa, H. Image moments and reconstruction by Krawtchouk via Clenshaw's reccurence formula. In Proceedings of the 2017 International Conference on Electrical and Information Technologies (ICEIT), Rabat, Morroco, 4–7 March 2017; pp. 1–7.
44. Venkataramana, A.; Raj, P.A. Recursive computation of forward Krawtchouk moment transform using Clenshaw's recurrence formula. In Proceedings of the IEEE 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Karnataka, India, 15–17 December 2011; pp. 200–203.
45. Mahmmod, B.M.; Abdul-Hadi, A.M.; Abdulhussain, S.H.; Hussien, A. On computational aspects of Krawtchouk polynomials for high orders. *J. Imaging* **2020**, *6*, 81. [CrossRef]

46. Zhang, G.; Luo, Z.; Fu, B.; Li, B.; Liao, J.; Fan, X.; Xi, Z. A symmetry and bi-recursive algorithm of accurately computing Krawtchouk moments. *Pattern Recognit. Lett.* **2010**, *31*, 548–554. [CrossRef]
47. Al-Utaibi, K.A.; Abdulhussain, S.H.; Mahmmod, B.M.; Naser, M.A.; Alsabah, M.; Sait, S.M. Reliable recurrence algorithm for high-order Krawtchouk polynomials. *Entropy* **2021**, *23*, 1162. [CrossRef]
48. Jassim, W.A.; Raveendran, P.; Mukundan, R. New orthogonal polynomials for speech signal and image processing. *IET Signal Process.* **2012**, *6*, 713–723. [CrossRef]
49. Krawtchouk, M. Sur une généralisation des polynomes d'Hermite. *Comptes Rendus* **1929**, *189*, 5-3.
50. Meixner, J. Orthogonale Polynomsysteme mit einer besonderen Gestalt der erzeugenden Funktion. *J. Lond. Math. Soc.* **1934**, *1*, 6–13. [CrossRef]
51. Feinsilver, P.; Kocik, J. Krawtchouk polynomials and Krawtchouk matrices. In *Recent Advances in Applied Probability*; Springer: Cham, Switzerland, 2005; pp. 115–141.
52. Hypergeometric Functions in Wolfram: Hypergeometric2F1. Wolfram Research, Inc. pp. 1998–2023. Available online: https://functions.wolfram.com/HypergeometricFunctions/Hypergeometric2F1/ (accessed on 1 January 2022).
53. Idan, Z.N.; Abdulhussain, S.H.; Al-Haddad, S.A.R. A new separable moments based on Tchebichef-Krawtchouk polynomials. *IEEE Access* **2020**, *8*, 41013–41025. [CrossRef]
54. Karmouni, H.; Jahid, T.; Lakhili, Z.; Hmimid, A.; Sayyouri, M.; Qjidaa, H.; Rezzouk, A. Image reconstruction by Krawtchouk moments via digital filter. In Proceedings of the IEEE 2017 Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 18–19 May 2017; pp. 1–7.
55. Koekoek, R.; Lesky, P.A.; Swarttouw, R.F. *Hypergeometric Orthogonal Polynomials and Their Q-Analogues*; Springer Science & Business Media: New York, NY, USA, 2010.
56. Yánez, R.; Dehesa, J.; Zarzo, A. Four-term recurrence relations of hypergeometric-type polynomials. *Il Nuovo Cimento B (1971–1996)* **1994**, *109*, 725–733. [CrossRef]