Step 1 - The value of $\hat{C}_n^p(0)$ is computed using Eq. (25).

Step 2 - The value of $\hat{C}_n^p(0)$ are computed as follows.

A- The values of $\hat{C}_n^p(0)$ within the range $n = 0, 1, \ldots, p - 1$ are computed using (26).

B- Eq. (27) is used to compute the values of $\hat{C}_n^p(0)$ within the range $n = p + 1, p + 2, \ldots, N - 1$.

Step 3 - The values of $\hat{C}_n^p(1)$ is computed using Eq. (28) within the range $n = 0, 1, \ldots, N - 1$.

Step 4 - Eq. (36) is used to compute CHPCs within the ranges $x = 2, 3, \ldots, N - 1$ and $n = x, x + 1, \ldots, N - 1$.

Step 5 - Lastly, CHPCs within the ranges $x = 2, 3, \ldots, N - 1$ and $n = 0, 1, \ldots, x - 1$ are defined in (35) by using the similarity identity.
COMPUTER SCIENCE | RESEARCH ARTICLE

On the computational aspects of Charlier polynomials

Alaa M. Abdul-Hadi1, Sadiq H. Abdulhussain1* and Basheera M. Mahammad1

Abstract: Charlier polynomials (CHPs) and their moments are commonly used in image processing due to their salient performance in the analysis of signals and their capability in signal representation. The major issue of CHPs is the numerical instability of coefficients for high-order polynomials. In this study, a new recurrence algorithm is proposed to generate CHPs for high-order polynomials. First, sufficient initial values are obtained mathematically. Second, the reduced form of the recurrence algorithm is determined. Finally, a new symmetry relation for CHPs is realized to reduce the number of recurrence times. The symmetry relation is applied to calculate 50% of the polynomial coefficients. The performance of the proposed recurrence algorithm is evaluated in terms of computational cost and reconstruction error. The evaluation involves a comparison with existing recurrence algorithms. Moreover, the maximum size that can be generated using the proposed recurrence algorithm is investigated and compared with those of existing recurrence algorithms. Comparison results indicate that the proposed algorithm exhibits better performance because it can generate a polynomial 44 times faster than existing recurrence algorithms. In addition, the improvement of the proposed algorithm is evaluated in terms of computational cost and reconstruction error. The evaluation involves a comparison with existing recurrence algorithms. Moreover, the maximum size that can be generated using the proposed recurrence algorithm is investigated and compared with those of existing recurrence algorithms. Comparison results indicate that the proposed algorithm exhibits better performance because it can generate a polynomial 44 times faster than existing recurrence algorithms. In addition, the improvement of the proposed algorithm is evaluated in terms of computational cost and reconstruction error. The evaluation involves a comparison with existing recurrence algorithms. Moreover, the maximum size that can be generated using the proposed recurrence algorithm is investigated and compared with those of existing recurrence algorithms. Comparison results indicate that the proposed algorithm exhibits better performance because it can generate a polynomial 44 times faster than existing recurrence algorithms.

ABOUT THE AUTHORS

Alaa M. Abdul-Hadi received his PhD in IT engineering from the national aerospace university (Kharkov Aviation Institute), Ukraine, in 2014. Since 2010 until now, he works as a faculty member in the department of computer engineering, university of Baghdad, Iraq. His research interest includes systems and networks security, trusted computing, and computer networks.

Sadiq H. Abdulhussain received the Ph.D. degree from Universiti Putra Malaysia in 2018. Since 2005, he works as a staff member with the Computer Engineering Department, Faculty of Engineering, University of Baghdad. His research interests include computer vision, signal processing, as well as speech and image processing.

Basheera M. Mahammad received the Ph.D. degree in computer and embedded system engineering from Universiti Putra Malaysia, in 2018. Since 2007, she works as a Staff Member with the Department of Computer Engineering, Faculty of Engineering, University of Baghdad. Her research interests include speech enhancement, signal processing, computer vision, RFID, and cryptography.

PUBLIC INTEREST STATEMENT

Discrete orthogonal polynomials are utilized in different fields such as image processing, speech enhancement, steganography, and others. There are many types of discrete orthogonal polynomials. Discrete Charlier polynomials is well-known and highly utilized in different applications. The existing algorithms have limitations in size when generated discrete Charlier polynomials. In this article, we present a new algorithm for that is able to generate discrete Charlier polynomial with high order and large size. Thus, large signal and image sizes can be analyzed.
algorithm over the traditional recurrence algorithms in terms of maximum-generated size is between 19.25 and 42.85.

**Subjects:** Computer Mathematics; Algorithms & Complexity; Computational Numerical Analysis

**Keywords:** orthogonal polynomials; Charlier polynomials; Charlier moments; recurrence algorithm

1. **Introduction**

Discrete orthogonal polynomials have attracted considerable attention from researchers in numerous scientific fields, particularly speech and image analyses (S. H. Abdulhussain, Ramli et al., 2019; Pee et al., 2017), because of their robustness to noise, do not exhibit redundancy, and do not require a priori information (S. H. Abdulhussain, Ramli, Hussain et al., 2019). Consequently, these polynomials have been extensively used in different applications, such as information hiding (Radeaf et al., 2019), face recognition (Akhmedova & Liao, 2019), edge detection (S. H. Abdulhussain, Ramli, Mahmmod et al., 2019), video content analysis (S. H. Abdulhussain, Rahman Ramli et al., 2019), and speech enhancement (Mahmmod et al., 2017, 2019). In addition, basis functions of orthogonal polynomials can be used as an approximate solution for differential equations (Mizel, 2008). Hu (Ming-Kuei, 1962) introduced the first type of moments, namely, geometric moments. This type of moments are used as a descriptor for images and considered invariant to image manipulation, such as translation, rotation, and scaling. However, geometric moments exhibit information redundancy, and thus, are non-orthogonal, i.e., a signal cannot be reconstructed from these moments.

To address this issue, the concept of continuous orthogonal moments was introduced by Teague (Teague, 1980). For example, Fourier–Mellin (Sheng & Shen, 1994), Legendre (Chong et al., 2004), and Zernike (Khotanzad & Hong, 1990) moments are continuous orthogonal polynomials. Such polynomials improve signal representation by using moments and reduce the effect of noise. However, continuous orthogonal moments involve coordinate transformation and approximation, result in computational load and numerical errors. Accordingly, discrete orthogonal polynomials were introduced in Mukundan (Mukundan et al., 2001) to address the issues of continuous orthogonal polynomials. Hahn (Yap et al., 2007), Krawtchouk (Yap et al., 2003), Tchebichef (Mukundan et al., 2001), Charlier (Zhu et al., 2010) are examples of discrete orthogonal polynomials.

To compute discrete orthogonal moments, discrete orthogonal polynomial coefficients (DOPCs) should be computed accurately.

The computation of DOPCs is computationally expensive and may lead to numerical errors due to the presence of a hypergeometric series and gamma functions. Thus, three-term recurrence (TTR) algorithms are utilized to compute DOPCs. The two types of recurrence relations are generally the TTRs in the \( n \)- and \( x \)-directions. However, different studies (Abdulhussain et al., 2017; S. H. Abdulhussain et al., 2018) have shown that DOPC values result in numerical propagation error when used to compute DOPCs for high orders. Thus, new recurrence algorithms are presented to address the problem of numerical propagation errors for Tchebichef and Krawtchouk polynomials.

Studies performed on Charlier polynomials (CHPs) are divided into 1) recurrence relation algorithms and 2) moment computation algorithms. Research involving recurrence algorithms has been introduced into \( n \)- and \( x \)-direction recurrence algorithms. Both types of recurrence algorithms cannot generate high-order polynomials because of the initial values used and the number of recurrence times performed. The second stream of studies involves the computation of Charlier moments (CHMs) with reduced computational cost. However, these studies are based on either the
n- or $x$-direction recurrence algorithm. To the best of our knowledge, there are no previous study has discussed the ability of the CHP. In addition, the main concerns of researchers are increasing the speed and minimizing the error (Razian & MahvashMohammadi, 2017). Motivated by the aforementioned issues and discussion, the current study aims to present a new recurrence relation to compute CHPs efficiently for high-order polynomials.

In this paper, an empirical study is performed to investigate the maximum order that can be generated using the existing algorithms and analyzing their problems. Due to the importance of the initial value and its effect on the computation polynomial coefficients, a sufficient initial values is proposed. In addition, a reduced form of the recurrence algorithm is presented to reduce the computation time as well as the propagation error. Lastly, we present a new symmetry relation that will reduce the numerical propagation error via the reduction of the recurrence times used.

The rest of this paper is structured as follows. The basic aspects of computing CHPs and the existing TTR algorithms are described in Section 2. The proposed recurrence algorithm is presented in Section 3. An experimental analysis is performed to evaluate the proposed recurrence algorithm in Section 4. Lastly, conclusions are drawn in Section 5.

2. Computation of Charlier polynomials and Charlier moments

CHPs are defined and CHMs are computed in this section. The existing TTR relation is also introduced.

2.1. The definition of CHP

The definition of CHP $C_p^n(x)$ for the $n$th order is obtained from (Zhu et al., 2010) as follows:

$$C_p^n(x) = {}_2F_0 \left( \begin{array}{c} -n, -x \end{array} \bigg| - \frac{1}{p} \right) \quad (1)$$

where $p$ represents the parameter of the CHP, and ${}_2F_0$ is the hypergeometric series which is given by (Koekoek et al., 2010):

$$\sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{k!} z^k \quad (2)$$

where $(a)_k$ represents the ascending factorial, which is also known as the Pochhammer symbol, and defined as follows:

$$(a)_k = a(a+1)(a+2) \cdots (a+b-1) = \frac{(a+b-1)!}{(b-1)!} = \frac{\Gamma(a+b)}{\Gamma(b)} \quad (3)$$

On the basis of Equations (1) and (2), CHP can be expressed as:

$$C_p^n(x) = \sum_{k=0}^{\infty} \frac{(-n)_k (-x)_k}{k!} \left( - \frac{1}{p} \right)^k \quad (4)$$

CHP fulfills the orthogonality conditions such that (Karmouni et al., 2018):

$$\sum_{x=0}^{N} C_p^n(x)C_p^m(x)\omega_C(x;p) = \rho_C(n;p)\delta_{nm} \quad (5)$$

where $\omega_C(x;p)$ and $\rho_C(n;p)$ are the weight function and squared norm of CHP $C_p^n(x)$, respectively. The weight function is defined as follows (Jahid et al., 2019):
\[ \omega_c(x;p) = \frac{e^{-p}p^x}{x!} \]  \hspace{1cm} (6)

and the squared norm \( \rho_c(n;p) \) is expressed as follows (Jahid et al., 2019):

\[ \rho_c(n;p) = \frac{n!}{p^n} \]  \hspace{1cm} (7)

The computation of CHP coefficients (CHPCs) by using Equation (4) produces numerical instability. Therefore, a weighted and normalized CHP (WNCHP) is required (S. H. Abdulhussain et al., 2018). The \( n \)th order of WNCHP is expressed as follows:

\[ \hat{C}_n^p(x) = \sqrt{\frac{\omega_c(x;p)}{\rho_c(n;p)}} C_n^p(x) \]  \hspace{1cm} (8)

### 2.2. Definition of Charlier moments

Moments, also known as transform coefficients, are defined as scalar quantities used to represent signals without redundancy (Abdulhussain et al., 2019; Mahmmod et al., 2018). To represent a 1D signal \( f(x) \) in the moment domain of CHP, CHMs are computed as follows:

\[ \phi_n = \sum_{x=0}^{N-1} C_n^p(x) f(x) \]  \hspace{1cm} (9)

\( n = 0, 1, \ldots, \text{Ord} - 1 \)

where \( \phi_n \) represents the CHMs of the function \( f(x) \), \( \text{Ord} \) is the maximum order used to represent the signal. The signal \( \tilde{f}(x) \) can be reconstructed from CHMs as follows:

\[ \tilde{f}(x) = \sum_{n=0}^{\text{Ord} - 1} \phi_n \hat{C}_n^p(x) \]  \hspace{1cm} (10)

For a 2D signal \( f(x,y) \) with a size of \( N_1 \times N_2 \), 2D CHMs \( \phi_{nm} \) is computed as follows:

\[ \phi_{nm} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x,y) C_n^p(x) C_m^p(y) \]  \hspace{1cm} (11)

\( n = 0, 1, \ldots, \text{Ord}_1 \) and \( m = 0, 1, \ldots, \text{Ord}_2 \)

where \( \text{Ord}_1 \) and \( \text{Ord}_2 \) represent the maximum order utilized for signal characterization. To reconstruct the 2D signal \( \tilde{f}(x,y) \) from CHMs, the process is performed as follows:

\[ \tilde{f}(x,y) = \sum_{n=0}^{\text{Ord}_1} \sum_{m=0}^{\text{Ord}_2} \phi_{nm} \hat{C}_n^p(x) \hat{C}_m^p(y) \]  \hspace{1cm} (12)

Notably, the reconstructed signal is \( \tilde{f} = f \) when all the moments are used in the reconstruction process.

### 2.3. Existing three-term recurrence algorithms

The computation of CHPCs by using Equation (8) is considered computationally expensive because of the utilization of the hypergeometric and gamma functions. Consequently, the TTR algorithm is used to compute CHPCs (Abdulhussain et al., 2017). Existing TTR algorithms are performed in the \( n \)- and \( x \)-directions. These TTR algorithms are presented and analyzed in detail in the following subsections.

#### 2.3.1. Recurrence Algorithm in the \( n \)-direction

The TTR algorithm of WNCHP in the \( n \)-direction is presented as follows (Zhu et al., 2010):
\[ C_{n+1}^p(x) = a_1 C_n^p(x) + a_2 C_{n-1}^p(x) \]  
(13)

where \( n = 1, 2, \ldots, N - 1 \); and \( x = 0, 1, \ldots, N - 1 \), such that

\[ a_1 = \frac{p - x + n}{p} - \sqrt{\frac{p}{n + 1}} \]

\[ a_2 = -\sqrt{\frac{n}{n + 1}} \]  
(14)

and initial values of TTR are

\[ C_0^p(x) = \sqrt{\frac{\omega_C(x; \rho)}{\rho_C(0; \rho)}} \]  
(15)

\[ C_1^p(x) = \frac{p - x}{p} \sqrt{\frac{\omega_C(x; \rho)}{\rho_C(1; \rho)}} \]  
(16)

where \( x = 0, 1, \ldots, N - 1 \). The TTR algorithm in the \( n \)-direction computes the WNCHP coefficients in the \( n \)th order by using the WNCHP values of the \( (n-1) \)th and \( (n-2) \)th orders. For the sake of simplicity the TTR algorithm in the \( n \)-direction is denoted by TTR\( n \).

In practice, this TTR\( n \) algorithm is inapplicable to large signal sizes and high-order polynomials. Figure 1 shows the WNCHP for different size (\( N \)) and parameter (\( p \)) values. The coefficient values and distribution of WNCHP is affected by the value of the CHP parameter (\( p \)). When \( N = 150 \), WNCHP is generated without any numerical instability; however, stable WNCHP coefficients cannot be generated when \( N = 180 \) because of the numerical instabilities and accumulated errors resulting from the nature of the recursive computation of polynomial coefficients (Karmouni et al., 2018). In addition, numerical errors occur in TTR because the initial values are zero (as shown in Figure 1) due to the nature of the formula used. To solve this problem, a TTR algorithm in the \( x \)-direction is proposed.

2.3.2. Recurrence Algorithm in the \( x \)-direction

The coefficient of WNCHP at the \( x \)th index for the \( n \)th order is determined using the previous WNCHP coefficients, \( (x-1) \)th and \( (x-2) \)th. The TTR algorithm of WNCHP in the \( x \)-direction is given as follows (Yamni et al., 2018):

\[ C_n^p(x) = \beta_1 C_{n-1}^p(x - 1) + \beta_2 C_{n-2}^p(x - 2) \]  
(17)

where \( n = 0, 1, \ldots, N - 1 \); and \( x = 2, 3, \ldots, N - 1 \);

\[ \beta_1 = \sqrt{\frac{\omega_C(x)}{\rho_C(0; \rho)}} \frac{x - 1 + p - n}{\sqrt{\omega_C(x - 1)}} \]

\[ \beta_2 = -\sqrt{\frac{\omega_C(x)}{\rho_C(0; \rho)}} \frac{x - 1}{\sqrt{\omega_C(x - 2)}} \]  
(18)

The initial values used in TTR are

\[ C_0^p(0) = \sqrt{\frac{\omega_C(0)}{\rho_C(0)}} \]  
(19)

\[ C_1^p(1) = \frac{p - n}{p} \sqrt{\frac{\omega_C(1)}{\rho_C(1)}} \]  
(20)
where \( n = 0, 1, \ldots, N - 1 \). For simplicity, the TTR algorithm in the \( x \)-direction is denoted by TTRx.

In practice, the TTRx algorithm cannot handle signals with high orders and large sizes. Figure 2 shows the WNCHPs generated using the TTRx algorithm for different size (\( N \)) and parameter (\( p \)) values. When \( N = 150 \), WNCHP is completely generated without numerical errors. However, the TTRx algorithm cannot generate firm WNCHP coefficients when \( N = 180 \) because of numerical instability (see Figure 2(d)), and the zero values of the computed initials (see Figure 2(e) and 2(f)).

3. Proposed recurrence relation

The proposed algorithm for computing CHPCs is presented in this section. In the following subsection, initial value analysis, CHP identity, and the proposed TTR algorithm based on the initial value and CHP identity are discussed. Then, a summary of the proposed algorithm is provided for further clarification. For more clarification, the nomenclatures and the abbreviation used in this paper are listed in Tables 1 and 2, respectively.

3.1. Initial values

The computation of the initial value is essential and affects the other values of CHPCs.

Both TTR algorithms (i.e., TTRn and TTRx) rely on two sets of computed initial values. For example, \( \tilde{C}_p^0(x) \) and \( \tilde{C}_p^1(x) \) are the two initial sets used in TTRn. In practice, computing the initial values by using Equations (15) and (16) for TTRn or Equations (19) and (20) for TTRx is infeasible because of zero and incorrectly computed values (Figures 1(d) and 2(d)). Figure 1(d) shows that the values within the range \( n = 0 \) and \( x = 172, \ldots, 174 \) are generated with zero-value coefficients. In addition, the values within the range \( n = 0 \) and \( x = 175, 176, \ldots, 179 \) are incorrectly generated due to the nature of the formula used. The incorrectly generated coefficient values are indicated as "Not a Number" in several testing environments, such as MATLAB, C++, and Python. In Figure 2(d), the values within the range \( x = 0 \) and \( n = 0, 1, \ldots, N - 1 \). For simplicity, the TTR algorithm in the \( x \)-direction is denoted by TTRx.
Table 1. Table of abbreviation

| Abbreviation | Description |
|--------------|-------------|
| CHP          | Charlier polynomials |
| DOPC         | Discrete orthogonal polynomial coefficients |
| TTR          | Three-term recurrence |
| CHM          | Charlier moments |
| WNCHP        | Weighted and normalized Charlier polynomials |
| TTRn         | Three-term recurrence in the x-direction |
| TTRx         | Three-term recurrence in the n-direction |
| CHPC         | Charlier polynomial coefficient |
| GSOP         | Gram-Schmidt orthonormalization process |
| ETIP         | Execution time improvement percentage |
| ETSR         | Execution time speedup ratio |
| DCHT         | Discrete Charlier transform |
| NMSE         | Discrete Charlier transform |
| NRE          | Normalized restriction error |

Table 2. List of nomenclatures

| Symbol       | Description |
|--------------|-------------|
| $c^p_n(x)$   | Charlier Polynomials |
| $c^p_n(x)$   | weighted and normalized Charlier Polynomials |
| $p$          | parameter of the Charlier polynomial |
| $\text{Ord}$ | Order of the polynomial |
| $f(x)$       | one dimensional signal |
| $f(x,y)$     | two dimensional signal |
| $\bar{f}(x)$| one dimensional reconstructed signal |
| $\bar{f}(x,y)$| two dimensional reconstructed signal |
| $R$          | Weighted and normalized Charlier polynomial matrix |
| $N$          | Polynomial size |
| $n$          | order index of Charlier polynomial |
| $x$          | size index of Charlier polynomial |
| $\Gamma()$   | Gamma function |
| $\omega(x)$  | Charlier polynomial weight function |
| $\rho_n(n)$  | Charlier polynomial norm |
| $\text{Time}_P$ | Execution time of the proposed polynomial |
| $\text{Time}_E$ | Execution time of the existed polynomial |
| $r$          | covariance coefficients |
| $\mathcal{M}$| covariance matrix |
| $\sigma^2$   | variance of the transform coefficients |
| $^T$         | transpose operation of a matrix |
| $J_m$        | normalized restriction error |
| $E_n$        | normalized mean square error |
| [ ]          | floor function |
| $M$          | moments |
| $i$          | image |
| $I_r$        | reconstructed image |
172, 173, … 179 are generated with zero-value coefficients. For further clarification, Equation (15) can be simplified as follows:

\[
C_p(x) = \sqrt{\frac{e^{-p}p^x}{x!}}
\]  

(21)

From Equation (21), the factorial swiftly increases as the values of index \(x\) increase. In addition, the term \(p^x\) increases as \(x\) increases. In practice, these terms are computed incorrectly. Note that, the same previous problem of TTRn is taken place for TTRx.

To overcome this problem, one initial value can be computed and a two-term recurrence can be used. This process can be performed by first computing \(C_p^0(0)\) and then estimating the values of \(C_p^n(0)\) recursively within the range \(n = 1, 2, \ldots, N - 1\) as follows:

\[
C_p^n(0) = \sqrt{\frac{p^n}{n!}}C_p^{n-1}(0)
\]  

(22)

In practice, however, the initial value \(C_p^0(0)\) cannot be computed for the parameter value \(p = 1480\). Thus, the best initial value that can be used to start computing CHPCs is a coefficient with a recognizable value. To achieve a recognizable initial value and considering that all the CHPCs rely on \(C_p^n(0)\), the coefficient of \(C_p^n(0)\) is inspected. Figure 3 shows the values of \(C_p^n(0)\) for different values of parameter \(p\) when \(N = 300\).

In 3, the peak value of \(C_p^n(0)\) for different parameter \(p\) values is located at \(n = p - 1\) and \(n = p\). For example, the peak of \(C_p^{29}(0) = C_p^{30}(0)\) is located at \(n = 29\) and \(n = 30\); the peak of \(C_p^{249}(0) = C_p^{250}(0)\) is located at \(n = 249\) and \(n = 250\). Accordingly, the current study proposes that the initial value should be computed at \(n = p - 1\) and \(n = p\). However, the values of \(C_{p-1}^p(0)\) and \(C_p^p(0)\) are equal at
Nevertheless, this initial value cannot be estimated for large values of parameter $p$ due to exponentiation and factorial functions. The key solution to this problem is using the logarithmic function as follows:

$$\hat{C}_n^p(0) = \sqrt[n]{e^{-p} p^{n-1} \frac{p}{n!}} = \sqrt[n]{e^{-p} p^{n-1} \frac{p}{n!}}$$

Figure 4 presents the result of $\hat{C}_n^p(0)$ for a wide range of parameter $p$ values. The maximum value of $p$ using Equation (23) to compute $\hat{C}_n^p(0)$ is 140, and (25) can compute $\hat{C}_n^p(0)$ for a wide range of parameter $p$ values.
To compute the remaining CHPCs for $\mathcal{C}_p^n(0)$, a two-term recurrence relation can be obtained and used. For the range $n = p - 2, p - 3, \ldots, 0$, the two-term recurrence relation is expressed as follows:

$$\mathcal{C}_p^{n-1}(0) = \sqrt{\frac{e^{\frac{p}{n(n+1)}x}}{T_{n+1}}} \mathcal{C}_p^n(0)$$

$$\mathcal{C}_p^{n-1}(0) = \sqrt{\frac{p}{n}} \mathcal{C}_p^n(0)$$ \quad (26)

For the range $n = p + 1, p + 2, \ldots, N - 1$ is presented as follows:

$$\mathcal{C}_p^{n+1}(0) = \sqrt{\frac{p}{n}} \mathcal{C}_p^n(0)$$ \quad (27)

The two-term recurrence relation can be used to compute the CHPCs of $\mathcal{C}_p^n(1)$ based on $\mathcal{C}_p^n(0)$ as follows:

$$\mathcal{C}_p^{n+1}(1) = \sqrt{\frac{p}{n}} \frac{(p-n)}{\sqrt[2]{\prod_{k=n-1}^{n} n}} \mathcal{C}_p^n(0)$$ \quad (28)

$$n = 0, 1, \ldots, N - 1$$

### 3.2. CHP identity and recurrence relation

In this section, we present the WNCHP identity that will reduce the numerical error of CHPCs based on TTR. This identity is used to compute CHPCs for one portion and then find CHPCs for the other portion. This identity is symmetric to the primary diagonal ($n = x$).

The following Pochhammer identities are used to prove the WNCHP identity (Spanier & Oldham, 1987):

$$(-a)_b = (-1)^b (a - b + 1)_b$$ \quad (29)

$$(a)_b = b! \left( \begin{array}{c} a + b - 1 \\ b \end{array} \right)$$ \quad (30)

Equations (29) and (30) can obtain the following identity:

$$(-a)_b = (-1)^b b! \left( \begin{array}{c} a \\ b \end{array} \right)$$ \quad (31)

By expanding Equation (8), the following form of WNCHP $\mathcal{C}_p^n(x)$ is derived:

$$\mathcal{C}_p^n(x) = \sum_{k=0}^{\infty} \left( \frac{(-n)_k (-x)_k}{k!} \left( -\frac{1}{p} \right)^k \right) \sqrt{\frac{e^{-\frac{p}{x} x^n}}{x^n}}$$ \quad (32)

By using Equations (31) and (32) can be rewritten as follows:

$$\mathcal{C}_p^n(x) = \sum_{k=0}^{\infty} \left( \frac{n}{k} \right) \left( \frac{x}{k} \right) k! \left( -\frac{1}{p} \right)^k \sqrt{\frac{e^{-\frac{p}{x} x^n}}{x^n}}$$ \quad (33)
For the similarity with respect to \((n = x)\) of the WNCHP plane, substituting \(n\) with \(x\) in Equation (33) yields:

\[
\hat{C}_p^x(n) = \sum_{k=0}^{\infty} \binom{n}{k} \binom{x}{k} \frac{1}{k!} \left[ \frac{e^{-p^r} p^n}{n! x^k} \right]^{\frac{1}{2}}
\]

(34)

From Equations (33) and (34), a conclusion can be drawn that

\[
\hat{C}_p^x(x) = \hat{C}_p^x(n)
\]

(35)

\(n = 0, 1, \ldots, N - 1\), and \(x = 0, 1, \ldots, N - 1\)

Thus, the polynomial space can be divided into two portions (P1 and P2), as shown in Figure 5. Through Equation (35), CHPCs in P1 are computed using TTR, and those in P2 are computed using the symmetry relation in Equation (35).

### 3.3. The TTR algorithm

From the presented WNCHP identity and the computed initial values, CHPCs in P1 are computed using the modified TTRx algorithm, as follows:

\[
\hat{C}_p^x(x + 1) = \gamma_1 \hat{C}_p^x(x) + \gamma_2 \hat{C}_p^x(x - 1)
\]

(36)

where

\(x = 1, 2, \ldots, N - 1\) and \(n = x, x + 1, \ldots, N - 1\)

\[
\gamma_1 = \frac{a - n + x}{\sqrt{\alpha(x + 1)}}
\]

(37)

\[
\gamma_2 = -\frac{x}{\sqrt{(x + 1)}}
\]

(38)

Figure 5: \(n - x\) plane of CHP divided into two parts (P1 and P2).
3.4. Summary of the proposed algorithm
This section summarizes the steps of the proposed TTR algorithm to further understand the computation of WNCHP. The steps are summarized as follows.

Step 1- The value of \( C_p^0(0) \) is computed using Equation (25).

Step 2- The value of \( C_p^n(0) \) are computed as follows.

A The values of \( C_p^n(0) \) within the range \( n = 0, 1, \ldots, p - 1 \) are computed using (26).

B Equation (27) is used to compute the values of \( C_p^n(0) \) within the range \( n = p + 1, p + 2, \ldots, N - 1 \).

Step 3- The values of \( C_p^0(1) \) is computed using Equation (28) within the range \( n = 0, 1, \ldots, N - 1 \).

Step 4- Equation (36) is used to compute CHPCs within the ranges \( x = 2, 3, \ldots, N - 1 \) and \( n = x, x + 1, \ldots, N - 1 \).

Step 5- Lastly, CHPCs within the ranges \( x = 2, 3, \ldots, N - 1 \) and \( n = 0, 1, \ldots, x - 1 \) are defined in (35) by using the similarity identity.

The aforementioned steps are illustrated in Figure 6.

4. Experimental analysis
This section presents the performance evaluation of the proposed method in terms of computational cost, energy compaction, signal reconstruction ability, and maximum size of the generated polynomial.

4.1. Analysis of computational cost
The performance of the proposed algorithm in terms of computational cost in obtaining CHPCs should be demonstrated. The computational cost of the proposed algorithm is compared with those of existing algorithms, namely, TTRn and TTRx. In addition, a recent algorithm is included in
the computation cost analysis. The recent recurrence algorithm is based on Gram-Schmidt orthonormalization process (GSOP) (Daoui et al., 2020).

The computation time experiment is performed on TTRn, TTRx, and the proposed algorithm using different CHP sizes and parameter $p = N/2$. Figure 7 shows the average computation time results for 10 runs. The time required to generate CHPCs using the TTRn and TTRx algorithms is comparable. For example, the time required to generate CHPCs using the TTRn and TTRx algorithms at $N = 2000$ is $-$0.619 s and $-$0.607 s, respectively. In addition, the time required to generate CHPCs remarkably increases polynomial size increases when using the TTRn and TTRx algorithms because of 1) the number of computed coefficients, i.e., $N \times N$, and 2) the mathematical expression used in the computation of CHPCs. However, the computation time exhibits higher reduction than those of existing algorithms. For example, at $N = 2000$, the computation time is $-$0.014 s, i.e., the execution time improvement percentage (ETIP) is $-97.7%$. Notably, ETIP is computed using the following formula (Hosny, 2008):

$$ETIP = 1 - \frac{Time_P}{Time_E}$$

(39)

where $Time_P$ and $Time_E$ are the computation times of the proposed and existing TTR algorithms, respectively. To further illustrate the advantage of the proposed algorithm, Table 3 lists the ETIP values of the proposed polynomial compared with those of existing recurrence algorithms (TTRn and TTRx). The proposed algorithm reduces the time required to generate CHP by $-97.7%$. To further explain the obtained results, it is well-known that the expensive operation that influence the computation cost of the recurrence algorithm are multiplication, division, and recurrence times. Due to the employment of the identity relation in the proposed algorithm, only 50% of the coefficients are computed. Accordingly, 50% of the expensive operations have been excluded. For instance, at $N = 1000$, there are 10,000 coefficients should be computed using the existing TTR algorithms. Each coefficient is computed by performing the TTR relation and each recurrence relation has parameters need to be computed. These parameters involve multiplication and division operation, which increase the execution time. On the other hand, for the proposed algorithm, only 5000 coefficients are computed, i.e., 50% of recurrence relations are excluded. Thus, the proposed algorithm reduces the number of operations required and it is considered effective in terms of computational complexity for a wide range of CHP sizes.

To emphasis the ability of the proposed algorithm in terms of the computational cost, the execution time speedup ratio (ETSR) is calculated between the proposed algorithm and TTRn, TTRx, and GSOP. Table 4 lists the ETSR between the proposed algorithm and existing algorithms. The ETSR is computed as follows:

$$ETSR = \frac{Time_E}{Time_P}$$

(40)

Figure 7. (a) Computation times for TTRn, TTRx, and the proposed algorithm, and (b) Computation times for GSOP and the proposed algorithm.
It can be inferred from Table 4 that the proposed algorithm outperforms the existing recurrence algorithms in terms of speedup ratio. For instance, when $N$ is 500, the average ETSR is 44x over TTRn and TTRx. The obtained speedup ratio is due to the reduction of the recurrence times. In addition, the average speedup ratio of the proposed algorithm over the GSOP is 17906x. The GSOP algorithm is considered computationally overload because of the utilization of the nested loops in its implementation.

4.2. Analysis of polynomial order and energy compaction

The distributions of signal moments typically differ in each transform (Abdulhussain et al., 2019). In addition, the arrangement of moment sequence indices $n$ is important in retrieving significant signal information. Thus, the moment energy distribution for discrete Charlier transform (DCHT) should be known prior to the analysis of signal reconstruction. Such analysis can be performed by following the mathematical procedure presented in (Jain, 1989; Zhu et al., 2010), which is based on a stationary Markov sequence with zero mean and length $N$. Suppose that the matrix $\mathcal{M}$ is the covariance matrix with different covariance coefficients ($r$). This matrix is given as follows (Jain, 1989):

$$\mathcal{M} = \begin{bmatrix}
1 & r & r^2 & \ldots & r^{N-1} \\
r & 1 & \ldots & \ldots & \vdots \\
r^2 & \vdots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
r^{N-1} & \ldots & \ldots & \ldots & 1
\end{bmatrix} \tag{41}$$

After transforming the covariance matrix ($\mathcal{M}$) into the Charlier transform domain, the diagonal of the transformed matrix $S$ is used to represent the variance $\sigma^2_l$ of the transform coefficients. To obtain matrix $S$, the following function is applied:

$$S = R \mathcal{M} R^T \tag{42}$$

where $R$ refers to the WNCHP matrix, and $(\cdot)^T$ represents the transpose operation of a matrix. Different values of parameter $p$ and the coefficient covariance $r$ are used in this experiment. The results of the experiment are listed in Table 5 by using $p = N/4$, $N/2$, $\ldots$, and $3N/4$ and $r = 0.85$, and 0.95.

The maximum value of the variance ($\sigma^2_l$) of DCHT is located at index $l = 0$, and the variance decreases as the index value $l$ increases. Consequently, the DCHT moment order, which can be used to reconstruct signal information, is given as follows: $n = 0, 1, 2, \ldots, N - 1$.

Energy compaction is considered an important property in checking the performance of discrete transform. It is used to evaluate the tendency of CHP to transform a large fraction of signal energy.

| Table 3. Comparison of ETIP values among the proposed polynomial, TTRn (ETIPn), and TTRx (ETIPx) |
|---|
| $N$ | 500 | 1000 | 1500 | 2000 | 2500 | 3000 | 3500 | 4000 | 4500 | 5000 | Average |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ETIPn | 97.95 | 97.75 | 97.68 | 97.75 | 97.62 | 97.75 | 97.71 | 97.76 | 97.63 | 97.67 | 97.73 |
| ETIPx | 97.95 | 97.72 | 97.65 | 97.70 | 97.62 | 97.73 | 97.72 | 97.72 | 97.60 | 97.64 | 97.70 |

| Table 4. Comparison of speedup ratio between the proposed polynomial and existing algorithms (TTRn (ETSRn), TTRx (ETSRx), and GSOP (ETSRG)) |
|---|
| $N$ | 500 | 1000 | 1500 | 2000 | 2500 | 3000 | 3500 | 4000 | 4500 | 5000 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ETSRn | 48 | 44 | 43 | 44 | 42 | 44 | 44 | 43 | 44 | 42 |
| ETSRx | 48 | 43 | 42 | 43 | 42 | 44 | 43 | 43 | 41 | 42 |
| ETSRG | 1344 | 3604 | 8310 | 12,190 | 16,533 | 21,928 | 24,612 | 28,296 | 29,500 | 32,737 |
into relatively few coefficients of moments. To check the effect of parameter value \( p \) on the energy compaction performance of CHP, the normalized restriction error (NRE) (Jain, 1989) is utilized as follows (Jain, 1989):

\[
J_m = \frac{\sum_{q=0}^{N-1} \sigma_q^2}{\sum_{q=0}^{N-1} \sigma_q^2} \tag{43}
\]

where \( J_m \) is the NRE, and \( \sigma_q^2 \) is the arrangement of \( \sigma_q^2 \) in descending order. NRE is computed using different values of parameter \( p \) and covariance coefficients \( r \).

Figure 8 presents NRE \( (J_m) \) as a function of the number of retained samples \( q \). Two values of covariance coefficients \( r \) are used. EC is influenced by the CHP parameter \( p \). For example, when parameter \( p \) increases from \( N/6 \) to \( N/3 \), the performance of CHP in terms of energy compaction increases because the NRE of \( p = N/3 \) reaches the minimum value before \( p = N/6 \). When the value of \( p \) increases to more than \( N/3 \), NRE is smaller than that of \( p = N/3 \). Therefore, CHP at \( p = N/3 \) uses less moments than the other values of \( p \) to reconstruct a signal efficiently.

### 4.3. Analysis of reconstruction error and maximum size

In this section, experiments are performed to evaluate the proposed algorithm against existing algorithms in terms of reconstruction error and maximum size can be generated. Different image sizes and CHP parameter values are used in the experiment. A cameraman image is selected as the test image for this experiment. The test image is resized several times to obtain \( I \). For each image size, WNCHP \( R \) is generated using TTRn, TTRx, and the proposed algorithm. Then, the image is transformed into the moment domain \( M \). Thereafter, the image is transformed into the spatial domain \( I_r \). Lastly, the normalized mean square error (NMSE) between the input and reconstructed images is computed as follows (S. H. Abdulhussain et al., 2018):

| \( l \) | \( p = N/4 \) | \( p = N/2 \) | \( p = 3N/4 \) | \( p = N/4 \) | \( p = N/2 \) | \( p = 3N/4 \) |
|-------|------------|------------|-------------|------------|------------|-------------|
| 0     | 6.179      | 7.295      | 5.722       | 6.179      | 7.295      | 5.722       |
| 1     | 2.553      | 2.791      | 2.313       | 2.553      | 2.791      | 2.313       |
| 2     | 2.441      | 1.335      | 1.618       | 2.441      | 1.335      | 1.618       |
| 3     | 1.558      | 0.971      | 1.059       | 1.558      | 0.971      | 1.059       |
| 4     | 0.898      | 0.697      | 1.015       | 0.898      | 0.697      | 1.015       |
| 5     | 0.310      | 0.402      | 0.666       | 0.310      | 0.402      | 0.666       |
| 6     | 0.252      | 0.401      | 0.562       | 0.252      | 0.401      | 0.562       |
| 7     | 0.274      | 0.222      | 0.395       | 0.274      | 0.222      | 0.395       |
| 8     | 0.127      | 0.217      | 0.361       | 0.127      | 0.217      | 0.361       |
| 9     | 0.168      | 0.128      | 0.277       | 0.168      | 0.128      | 0.277       |
| 10    | 0.104      | 0.149      | 0.198       | 0.104      | 0.149      | 0.198       |
| 11    | 0.116      | 0.094      | 0.178       | 0.116      | 0.094      | 0.178       |
| 12    | 0.082      | 0.099      | 0.120       | 0.082      | 0.099      | 0.120       |
| 13    | 0.094      | 0.084      | 0.121       | 0.094      | 0.084      | 0.121       |
| 14    | 0.062      | 0.067      | 0.101       | 0.062      | 0.067      | 0.101       |
| 15    | 0.082      | 0.081      | 0.072       | 0.082      | 0.081      | 0.072       |
where $E_n$ represents the NMSE between images $I$ and $I_r$. The image is resized three times ($N = 144, 160,$ and $256$), and the three values of the CHP parameter are ($p = [N/3], [N/2],$ and $[2N/3]$). The experiment results of NMSE for TTRn, TTRx, and the proposed algorithm are presented in Figure 9.

As shown in Figure 9, the image can be fully reconstructed using TTRn, TTRx, and the proposed algorithm for image size $N \times N = 144 \times 144$ and all the values of parameter $p$. However, for image size $N \times N = 160 \times 160$, the image can be fully reconstructed using TTRn for parameter values $[N/3]$ and $[N/2]$ but fails to recover any image information for any order when the parameter value is $[2N/3]$. In addition, the reconstruction of the image by using TTRx can be fully performed for parameter values $[N/3]$ and $[2N/3]$, and the reconstructed image exhibits an error when the parameter value is $[N/2]$ for a moment value greater than 40 because of the zero values of the polynomial coefficients for high-order polynomials (see Figure 2). For image size $N \times N = 256 \times 256$, image reconstruction by using TTRx fails to reconstruct the image for any parameter value due to the propagation error in the computation of CHP. Meanwhile, TTRn successfully reconstructs the image only for CHP parameter values $[N/3]$ and $[2N/3]$ but fails to reconstruct any image information for the parameter value of $[N/2]$. By contrast, the proposed algorithm can efficiently reconstruct the image for all the image sizes and CHP parameter values. The maximum size that can be generated is further investigated using the proposed polynomial.
To find the maximum size that can generate CHP without propagation error for each algorithm, the following steps are performed for each algorithm:

1. Small values are set accordingly for polynomial order $N$ and parameter $p$.
2. The cameraman image $I$ is resized to $N \times N$.
3. CHP $R$ with an order of $N$ is generated.
4. Moments $M = R \times I \times R^{T}$ are computed.
5. Image $I_r = R^{T} \times M \times R$ is reconstructed.
6. The NMSE between $I$ and $I_r$ is found.
7. If the NMSE less than $10^{-3}$, then the size is increased; otherwise, the previous value of $N$ is considered the maximum size that the algorithm can generate. Note for small image size the threshold is set to $18 \times 10^{-3}$.

Table 6 lists the results obtained from the aforementioned procedure. The results include the maximum size for TTRn, TTRx, and the proposed algorithms corresponding to different values of the CHP parameter $p$. The results show that the proposed algorithm is better than existing algorithms in terms of the size that can be generated for CHP with different values of parameter $p$. Improvement in the proposed algorithm and the best performance in existing algorithms (TTRn and TTRx) are also reported to for better illustration. The improvement values show that the minimum improvement is 19.25 at $p = 5N/6$, which increases to 42.85 at $p = N/2$. 

Figure 9. NMSE of image reconstruction for (a) $N = 144$, and $p = \lfloor N/3 \rfloor$, (b) $N = 144$, and $p = \lfloor N/2 \rfloor$, (c) $N = 160$, and $p = \lfloor N/3 \rfloor$, (d) $N = 160$, and $p = \lfloor N/2 \rfloor$, (e) $N = 160$, and $p = 2N/3$, (f) $N = 256$, and $p = \lfloor N/3 \rfloor$, (g) $N = 256$, and $p = \lfloor N/2 \rfloor$, and (h) $N = 256$, and $p = 2N/3$. 

Abdul-Hadi et al., Cogent Engineering (2020), 7: 1763553
https://doi.org/10.1080/23311916.2020.1763553
5. Conclusion
This study introduces a new identity of WNCHP and the reduced form to compute the initial values of WNCHP. Then, they are used to present a new recurrence algorithm for the efficient computation of WNCHP with high order. The analyses of the recurrence relation and numerical instability of CHP are explained. The identity relation of CHP to the main diagonal is beneficial for reducing the number of recurrence times. The CHP plane is divided into two portions, CHPCs are computed for one portion, and then the identity relation is utilized to find the CHPCs of the second portion. The analysis clearly indicates that the proposed recurrence algorithm has significantly less computation load than existing recurrence algorithms. In addition to its ability to reduce numerical propagation errors, the proposed algorithm also accelerates the computational speed of CHPCs. Although the proposed recurrence algorithm achieves remarkable results, its performance can still be improved to generate higher-order polynomials by examining the causes of the zero values in the coefficients, particularly for parameter \( p \) values that are less than \( N/2 \).

Funding
The authors received no direct funding for this research.

Author details
Alaa M. Abdul-Hadi
E-mail: alaa.m.abdulhadi@coeng.uobaghdad.edu.iq
ORCID ID: http://orcid.org/0000-0002-3682-2243
Sadiq H. Abdulhussain
E-mail: sadiq76@yahoo.com
ORCID ID: http://orcid.org/0000-0002-6439-0082
Basheera M. Mahmmod
E-mail: basheera.m@coeng.uobaghdad.edu.iq
ORCID ID: http://orcid.org/0000-0002-4121-0843
1 Department of Computer Engineering, University of Baghdad, Al-Jadriya, Baghdad 10071, Iraq.

Cover Image
Source: Author.

Citation information
Cite this article as: On the computational aspects of Charlier polynomials, Alaa M. Abdul-Hadi, Sadiq H. Abdulhussain & Basheera M. Mahmmod, Cogent Engineering (2020), 7: 1763553.

References
Abdulhussain, S. H., Ramli, A. R., Al-Haddad, S. A. R., Mahmmod, B. M., & Jassim, W. A. (2017). On computational aspects of tchebichef polynomials for higher polynomial order. IEEE Access, 5(1), 2470–2478. [Online]. http://ieeexplore.ieee.org/document/7856982/
Abdulhussain, S. H., Ramli, A. R., Al-Haddad, S. A. R., Mahmmod, B. M., & Jassim, W. A. (2018, March). Fast recursive computation of krawtchouk polynomials. Journal of Mathematical Imaging and Vision, 60(3), 285–303. [Online]. https://doi.org/10.1007/s10851-017-0758-9
Abdulhussain, S. H., Ramli, A. R., Hussain, A. J., Mahmmod, B. M., & Jassim, W. A. (2019). Orthogonal polynomial embedded image kernel. Proceedings of the international conference on information and communication technology - ICICT '19, ser. ICICT ’19 (pp. 215–221). New York, USA: ACM Press. [Online]. Available: https://doi.org/10.1145/3321289.3321310
Abdulhussain, S. H., Ramli, A. R., Mahmmod, B. M., Al-Haddad, S. A. R., & Jassim, W. A. (2019). Image edge detection operators based on orthogonal polynomials. International Journal of Image and Data Fusion, 8(3), 293–308. doi:10.1080/19479832.2017.1326405
Abdulhussain, S. H., Ramli, A. R., Mahmmod, B. M., Saripan, M. I., Al-Haddad, S., & Jassim, W. A. (2019a, November). A new hybrid form of krawtchouk and tchebichef polynomials: Design and application. Journal of Mathematical Imaging and Vision, 61(4), 1–8. Budapest, Hungary, Hungary: IEEE. [Online]. https://ieeexplore.ieee.org/document/8851750/
Abdulhussain, S. H., Ramli, A. R., Al-Haddad, S. A. R., Mahmmod, B. M., & Jassim, W. A. (2019). An efficient hybrid form of krawtchouk and tchebichef polynomials: Design and application. Journal of Mathematical Imaging and Vision, 61(4), 1–8. Budapest, Hungary, Hungary: IEEE. [Online]. https://ieeexplore.ieee.org/document/8851750/

| \( p \)  | \( TTRn \) | \( TTRx \) | Proposed | Improvement |
|------|------|------|--------|-----------|
| \( N/6 \) | 80 | 76 | 1540 | 19.25 |
| \( N/5 \) | 92 | 90 | 1830 | 19.89 |
| \( N/4 \) | 116 | 116 | 2330 | 20.09 |
| \( N/3 \) | 168 | 162 | 3430 | 20.42 |
| \( N/2 \) | 162 | 172 | 7370 | 42.85 |
| \( 2N/3 \) | 154 | 172 | 5520 | 32.09 |
| \( 3N/4 \) | 150 | 172 | 4910 | 28.55 |
| \( 4N/5 \) | 148 | 172 | 4600 | 26.74 |
| \( 5N/6 \) | 148 | 172 | 4420 | 25.70 |
Abdulhussain, S. H., Ramlı, A. R., Mahmmod, B. M., Saripan, M. I., Al-Haddad, S. A. R., & Jassim, W. A. (2019b, July). Shot boundary detection based on orthogonal polynomial. Multimedia Tools and Applications, 78(14), 20 361–20 382. [Online]. http://link.springer.com/10.1007/s11404-019-7364-3

Akhemdova, F., & Liao, S. (2019). Face recognition with discrete orthogonal moments. In Mahmoud Hassaballah and Khalid M. Hosny (eds.), Recent advances in computer vision: Theories and Applications. Springer (pp. 189–209). [Online]. http://link.springer.com/10.1007/978-3-030-03000-1_8

Chong, C.-W., Raveendran, P., & Mukundan, R. (2004, January). Translation and scale invariants of Legendre moments. Pattern Recognition, 37(1), 119–129. [Online]. https://www.elsevier.com/science/article/abs/pii/S0031320303002516

Dauoi, A., Yamni, M., El Ogni, O., Karmouni, H., Sayyouri, M., & Qjidaa, H. (2020, February). Stable computation of higher order charlier moments for signal and image reconstruction. Information Sciences, 521, 251–276. [Online]. https://doi.org/10.1016/j.ins.2020.02.019

Hart, J. F. (1978). Computer approximations. Krieger Publishing Co., Inc.

Hosny, K. M. (2008, March). Fast computation of accurate Zernike moments. Journal of Real-Time Image Processing, 3(1–2), 97–107. [Online]. http://link.springer.com/10.1007/s11554-007-0058-5

Jahid, T., Karmouni, H., Hmimid, A., Sayyouri, M., & Qjidaa, H. (2019, may). Fast computation of Charlier moments and its inverses using Clenshaw’s recurrence formula for image analysis. Multimedia Tools and Applications, 78(19), 12 183–12 201. [Online]. http://link.springer.com/10.1007/s11554-018-7657-z

Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc.

Karmouni, H., Hmimid, A., Jahid, T., Sayyouri, M., Qjidaa, H., & Rezzouk, A. (2018, September). Fast and stable computation of the charlier moments and their inverses using digital filters and image block representation. Circuits, Systems, and Signal Processing, 37(9), 4015–4033. [Online]. https://doi.org/10.1007/s00034-018-0755-2

Khotanzad, A., & Hong, Y. (1990, may). Invariant image recognition by Zernike moments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5), 489–497. [Online]. http://ieeexplore.ieee.org/document/55109/

Koekoek, R., Lesky, P. A., & Swarttouw, R. F. (2010). Hypergeometric orthogonal polynomials and their q-analogues. Springer Science & Business Media.

Mahmmod, B. M., Ramlı, A. R., Abdulhussain, S. H., Al-Haddad, S. A. R., & Jassim, W. A. (2019a). Low-distortion MMSE speech enhancement estimator based on laplacian prior. IEEE Access, 5(1), 9866–9881. [Online]. http://ieeexplore.ieee.org/document/7914629/

Mahmmod, B. M., Ramlı, A. R., Boker, T., Al-Obeidat, F., Abdulhussain, S. H., & Jassim, W. A. (2019). Speech enhancement algorithm based on super-gaussian modeling and orthogonal polynomials. IEEE Access, 7, 103 485–103 504. [Online]. https://ieeexplore.ieee.org/document/8766088/

Ming-Kuei, H. (1962, February). Visual pattern recognition by moment invariants. IEEE Transactions on Information Theory, 8(2), 179–187. [Online]. http://ieeexplore.ieee.org/document/1057692/

Mitzel, A. K. E. (2008). Orthogonal functions solving linear functional differential equations using chebyshev polynomial. Baghdad Science Journal, 5(1), 143–148. [Online]. https://doi.org/10.21123/bsj.5.1.143-148

Mukundan, R., Ong, S., & Lee, P. (2001). Image analysis by Tchebichef moments. IEEE Transactions on Image Processing, 10(9), 1357–1364. [Online]. http://ieeexplore.ieee.org/document/941859/

Pee, C.-Y., Ong, S., & Raveendran, P. (2017, June). Numerically efficient algorithms for anisotropic scale and translation Tchebichef moment invariants. Pattern Recognition Letters, 92, 68–74. [Online]. https://www.sciencedirect.com/science/article/abs/pii/S0167865517301294

Radeef, H. S., Mahmmod, B. M., Abdulhussain, S. H., & Al-Jumaieel, D. (2019). A stegeanography based on orthogonal moments, in Proceedings of the international conference on information and communication technology – ICT ’19, ser. ICT ’19 (pp. 147–153). New York, USA: ACM Press. [Online]. https://doi.org/10.1145/3321289.3321324

Razian, S. A., & MahvashMohammadi, H. (2017). Optimizing raytracing algorithm using cuda. Emerging Science Journal, 1(3), 167–178. [DOI]. http://dx.doi.org/10.28991/jse-01119

Sheng, Y., & Shen, L. (1994, June). Orthogonal Fourier–Mellin moments for invariant pattern pattern recognition. Journal of the Optical Society of America A, 11(6), 1748. [Online]. https://www.osapublishing.org/abstract.cfm?URI=josaa-11-6-1748

Spanier, B. C. J., & Oldham, K. B. (1987). An atlas of functions. Hemisphere publishing corporation.

Teague, M. R. (1980). Image analysis via the general theory of moments. JOSA, 70(8), 920–930. [https]. doi.org/10.1364/JOSA.70.000920

Yamni, M., Dauoi, A., Ori, O. E., Karmouni, H., Sayyouri, M., & Qjidaa, H. (2018). Influence of Krawtchouk and Charlier moment’s parameters on image reconstruction and classification. Procedia Computer Science, 148(Cds), 418–427. [Online]. https://linkinghub.elsevier.com/retrieve/pii/S1877050919300559

Yap, P. T., Paramesran, R., & Ong, S. H. (2003, November). Image analysis by krawtchouk moments. IEEE Transactions on Image Processing, 12(11), 1367–1377. [Online]. http://ieeexplore.ieee.org/document/1240103/

Yap, P.-T., Paramesran, R., & Ong, S.-H. (2007, November). Image analysis using hahn moments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(11), 2057–2062. [Online]. http://ieeexplore.ieee.org/document/9302768/

Zhu, H., Liu, M., Shu, H., Zhang, H., & Luo, L. (2010). General form for obtaining discrete orthogonal moments. IET Image Processing, 4(5), 335. [Online]. http://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2009.0195
