



LJMU Research Online

Mahmmod, BM, Abdulhussain, SH, Naser, MA, Alsabah, M, Hussain, A and Al-Jumeily, D

3D Object Recognition Using Fast Overlapped Block Processing Technique

<https://researchonline.ljmu.ac.uk/id/eprint/18800/>

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

**Mahmmod, BM ORCID logoORCID: <https://orcid.org/0000-0002-4121-0843>,
Abdulhussain, SH ORCID logoORCID: <https://orcid.org/0000-0002-6439-0082>, Naser, MA, Alsabah, M, Hussain, A ORCID logoORCID:
<https://orcid.org/0000-0001-8413-0045> and Al-Jumeily, D ORCID logoORCID:**

LJMU has developed **LJMU Research Online** for users to access the research output of the University more effectively. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LJMU Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain.

The version presented here may differ from the published version or from the version of the record. Please see the repository URL above for details on accessing the published version and note that access may require a subscription.

For more information please contact researchonline@ljmu.ac.uk

<http://researchonline.ljmu.ac.uk/>

Article

3D Object Recognition Using Fast Overlapped Block Processing Technique

Basheera M. Mahmmod ¹, Sadiq H. Abdulhussain ¹, Marwah Abdulrazzaq Naser ², Muntadher Alsabah ², Abir Hussain ^{3,4,*} and Dhiya Al-Jumeily ^{3,*}

¹ Department of Computer Engineering, University of Baghdad, Al-Jadriya, Baghdad 10071, Iraq

² Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 4ET, UK

³ School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK

⁴ Department of Electrical Engineering, University of Sharjah, Sharjah 27272, United Arab Emirates

* Correspondence: a.hussain@ljamu.ac.uk or abir.hussain@sharjah.ac.ae (A.H.); d.aljumeily@ljamu.ac.uk (D.A.-J.)

Abstract: Three-dimensional (3D) image and medical image processing, which are considered big data analysis, have attracted significant attention during the last few years. To this end, efficient 3D object recognition techniques could be beneficial to such image and medical image processing. However, to date, most of the proposed methods for 3D object recognition experience major challenges in terms of high computational complexity. This is attributed to the fact that the computational complexity and execution time are increased when the dimensions of the object are increased, which is the case in 3D object recognition. Therefore, finding an efficient method for obtaining high recognition accuracy with low computational complexity is essential. To this end, this paper presents an efficient method for 3D object recognition with low computational complexity. Specifically, the proposed method uses a fast overlapped technique, which deals with higher-order polynomials and high-dimensional objects. The fast overlapped block-processing algorithm reduces the computational complexity of feature extraction. This paper also exploits Charlier polynomials and their moments along with support vector machine (SVM). The evaluation of the presented method is carried out using a well-known dataset, the McGill benchmark dataset. Besides, comparisons are performed with existing 3D object recognition methods. The results show that the proposed 3D object recognition approach achieves high recognition rates under different noisy environments. Furthermore, the results show that the presented method has the potential to mitigate noise distortion and outperforms existing methods in terms of computation time under noise-free and different noisy environments.

Keywords: 3D recognition; overlapped block processing; features extraction; orthogonal moments; orthogonal polynomials; SVM; Charlier polynomials



Citation: Mahmmod, B.M.; Abdulhussain, S.H.; Naser, M.A.; Alsabah M.; Hussain, A.; Al-Jumeily, D. 3D Object Recognition Using Fast Overlapped Block Processing Technique. *Sensors* **2022**, *22*, 9209. <https://doi.org/10.3390/s22239209>

Academic Editor: Petros Daras

Received: 30 October 2022

Accepted: 24 November 2022

Published: 26 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Significant effort has been dedicated to developing efficient and reliable remote health-care systems with the Internet of Things (IoT) applications [1]. This development can be achieved through transmitting efficient and secure medical images and videos of the patients and processing them in a fast and reliable way. To this end, advanced remote monitoring schemes of the patients become essential. In particular, efficient three-dimensional (3D) object recognition techniques could be beneficial to process the images and videos of medical systems. This is due to the ability of object recognition to enable feature extraction, which is essential as it provides unique characteristics that can identify objects [2]. Besides, object recognition is also considered as having the most significant importance in the industrial environment, as it represents each object individually and can distinguish the object [3]. These cues are used to extract discriminative features for accurate recognition [4]. Therefore, there is an increasing interest in object recognition, especially in the fields of machine vision, pattern recognition, and machine learning applications [5–7]. Various

domains, including facial identification [8], gender description [9], and gesture analysis, among others, use object recognition. Object recognition is also used in object identification, medical diagnosis, security applications, multimedia communication, and computer interface applications [4,10].

1.1. Related Works

Object recognition and classification can be considered essential techniques, which are beneficial in various applications such as healthcare systems, pattern recognition, molecular biology, and computer vision [11–15]. To this end, significant research works have been developed for efficient 3D object recognition. Besides, feature extraction for 3D objects is extremely useful for classification [16]. Extensive researches have been carried out to develop 3D object classification methods. Some of these works are based on the principles of moment invariants and 3D moments. To this end, a method of 3D translation, rotation, and scale invariants (TRSI) was developed in [17] from geometric moments and an alternative approach was presented later in [18]. A tensor approach to derive the rotation invariants from the geometric moments was proposed in [19]. Besides, an automatic algorithm was proposed in [20,21] to generate 3D rotation invariants from geometric moments. Recently, a 3D Hahn moments combined with convolutional neural networks (CNN) was proposed in [22] to enhance the 3D object classification. Specifically, the work in [22] proposed a hybrid approach based on combining the 3D discrete Hahn moments and CNN to improve 3D object classification. A multi-layer artificial neural network (ANN) perception approach was proposed in [23] for the classification and recognition of 3D images. In [24], a deep learning approach based on neural network and Racah-based moments was proposed for 3D shape classification. Additionally, in [16], an approach based on the combination of 3D discrete orthogonal moments and deep neural network (DNN) algorithms was proposed to improve the classification accuracy of the 3D object features. In [25], a 3D discrete Krawtchouk moments method was proposed for content-based search and retrieval applications. In [26], a 3D image analysis was considered using Krawtchouk and Tchebichef polynomials, where orthogonal moments were exploited to characterize various types of 2D and 3D images. To this end, orthogonal moments are used in many applications such as image analysis [27,28], face recognition [29], pattern recognition [30,31], steganography [32], image reconstruction [33,34], and medical image analysis [35,36].

The recognition process depends extremely on the feature extraction process, which is used to distinguish between different objects. To this end, the process of object localization and object normalization is considered essential for feature extraction technique [37]. As such, essential issues in object recognition and computer vision applications are the extraction of significant features from objects [38]. Object recognition to date is still a challenging problem that affects pattern recognition. This is because the accuracy of object recognition can be affected by class variations [4,39,40]. In particular, different methods are utilized to extract the features from the images. These methods can be classified as deep-learning-based methods, orthogonal-moment-based methods, and texture-based methods [41–45]. While the recognition accuracy of deep-learning-based methods can be very high, these methods run into a substantial amount of computational complexity, as explained in [46–48]. In the orthogonal moment approaches, the features of the object are calculated efficiently through the use of Orthogonal Polynomials (OPs) techniques [49]. Due to their effectiveness, orthogonal moments (OMs) and OPs have been widely exploited in recent years for pattern recognition, form descriptors, and image analysis [50,51]. The OMs-based method gives a powerful capability for evaluating the image components because the image components can be efficiently represented in the transform domain [49]. In many object recognition applications, OMs can be utilized to extract features. It is possible to consider the OMs as a scalar approach that is utilized to define and characterize a function. Such OMs can be used to achieve an effective extraction of the features. The OPs function also contains the coordinates of an image in addition to OMs [52,53]. According to work performed in [44], OMs can be exploited in feature extraction from images with various geometric

invariants, including translation, scaling, and rotation. In general, various types of moments can be used for image processing. For instance, due to their simplicity, geometric moments are favored above other types of moments [54]. To depict an image with the least amount of redundancy possible, a Zernike and pseudo-Zernike moments approach was developed in [55]. In [55], a moments-based approach was proposed by exploiting the fractional quaternion for colored image detection [56]. This is because the fractional quaternion, which is considered an opposed approach to integer-order polynomials, can represent functions, according to [44]. Furthermore, the diagnosis of plant diseases has been accomplished using fractional-order moments [57]. In [58], the image analysis used Zernike and Legendre polynomials, which act as the kernel functions for Zernike and Legendre moments, respectively. In particular, the Zernike moments approach has the property of invariance in addition to its capability of image data storing and processing with the least amount of redundancy. However, because the Zernike moments approach focused only on the continuous domain, such an approach would require image coordinate adjustments and transformations for discrete domain [59,60]. To address the challenge of computing continuous moments in image analysis, the discrete OM approach has recently been proposed [61]. To this end, Mukundan presented a series of moments in [62] that uses discrete Tchebichef polynomials to analyze the image.

Typically, the extraction approaches are divided into global and local features. The former is also called a holistic-based approach [63], which can capture the essential characteristics of the full human face image. At the same time, it is known as the component-based approach or block-processing-based approach, from specific areas in images [64]. In the global feature-based approach, various imaging setups are used to achieve improved performance for feature extraction [65]. To this end, several feature extraction techniques have been proposed so far to enable a global feature-based approach [66,67]. In block processing or what is known as the local feature-based approach, the image features can be extracted locally by utilizing OM, which entails processing the image's blocks after it has been divided into several blocks to ease their processing. In this approach, the signals such as images and videos can be divided efficiently into several blocks so that they transfer to another domain to extract the features [68]. The signal characteristics can be stored locally in memory to prepare it for the next step of processing. The work in [63] demonstrates that the (local) block-processing-based approach achieves better performance in feature extraction compared with the (global) holistic-based approach.

One technique for extracting local features is the local binary patterns method [69–71]. In addition, the combination of global- and local-based approaches, which is termed the hybrid features extraction-based approach, aims to achieve the highest object recognition accuracy [72,73]. It is demonstrated that block processing, which represents local feature extraction, can achieve the highest recognition accuracy with the trade-off of higher computation complexity. Specifically, compared with global features, local features are thought to be more reliable and improve recognition accuracy, see, e.g., [74–76]. To this end, partitioning the images using image block processing has the potential of extracting the blocks of any image and analyzing them sequentially. From the perspective of computer memory, this operation is not sequential, which is seen as a major flaw in performance and a crucial difference between the memory and the speed of the CPU. While such an operation would result in additional cache misses and replacements, accessing the complete matrix sequentially can aid in maintaining spatial locality [68]. The removal of additional procedures will speed up the extraction of local features. Specifically, extracting local features from the image blocks using discrete transform will decrease the computational complexity, which is called a fast overlapped block-processing method for feature extraction [68]. Although several advanced methods have been proposed for object recognition, the accuracy and running time are to date considered challenging issues that need to be addressed. Therefore, finding a quick and accurate mechanism for 3D object detection is necessary. Additionally, most of the exciting works need to account for the impact of undesirable noise on recognition. Hence, there is a limited understanding of the effect of noisy environments. Therefore,

investigating the proposed method in the noise condition is significant to characterize the effectiveness of the feature extraction for object recognition processes.

1.2. Paper Contributions

To overcome the aforementioned challenges, a robust object recognition algorithm that exploits Charlier polynomials and their moments is proposed. The proposed algorithm has a powerful capability for the characterization and feature extraction of the signals of the 3D objects effectively. In addition, to extract the features effectively and in a fast manner, this paper exploits an overlap block-processing technique to provide a construction of auxiliary matrices, which essentially extends the original signal to prevent the time delay in the loops computation. Furthermore, the proposed method is evaluated in the noise condition to characterize the effectiveness of the proposed method in feature extraction for object recognition processes. The major contributions of this paper can be summarized as follows: (1) Proposing an advanced design for robust 3D object recognition, which takes into account the accuracy, computational complexity, and execution time. (2) Exploiting the powerful Charlier polynomials to extract the features of the 3D objects. (3) Developing a fast overlapped block-processing algorithm, which shows more accurate processing for the blocks of the image to perform fast feature extraction with low complexity. The proposed overlapped block-processing method is mainly used to decrease the computation time. (4) Finally, implementing the support vector machine (SVM) to classify object recognition features accurately. To this end, a well-known dataset known as the McGill benchmark dataset is used for performance evaluation [77]. The results demonstrate that the proposed method achieves high recognition accuracy with lower computational complexity. Furthermore, the results demonstrate that the proposed method is able to reduce noise distortion and outperforms traditional methods under both clean and noisier environments. These achievements signify the importance of the proposed method for the future implementation of 3D object recognition.

1.3. Paper Organization

The paper is organized as follows. In Section 2, the orthogonal polynomials and their moments are introduced. In Section 3, the methodology of the proposed method for feature extraction and recognition of 3D objects is presented. In Section 4, the performance evaluation of the proposed method and the numerical results are discussed. Finally, the conclusion of the paper is presented.

2. Preliminaries of OPs and Their OMs

The mathematical definition of the utilized OPs is explained in this section. Additionally, this section also describes the computation of the OMs for the 3D signals.

2.1. Charlier Polynomials Computation and Their Moments

This subsection discusses the Charlier polynomials and their moments. In addition, the existing three-term recurrence (TTR) relation is described. Several studies have considered the use of Charlier polynomials due to its accuracy and effectiveness [78]. To this end, research on the application of Charlier polynomials has been divided into two main areas: moment computation algorithms and recurrence relation algorithms. For the recurrence relation-based algorithms, the research works exploit the n -direction and x -direction of the matrix. However, generating high-order polynomials is not possible in these recurrence algorithms. This is due to the use of the initial values and the number of recurrence times. The research works make use of either the x -direction or the n -direction of the recurrence algorithm as their calculation algorithms. To the best of our knowledge, no research studies have looked into using Charlier polynomials and their moments for 3D object detection. This paper investigates the effect of using Charlier polynomials for 3D object recognition. This paper also aims to provide an efficient method for achieving a recurrence relation to compute Charlier polynomials for high-order polynomials.

In what follows, the Charlier polynomials and their moments computation are presented.

2.2. Computation of Charlier Polynomials

Charlier polynomials $C_n(y; p)$ of d th dimension can be calculated as follows:

$$C_n(x; p) = {}_2F_0 \left(\begin{matrix} -n, -x \\ - \end{matrix} \middle| -\frac{1}{p} \right), \quad (1)$$

$$n, x = 0, 1, 2, \dots, N-2, N-1; p > 0,$$

where p denotes the parameter of the Charlier polynomials, and ${}_2F_0$ represents the mathematical formulation of the hypergeometric series, which is expressed as [79]

$${}_2F_0 \left(\begin{matrix} a_1, a_2 \\ - \end{matrix} \middle| z \right) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k}{k!} (z)^k, \quad (2)$$

where $(a)_k$ denotes the ascending factorial, which is termed as the Pochhammer symbol [79].

Following the expressions provided by Equations (1) and (2), Charlier polynomials can be written as

$$C_n(x; p) = \sum_{k=0}^{\infty} \frac{(-n)_k (-x)_k}{k!} \left(-\frac{1}{p} \right)^k. \quad (3)$$

It is worth noting that the orthogonality condition should be met with Charlier polynomials. Besides, the weighted function can be applied to the Charlier polynomials so that

$$\sum_{x=0}^D C_n(x; p) C_m(x; p) \omega_C(x; p) = \rho_C(n; p) \delta_{nm}, \quad (4)$$

where $D = N - 1$, $\omega_C(x; p)$, which denotes the weighted function and $\rho_C(d; p)$ represents the squared norm of Charlier polynomials dx . The weighted function and the squared norm of Charlier polynomials are provided in expressions (5) and (6), respectively.

$$\omega_C(x; p) = \frac{e^{-p} p^x}{x!}. \quad (5)$$

$$\rho_C(n; p) = \frac{n!}{p^n}. \quad (6)$$

It is worth noting that the calculation of the Charlier polynomials' coefficients provided by the expression in Equation (3) may cause numerical instability. Hence, to overcome this issue, a weighted normalized Charlier polynomial is applied. To this end, the n th order of weighted normalized Charlier polynomials can be expressed as

$$\hat{C}_n(x; p) = \sqrt{\frac{\omega_C(x; p)}{\rho_C(n; p)}} C_n(x; p). \quad (7)$$

2.3. Computation of Charlier Moments

This subsection discusses the computation of Charlier moments. The Charlier moments, denoted as transform coefficients, are scalar quantities utilized to demonstrate signals without redundancy [49,80]. For a one-dimensional (1D) signal, denoted as $f(x)$, Charlier moments can be computed in the moment domain as

$$\mu_n = \sum_{x=0}^{N-1} \hat{C}_n(x; p) f(x) \quad (8)$$

$$n = 0, 1, \dots, Ord - 1,$$

where μ_n denotes the Charlier moments and Ord represents the maximum number of orders utilized for signal representation. To obtain the signal $\tilde{f}(x)$ from the Charlier domain (moments domain), inverse transform can be utilized as follows:

$$\tilde{f}(x) = \sum_{n=0}^{Ord-1} \mu_n \hat{C}_n(x; p) \quad (9)$$

For a two-dimensional (2D) signal $f(x, y)$ of size $N \times M$, the Charlier moments with 2D signal, denoted as μ_{nm} , can be computed as

$$\begin{aligned} \mu_{nm} &= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \hat{C}_n(x; p) \hat{C}_m(y; p) \\ n &= 0, 1, \dots, Ord_1 \text{ and} \\ m &= 0, 1, \dots, Ord_2, \end{aligned} \quad (10)$$

where the parameters Ord_1 and Ord_2 denote the highest order used for the representation of the signal. To reconstruct the 2D signal $\tilde{f}(x, y)$, denoted as $\tilde{f} = f$, from the Charlier domain, the following inverse transformation is used:

$$\tilde{f}(x, y) = \sum_{n=0}^{Ord_1} \sum_{m=0}^{Ord_2} \mu_{nm} \hat{C}_n(x; p) \hat{C}_m(y; p) \quad (11)$$

To compute the moments for higher dimensional space, in our case, the 3D signal, $f(x, y, z)$, the following formula is used:

$$\begin{aligned} \mu_{nml} &= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} f(x, y, z) \hat{C}_n(x; p) \hat{C}_m(y; p) \hat{C}_l(z; p) \\ n &= 0, 1, \dots, Ord_1, \\ m &= 0, 1, \dots, Ord_2 \text{ and} \\ l &= 0, 1, \dots, Ord_3. \end{aligned} \quad (12)$$

2.4. Charlier Coefficients Computation Using Recurrence Relation Algorithm

This section presents the algorithm exploited to compute the coefficients of Charlier polynomials. It is worth noting that the algorithm used in this paper is the recurrence relation, which has been presented in [78].

The computation of initial values of Charlier polynomials' coefficients is essential for obtaining an efficient and reliable recurrence relation algorithm. It should be noted that both three-term recurrence relations algorithms in the x -direction and the n -direction depend on two sets of initial values. To this end, $\hat{C}_0(x; p)$ and $\hat{C}_1(x; p)$ are the two initial values used in the three-term recurrence relation algorithm in the x -direction. In general, calculating the set of initial values is mathematically intractable. This is attributed to incorrectly computed values. To address this issue, a logarithmic function is used [78]:

$$\hat{C}_p(0; p) = \left(e^{-(\log \Gamma(p) + p + (1-p)\log(p))} \right)^{0.5}, \quad (13)$$

where $\log \Gamma$ denotes the logarithmic mathematical operation for the gamma function.

To obtain the remaining Charlier polynomials' coefficients for $\hat{C}_n(0; p)$, the two-term recurrence relation is utilized. We have two ranges for $x = 1$, which are $n < p - 1$ and $n > p$. For the range $n < p - 1$, $n = p - 2, p - 3, \dots, 0$, the two-term recurrence relation can be written as

$$\hat{C}_{n-1}(0; p) = \sqrt{\frac{n}{p}} \hat{C}_n(0; p). \quad (14)$$

For the range $n > p, n = p + 1, p + 2, \dots, N - 1$, the following expression is used:

$$\hat{C}_n(0; p) = \sqrt{\frac{p}{n}} \hat{C}_{n-1}(0; p) \quad (15)$$

After computing the coefficients for $x = 0$, they are used to compute the Charlier polynomials' coefficients for $x = 1$ using the following recurrence relation:

$$\begin{aligned} \hat{C}_n(1; p) &= \frac{(p - n)}{\sqrt{n}} \hat{C}_n(0; p) \\ n &= 0, 1, \dots, N - 1. \end{aligned} \quad (16)$$

To this end, the polynomial space of the Charlier polynomials is divided into two portions: lower triangle and upper triangle [78]. These portions are known as “Part 1” and “Part 2”, which are shown in Figure 1. Charlier polynomials' coefficients in the lower triangle matrix (“Part 1”) are obtained using three three-term recurrence relations. In addition, Charlier polynomials' coefficients in the upper triangle matrix (“Part 2”) are obtained using the symmetry relation provided in the expression given by

$$\begin{aligned} \hat{C}_n(x; p) &= \hat{C}_x(n; p) \\ n &= 0, 1, \dots, N - 1, \text{ and} \\ x &= 0, 1, \dots, n - 1. \end{aligned} \quad (17)$$

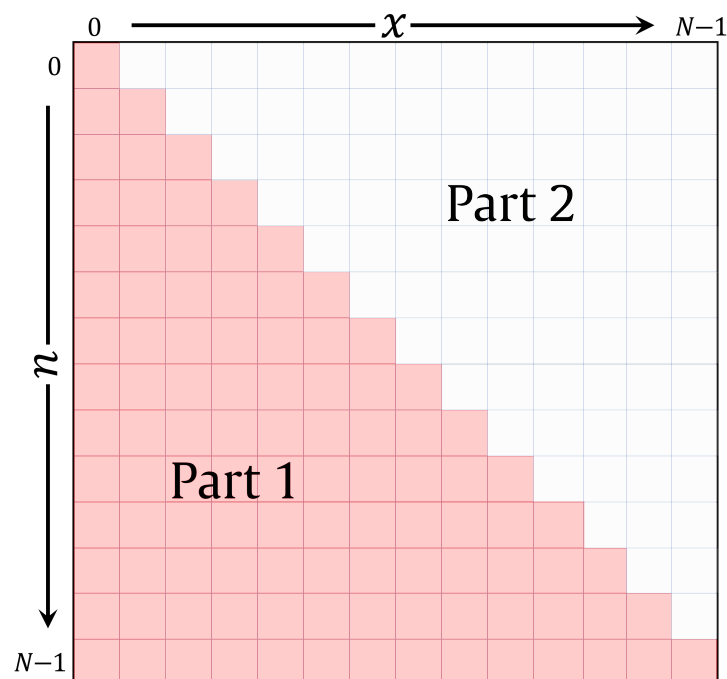


Figure 1. An illustration of the plan dividing into $n - x$ parts for Charlier polynomials.

After the weighted normalized Charlier polynomials' identity and initial values calculations are presented, the calculation of the Charlier polynomials' coefficients in “Part 1” is performed by exploiting the three-term recurrence in the x -direction. To this end, the following calculations are

$$\hat{C}_n(x + 1; p) = A \hat{C}_n(x; p) + B \hat{C}_n(x - 1; p), \quad (18)$$

where $x = 1, 2, \dots, N - 1$ and $n = x, x + 1, \dots, N - 1$; the parameters A and B are obtained, respectively, as [78]

$$A = \frac{a - n + x}{\sqrt{a(x + 1)}}, \quad (19)$$

$$B = -\sqrt{\frac{x}{(x + 1)}}. \quad (20)$$

For more clarification, the utilized algorithm for the weighted normalized Charlier polynomials is summarized in Algorithm 1.

Algorithm 1 The algorithm used to compute the Charlier polynomials

Input: N = Polynomial size, p = Polynomial parameter.

Output: \hat{C} = Charlier polynomials.

```

1: Initialize  $\hat{C}$  with a size of  $N \times N$ 
2:  $\hat{C}_p(0; p) \leftarrow \left( e^{-(\log \Gamma(p) + p + (1-p)\log(p))} \right)^{0.5}$  {Compute initial value.}
3:  $n \leftarrow [p - 2, p - 1, \dots, 0]$  {Set the range of  $n$ .}
4: for  $i$  in range  $n$  do
5:    $\hat{C}_{i-1}(0; p) \leftarrow \sqrt{\frac{i}{p}} \hat{C}_i(0; p)$ 
6: end for
7:  $n \leftarrow [n = p + 1, p + 2, \dots, N - 1]$  {Set the range of  $n$ .}
8: for  $i$  in range  $n$  do
9:    $\hat{C}_i(0; p) \leftarrow \sqrt{\frac{p}{n}} \hat{C}_{i-1}(0; p)$ 
10: end for
11: for  $n = 0$  to  $N - 1$  do
12:    $\hat{C}_n(1; p) \leftarrow \frac{(p-n)}{\sqrt{n}} \hat{C}_n(0; p)$ 
13: end for
14: {Compute the coefficients in “Part 1”}
15: for  $x = 1$  to  $N - 1$  do
16:   for  $n = x$  to  $N - 1$  do
17:      $A \leftarrow \frac{a-n+x}{\sqrt{a(x+1)}}$ 
18:      $B \leftarrow -\sqrt{\frac{x}{(x+1)}}$ 
19:      $\hat{C}_n(x + 1; p) \leftarrow A \hat{C}_n(x; p) + B \hat{C}_n(x - 1; p)$ 
20:   end for
21: end for
22: {Compute the coefficients in “Part 2”}
23: for  $n = 1$  to  $N - 1$  do
24:   for  $x = 0$  to  $n - 1$  do
25:      $\hat{C}_n(x; p) \leftarrow \hat{C}_x(n; p)$ 
26:   end for
27: end for
28:
29: return  $\hat{C}$  {Note that  $H_d$  in Equation (27) is equal to  $\hat{C}$ .}

```

3. Methodology of the Proposed Feature Extraction and Recognition Method of 3D Object

This section presents the feature extraction and recognition processes for the presented 3D object recognition algorithm.

For any recognition system, a feature extraction process is employed to represent signals. As a result, local feature extraction can be used to enable more effective object recognition systems rather than global feature extraction due to their effectiveness, as discussed earlier in the introduction. Therefore, the 3D image might be separated into blocks to increase recognition accuracy. Each block has a size of $B_x \times B_y \times B_z$. The Charlier

polynomials are generated using the procedures in Section 2.4, where Charlier polynomials can be generated with parameter p . The brief methodology of the presented 3D recognition algorithm is shown in Figure 2. First, the 3D image information is obtained. Then, the Charlier polynomials are generated with parameter p . Next, the overlapped polynomials are generated to reduce the computation cost. After that, the fast 3D moments' computation is used to transform the 3D images into the moment domain. Finally, the features are normalized and used to train the SVM model for recognition.

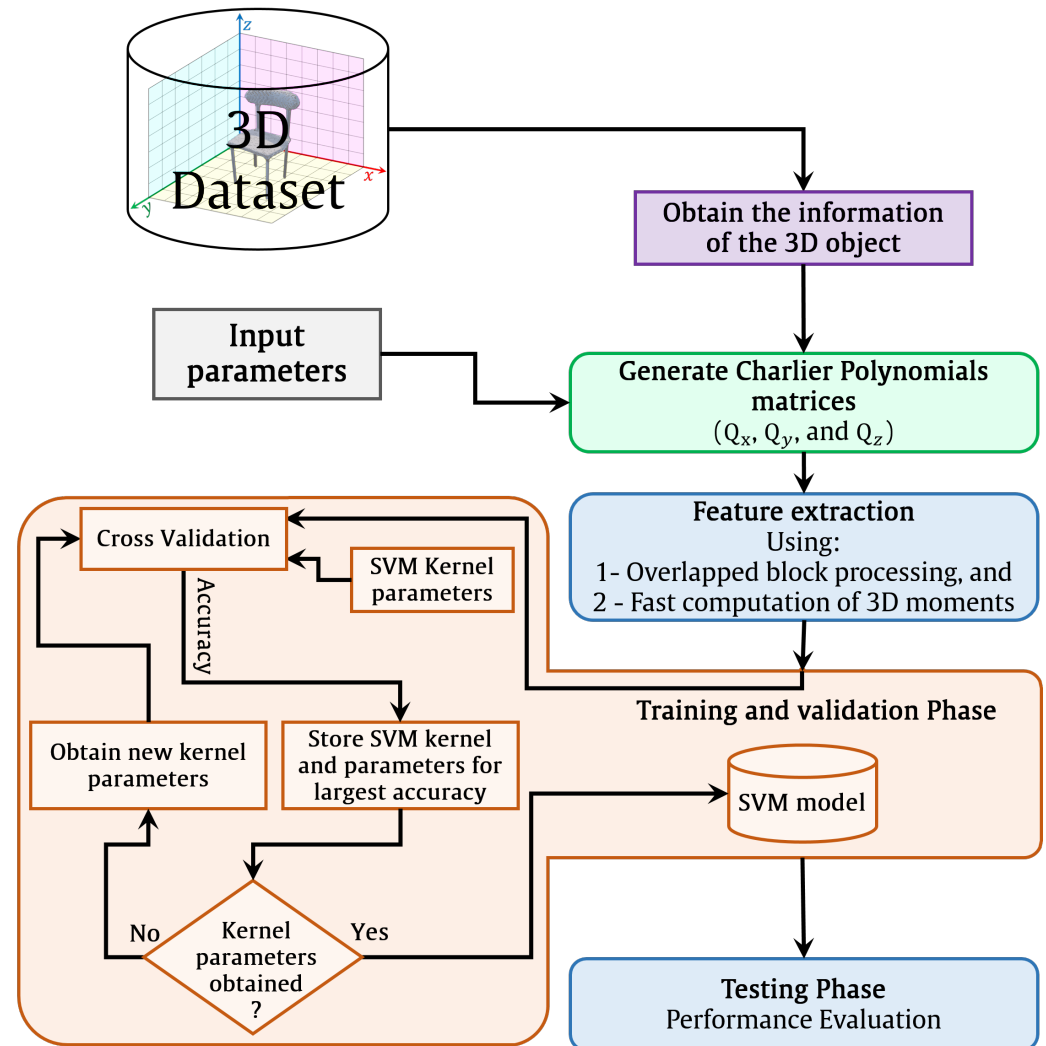


Figure 2. The brief methodology of the presented 3D recognition algorithm.

The global-based feature extraction approach is, to some extent, inaccurate for noisy environments, which highly impedes the characterization of efficient 3D object algorithms in more realistic settings. Moreover, the performance of 3D object recognition accuracy may be degraded in noisy environments [45,81]. Therefore, preprocessing for the 3D object becomes essential to mitigate the effect of noise but it may come at the expense of increasing the computation complexity. The extraction of local features leads to a high computation cost because the traditional method is used, which is considered a bottleneck for real-time application [82]. The local features are extracted after partitioning the 3D object into sub-blocks. For more clarification, see Figure 3.

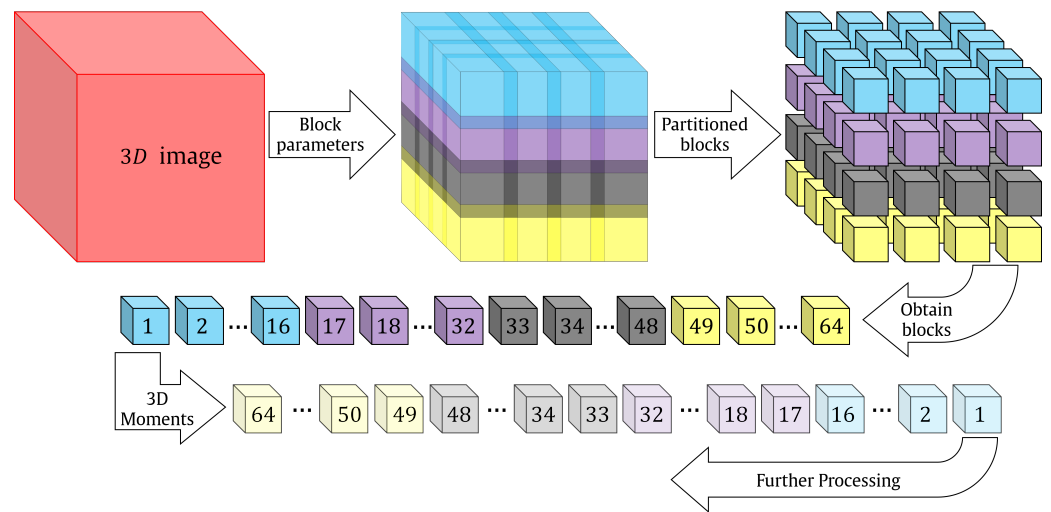


Figure 3. Flow diagram of the traditional block-processing algorithms.

A fast overlapped block-processing technique is exploited to overcome the above challenges. To extract local features, most applications use a non-overlapped block-processing technique. On the other hand, overlapped block processing could enhance the accuracy of 3D object recognition [45,81]. Typically, the processing of the blocks in parallel will significantly raise the cost of computing. We solved this issue by using the fast overlapped block processing described in [68]. The fundamental idea behind fast overlapped block processing (FOBP) is to extend the image by adding auxiliary matrices, which does away with the requirement for nested loops. The computing cost of the feature extraction procedure will be drastically reduced by eliminating the nested loops (see Figure 4). Suppose a 3D image F with a size of $N_x \times N_y \times N_z$ needs to be partitioned into overlapped blocks. The size of the blocks are $B_x \times B_y \times B_z$ with overlapping sizes of v_x , v_y , and v_z in the x , y , and z -direction, respectively. This lead to a total blocks (T_{Blocks}) of

$$\begin{aligned} T_{Blocks} &= Blocks_x \times Blocks_y \times Blocks_z, \\ T_{Blocks} &= \frac{N_x}{B_x - 2v_x} \times \frac{N_y}{B_y - 2v_y} \times \frac{N_z}{B_z - 2v_z}. \end{aligned} \quad (21)$$

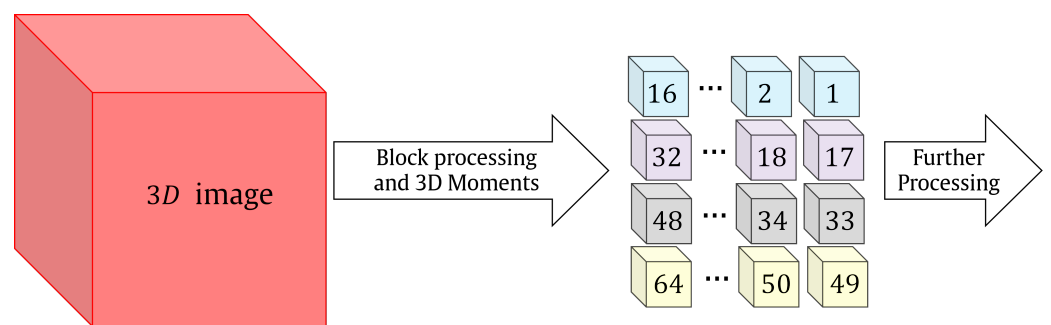


Figure 4. Flow diagram of the proposed fast block-processing algorithm.

For further details about the expressions above, see Figure 5.

Suppose the matrix G represents the extended version of F and can be computed as follows [68]:

$$G = \sum_z E_z [E_y F E_x^T], \quad (22)$$

where E_x , E_y , and E_z are rectangular matrices with sizes of $(B_x \cdot Blocks_x \times N_x)$, $(B_y \cdot Blocks_y \times N_y)$, and $(B_z \cdot Blocks_z \times N_z)$, respectively. For further elucidation, the matrix E_d is shown in Figure 6, where d represents the dimensions (x , y , and z).

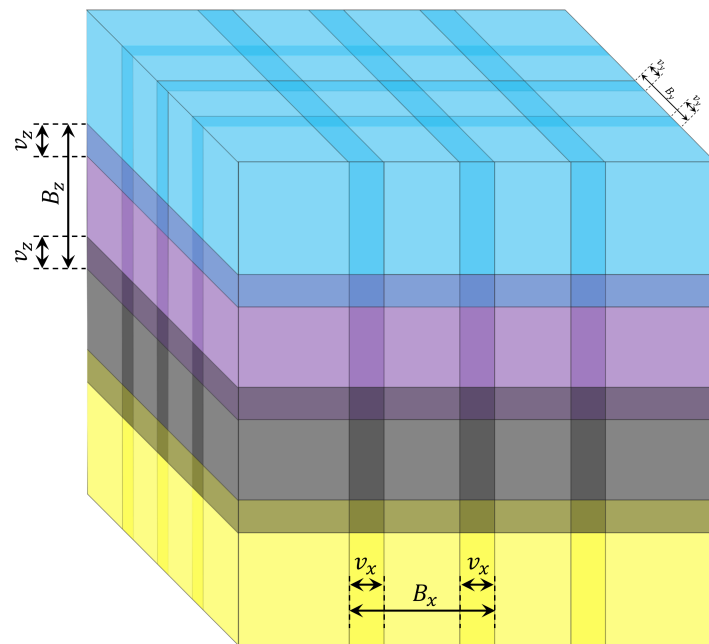


Figure 5. Details of the 3D image with overlapped blocks.

$$\begin{bmatrix}
 \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\
 \vdots & \vdots \\
 \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}
 \end{bmatrix}$$

Figure 6. Details of E_d matrix. The size of E_d matrix is $B_d \cdot \text{Blocks}_d \times N_d$. The size of the identity matrix is $B_d \times B_d$.

To compute the moments (\mathbf{M}) for a 3D image using matrix multiplication, Equation (12) can be rewritten as follows [83]:

$$\mathbf{M} = \sum_z \mathbf{R}_z \left[\mathbf{R}_y \mathbf{G} \mathbf{R}_x^T \right] \quad (23)$$

By substituting Equation (22) in Equation (23), we obtain

$$\mathbf{M} = \sum_z \mathbf{R}_z \left[\mathbf{R}_y \left[\sum_z \mathbf{E}_z \left[\mathbf{E}_y \mathbf{F} \mathbf{E}_x^T \right] \right] \mathbf{R}_x^T \right]. \quad (24)$$

Note that \mathbf{M} represents the matrix form of the moments μ_{nml} .

By following the proof presented in [68], Equation (24) can be rewritten as follows:

$$\mathbf{M} = \sum_z \mathbf{Q}_z \left[\mathbf{Q}_y \mathbf{G} \mathbf{Q}_x^T \right], \quad (25)$$

where \mathbf{Q}_d are computed as follows:

$$\mathbf{Q}_d = \mathbf{R}_d \mathbf{E}_d, \quad (26)$$

where the matrix \mathbf{R}_d can be obtained as follows:

$$\mathbf{R}_d = \mathbb{I} \otimes \mathbf{H}_d, \quad (27)$$

where matrix \mathbb{I} denotes an identity matrix, \otimes denotes the Kronecker product, and H_d represents the Charlier polynomials. Note that d represents the dimensions x , y , or z . Due to the matrices independence from the image, they are generated, stored, and repeatedly used [45,68]. The process for the matrices generation is depicted in Figure 7.

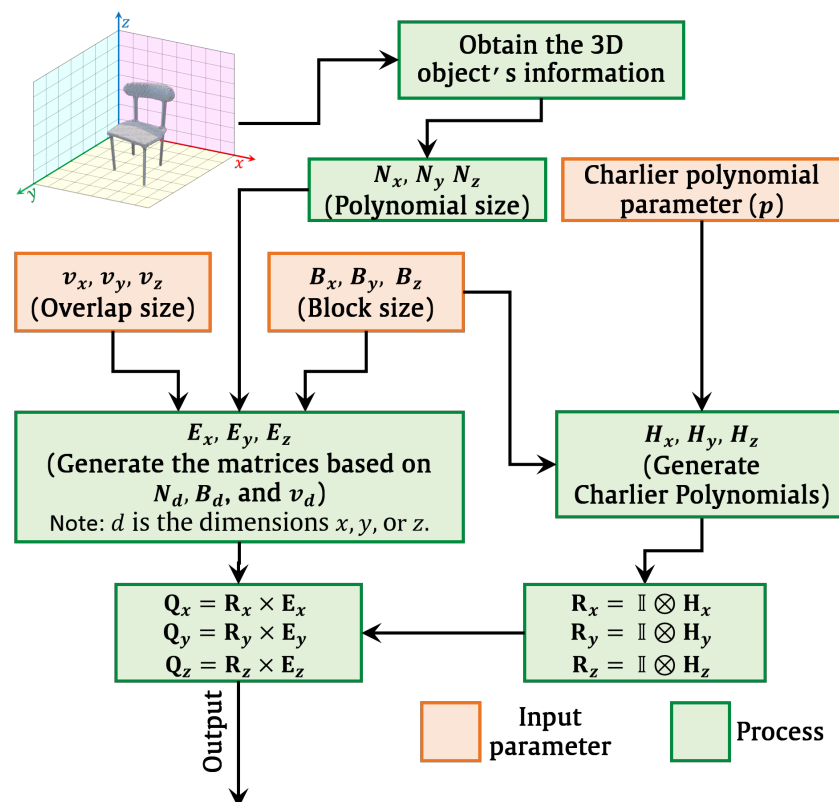


Figure 7. Flow diagram demonstrates the generation of the matrices \mathbf{Q}_x , \mathbf{Q}_y , and \mathbf{Q}_z .

After the matrices (\mathbf{Q}_x , \mathbf{Q}_y , and \mathbf{Q}_z) are generated, the images are transformed into the Charlier moment domain to extract features (see Algorithm 2). Then, these features are normalized to obtain the feature vector. Finally, the objects are classified based on the extracted features.

Algorithm 2 The 3D moments computation [83]**Input:** F = 3D image, Q_d = Charlier polynomials.**Output:** FV = Charlier moments.

- 1: Generate extended 3D image (G) from the 3D image F {Equation (22).}
- 2: Get stored Charlier polynomials Q_x , Q_y , and Q_z {Using Equation (26).}
- 3: **for** $z = 1$ to Ord_z **do**
- 4: $M \leftarrow M + R_z \otimes [Q_x G Q_y]$
- 5: **end for**
- 6: $FV \leftarrow \text{reshape}(M)$ {Reshape the computed moments as a feature vector.}
- 7: **return** FV {Note: in the training and testing phases, the feature vector is normalized.}

In this paper, the normalized feature vector is obtained and considered an input to the classifier. To this end, a label (ID) is considered for each input image of the objects. The classification procedure is performed in this paper using SVM. The SVM technique is selected here due to its effectiveness in optimizing the margin between hyperplane separation classes and data [84]. Furthermore, the SVM technique can be very efficient for object recognition. This is attributed to the fact that SVM is more robust to signal fluctuation [85]. In this paper, LIB-SVM is used in the classification process [86].

Figure 8 shows a model of the proposed 3D object recognition method.

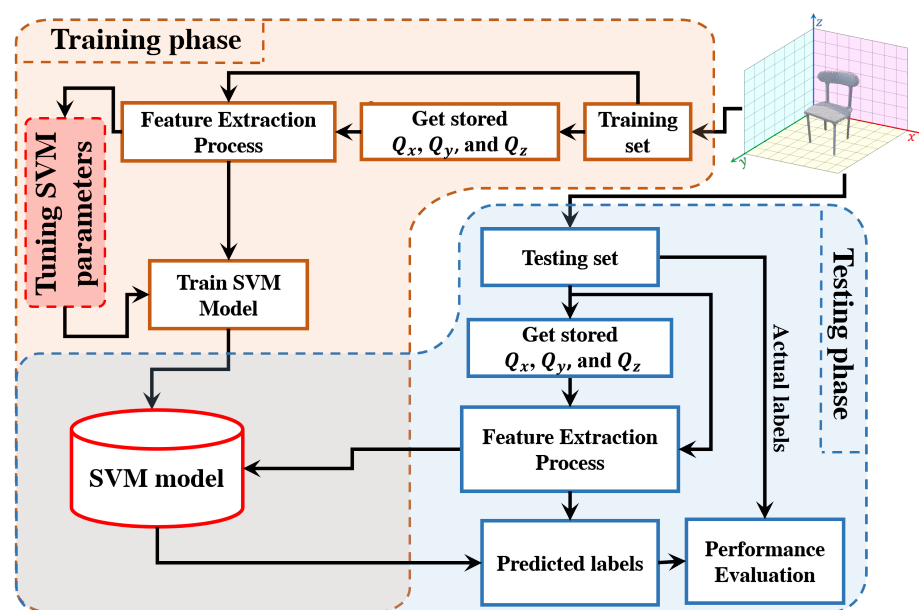


Figure 8. A diagram shows the flow process for the proposed 3D object recognition method.

4. Experiments and Discussions

In this section, the performance of the proposed Charlier polynomials algorithm for 3D object recognition is evaluated. In this experiment, the well-known McGill dataset—developed in [77]—is used as a benchmark dataset. In particular, this dataset contains 19 classes, denoted as 3D objects. These 3D objects are named as planes, spiders, spectacles, snakes, pliers, octopus, teddies, dolphins, fours, ants, humans, tables, chairs, dinosaurs, fishes, hands, cups, craps, and birds. Samples of the aforementioned 3D objects are shown in Figure 9.

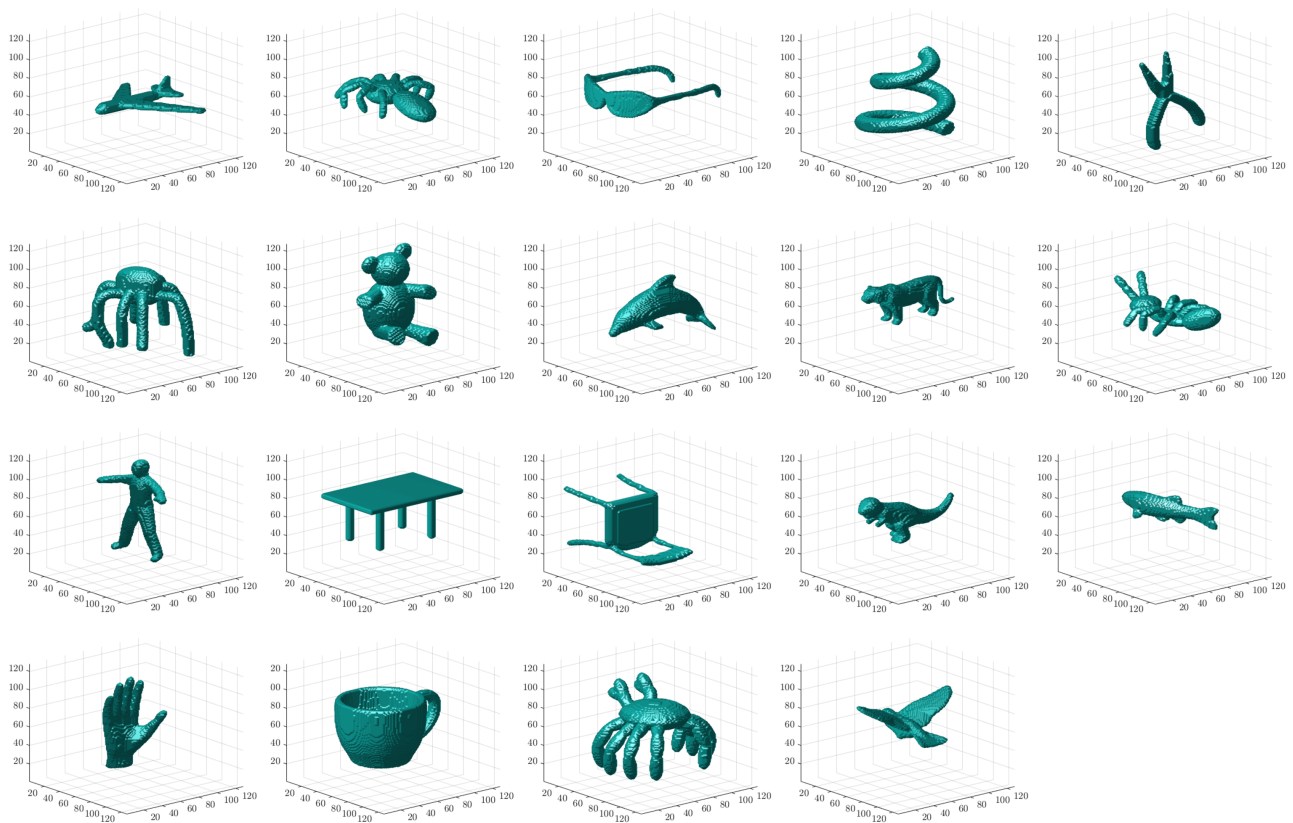


Figure 9. An illustration of samples of 3D objects.

In this experiment, the results are obtained over 19 different objects and various effects. These effects are translations and rotations. The sample objects are translated in the x , y , and z axes and their combinations (xy , xz , yz , and xyz) range from (1, 1, 1) to (10, 10, 10) with a step of (1, 1, 1). In addition, for each direction, the sample objects are rotated in the x , y , z , xy , xz , yz , and xyz axes between 10° and 360° with a step of 10° . The resulting number of samples per object is 1252, which produces a total number of 23,788 samples for all objects.

The flow diagram process of the 3D object recognition is shown in Figure 8. For the 3D object recognition, different block sizes are considered in this experiment, which are given by the block sizes of $64 \times 64 \times 64$, $32 \times 32 \times 32$, and $16 \times 16 \times 16$. Tables 1–3 present the performance results of block sizes of $64 \times 64 \times 64$, $32 \times 32 \times 32$, and $16 \times 16 \times 16$, respectively. Besides, different overlap sizes are also considered in addition to the sizes of the testing and training sets considered during this experiment, which are given as 70% and 30%, respectively. As discussed earlier, the proposed solution for 3D object recognition and feature extraction is Charlier polynomials (see Algorithm 1) with parameter $p = \text{Block Size}/2$. The SVM model is used in the proposed algorithm for object classification. The LIB-SVM library developed in [86] is used to train the extracted features. The SVM kernel exploits LIB-SVM and uses the radial basis function. In the training phase, five-fold cross-validation is utilized to obtain the values of the SVM parameters (see Figure 2). The recognition accuracy is the number of correct predictions divided by the total number of predictions as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (28)$$

Tables 1–3 reported the recognition rate for clean and noisy environments. First, we will discuss the clean environment results; then, the noisy environment will be considered successively for different types of noise.

The results in Table 1 show that the accuracy of block size of $64 \times 64 \times 64$ starts at 68.25% and increases to 80.04% as the overlap block size is increased from $0 \times 0 \times 0$ to $16 \times 16 \times 16$, which shows an improvement ratio of 14.73%. This implies that increasing the overlap block size can help in improving the recognition accuracy. For the block size of $32 \times 32 \times 32$ given in Table 2, the object recognition accuracy starts with a value of 76.28% at overlap size of $0 \times 0 \times 0$, which achieves an accuracy improvement of 8.03% higher than that obtained with the block size of $64 \times 64 \times 64$ at an overlap size of $0 \times 0 \times 0$. In addition, the object recognition accuracy of the block size of $32 \times 32 \times 32$ is increased to 80.10% at an overlap block size of $4 \times 4 \times 4$. For the block size of $16 \times 16 \times 16$ given in Table 3, the object recognition accuracy is increased from 70.58% to 80.10% as the overlap size is increased from $0 \times 0 \times 0$ to $2 \times 2 \times 2$.

Table 1. Evaluation of the accuracy (%) of the proposed algorithm of 3D object recognition for the block size of ($64 \times 64 \times 64$) under different environments.

Environment	Block Size = $64 \times 64 \times 64$				
	Overlap Size ($v_x \times v_y \times v_z$)				16
	0	2	4	8	
Clean	68.25	70.42	72.44	78.52	80.04
GN 0.0001	39.44	61.18	68.96	78.58	80.16
GN 0.0002	36.95	57.48	67.84	78.58	80.18
GN 0.0003	35.32	55.44	67.14	78.59	80.13
GN 0.0004	33.75	53.44	66.67	78.65	80.01
GN 0.0005	32.26	51.38	66.23	78.63	80.03
SPN 0.1	37.86	58.80	68.36	78.56	80.20
SPN 0.2	31.33	49.28	65.53	78.62	80.04
SPN 0.3	29.39	42.71	61.52	78.36	79.72
SPN 0.4	27.15	39.24	57.66	78.14	78.97
SPN 0.5	26.09	36.56	52.81	77.67	78.32
SPKN 0.2	68.24	70.41	72.40	78.55	80.03
SPKN 0.4	68.22	70.47	72.33	78.53	80.04
SPKN 0.6	68.17	70.39	72.37	78.46	80.04
SPKN 0.8	68.25	70.35	72.43	78.48	79.97
SPKN 1.0	68.21	70.51	72.30	78.53	80.03

To this end, the highest accuracy performance is achieved at a block size of $64 \times 64 \times 64$ when an overlap size of $16 \times 16 \times 16$ is used, at a block size of $32 \times 32 \times 32$ when an overlap size of $4 \times 4 \times 4$ is exploited, and at a block size of $16 \times 16 \times 16$ when an overlap size of $2 \times 2 \times 2$ is utilized, as illustrated in Tables 1–3, respectively. In a nutshell, the best accuracy can be achieved when the overlap block size is increased and the block size is decreased.

Different noisy environments are considered for further clarification and evaluation of the proposed object recognition method, and the results for each type are reported. It is worth noting that (GN) stands for Gaussian noise, (SPN) stands for salt-and-pepper noise, and SPKN stands for Speckle noise. Note that different noise levels are considered for each type of noise. From Table 1, it is obvious for the case of GN with all its different densities values from 0.0001 to 0.0005 that the accuracy is increased as the values of the overlap block size are increased. In addition, the same observation is perceived for SPN and SPKN for all noise density values. Moreover, Table 2 shows that for all types of noise and noise densities, higher accuracy is achieved at the highest overlap block size. On the other hand, for a block

size of $16 \times 16 \times 16$, higher accuracy is obtained for the overlap block size equal to $1 \times 1 \times 1$ for all noisy environments.

Table 2. Evaluating the accuracy (%) of the proposed algorithm of 3D object recognition for the block size of $(32 \times 32 \times 32)$ under different environments.

Environment	Block Size = $32 \times 32 \times 32$			
	Overlap Size ($v_x \times v_y \times v_z$)			
	0	1	2	4
Clean	76.28	77.18	78.86	80.10
GN 0.0001	76.21	77.15	78.79	80.16
GN 0.0002	76.16	77.11	78.73	80.14
GN 0.0003	76.18	77.02	78.67	80.11
GN 0.0004	75.98	77.01	78.60	80.14
GN 0.0005	75.75	76.98	78.62	80.14
SPN 0.1	76.25	77.21	78.80	80.16
SPN 0.2	75.46	76.94	78.56	80.10
SPN 0.3	74.39	76.56	78.22	79.92
SPN 0.4	73.79	75.64	78.00	79.58
SPN 0.5	73.03	74.78	77.64	79.21
SPKN 0.2	76.25	77.22	78.90	80.11
SPKN 0.4	76.26	77.18	78.87	80.13
SPKN 0.6	76.33	77.17	78.90	80.11
SPKN 0.8	76.28	77.21	78.84	80.13
SPKN 1.0	76.25	77.18	78.84	80.04

The results show that the best-case scenario can be obtained at a block size of $32 \times 32 \times 32$ and an overlap size of $4 \times 4 \times 4$. The accuracy of object recognition starts at very low values with the block size of $64 \times 64 \times 64$, which is given in Table 1 when an overlap size of $0 \times 0 \times 0$ is considered. The recognition accuracy is then increased to the highest values at a block size of $64 \times 64 \times 64$ and an overlap size of $16 \times 16 \times 16$, as given in Table 1; a block size of $32 \times 32 \times 32$ and overlap size of $4 \times 4 \times 4$, as shown in Table 2; and a block size of $16 \times 16 \times 16$ and overlap size of $2 \times 2 \times 2$, as demonstrated in Table 3.

To evaluate the performance of the presented algorithm, a comparison is performed with existing works in terms of recognition accuracy. The existing works in the comparison are Geometric Moment Invariants (GMI), Tchebichef–Tchebichef–Krawtchouk Moment Invariants (TTKMI), Tchebichef–Krawtchouk–Krawtchouk Moment Invariants (TKKMI), Krawtchouk–Krawtchouk–Krawtchouk Moment Invariants (KKKMI), Tchebichef–Tchebichef–Tchebichef Moment Invariants (TTTMI), Hahn Moment Invariants (HMI), Krawtchouk Moment Invariants (KMI), Tchebichef Moment Invariants (TMI), and Direct Krawtchouk Moment Invariants (DKMI). The results of the presented and existing works are reported in Table 4.

It can be observed from Table 4 that the average recognition accuracy of the presented algorithm outperforms the accuracy of the existing works. According to the results obtained from this table, the recognition accuracy of the presented algorithm is significantly high compared with that computed from the existing algorithms for all the given values of the block size ($64 \times 64 \times 64$, $32 \times 32 \times 32$, and $16 \times 16 \times 16$) and overlap size ($16 \times 16 \times 16$, $4 \times 4 \times 4$, and $1 \times 1 \times 1$). Therefore, it can be concluded that the presented algorithm can be useful in object recognition applications.

Table 3. Evaluation of the accuracy (%) of the proposed algorithm of 3D object recognition for the block size of $(16 \times 16 \times 16)$ under different environments.

Environment	Block Size = $16 \times 16 \times 16$		
	Overlap Size ($v_x \times v_y \times v_z$)		
	0	1	2
Clean	70.58	80.24	80.10
GN 0.0001	70.73	80.31	80.08
GN 0.0002	70.75	80.27	80.08
GN 0.0003	70.76	80.30	80.04
GN 0.0004	70.80	80.32	80.06
GN 0.0005	70.80	80.34	79.99
SPN 0.1	70.72	80.30	80.11
SPN 0.2	70.83	80.30	80.08
SPN 0.3	70.76	80.10	79.80
SPN 0.4	70.61	79.77	79.51
SPN 0.5	70.44	79.55	79.31
SPKN 0.2	70.63	80.28	80.10
SPKN 0.4	70.55	80.28	80.06
SPKN 0.6	70.55	80.30	80.08
SPKN 0.8	70.51	80.35	80.06
SPKN 1.0	70.65	80.32	80.10

Table 4. Comparison on the McGill database between the presented algorithm and existing works.

Algorithm Name	Average Accuracy
GMI [26]	70.26%
TTKMI [26]	72.87%
TKKMI [26]	72.19%
KKKMI [26]	71.11%
TTTMI [26]	71.57%
TMI [30]	60.54%
KMI [30]	60.32%
HMI [30]	60.89%
DKMI [30]	62.01%
Ours (block size = $64 \times 64 \times 64$, and overlap size = $16 \times 16 \times 16$)	79.87%
Ours (block size = $32 \times 32 \times 32$, and overlap size = $4 \times 4 \times 4$)	80.02%
Ours (block size = $16 \times 16 \times 16$, and overlap size = $1 \times 1 \times 1$)	80.21%

Furthermore, in order to provide further performance evaluation of the proposed method, the computation time of the proposed algorithm is compared with the traditional algorithm. To this end, Figure 10 illustrates an average computation time for 10 runs for both the proposed and traditional algorithm under different values of block sizes and overlap sizes. In addition, the percentage performance improvement between the proposed

algorithm and the traditional algorithm is also provided. This percentage performance improvement is obtained by dividing the results from the traditional algorithm by those obtained from the proposed algorithm. Figure 10 shows that the proposed algorithm significantly outperforms the traditional algorithm, where the average performance improvement across whole values is recorded as around 4.70 compared with the traditional algorithm. The proposed recognition algorithm achieves the highest percentage performance improvement when the block size is $16 \times 16 \times 16$ with an overlapped size of $4 \times 4 \times 4$, which is recorded as 7.86. This clearly signifies the robustness of our algorithm when a small block size, i.e., $16 \times 16 \times 16$, is considered.

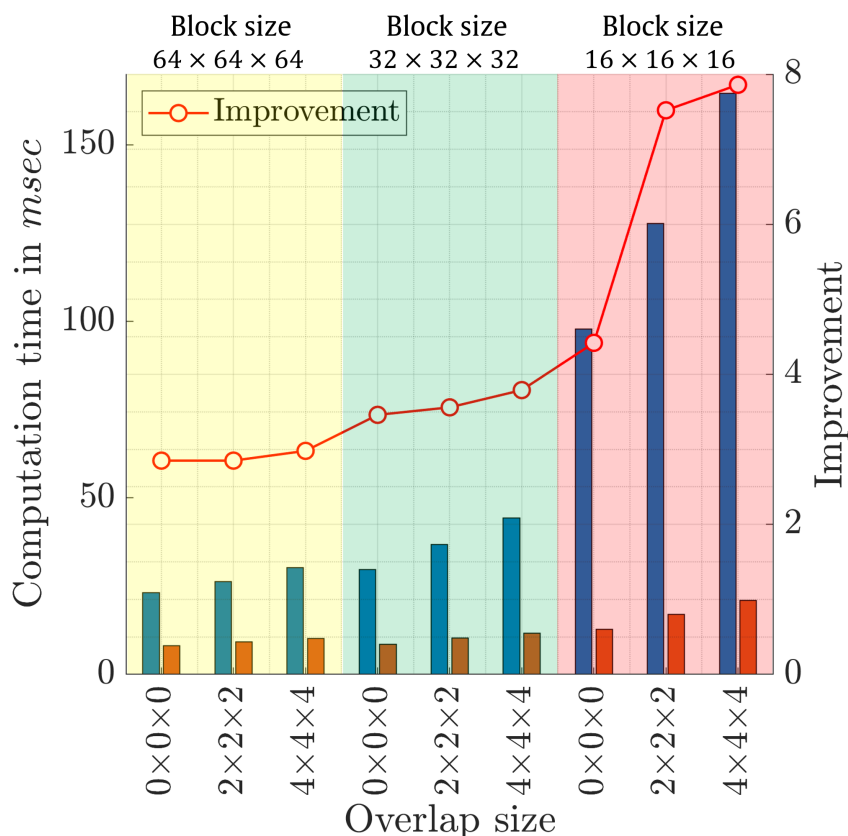


Figure 10. Computation time in msec comparing the proposed and traditional algorithms.

5. Conclusions

This paper presents an efficient algorithm for 3D object recognition with low computational complexity and fast execution time based on Charlier polynomials. The proposed algorithm has a powerful capability for extracting the features of the 3D object in a fast manner. This was attributed to the overlapped block-processing technique, which allows the signals to be virtually extended to auxiliary matrices to avoid the time delay during loops computation. In addition, in order to characterize the effectiveness of the proposed 3D object recognition method, a noise environment was considered in the evaluation and comparison. This paper also implemented the SVM algorithm to classify the 3D object features. The proposed 3D object recognition method was evaluated under different environments. The results illustrate that the proposed 3D object approach achieved high recognition accuracy as well as low computation time under the different noisy environments considered. This achievement signifies the importance of the proposed 3D object recognition method for future applications.

Author Contributions: Conceptualization, B.M.M. and S.H.A.; methodology, B.M.M. and S.H.A.; validation, M.A.N., M.A., A.H. and D.A.-J.; investigation, M.A.N., M.A., A.H. and D.A.-J.; resources, B.M.M.; writing—original draft preparation, B.M.M., S.H.A., M.A.N. and M.A.; writing—review and editing, A.H. and D.A.-J.; visualization, B.M.M., S.H.A., A.H. and D.A.-J.; supervision, S.H.A.; project administration, S.H.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: All data are available within the manuscript.

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

Abbreviations

The following abbreviations are used in this manuscript:

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
IoT	Internet of Things
TRSI	Translation, rotation, and scale invariants
CNN	Convolutional neural networks
ANN	Artificial neural network
OM	Orthogonal moments
OP	Orthogonal polynomials
TTR	Three-term recurrence
FOBP	Fast overlapped block processing
SVM	Support vector machine
GN	Gaussian noise
SPN	Salt-and-Pepper noise
SPKN	Speckle noise

References

1. Alsabah, M.; Naser, M.A.; Mahmmoud, B.M.; Abdulhussain, S.H.; Eissa, M.R.; Al-Baidhani, A.; Noordin, N.K.; Sait, S.M.; Al-Utaibi, K.A.; Hashim, F. 6G wireless communications networks: A comprehensive survey. *IEEE Access* **2021**, *9*, 148191–148243. [\[CrossRef\]](#)
2. Maafiri, A.; Elharrouss, O.; Rfifi, S.; Al-Maadeed, S.A.; Chougali, K. DeepWTPCA-L1: A new deep face recognition model based on WTPCA-L1 norm features. *IEEE Access* **2021**, *9*, 65091–65100. [\[CrossRef\]](#)
3. Lim, K.B.; Du, T.H.; Wang, Q. Partially occluded object recognition. *Int. J. Comput. Appl. Technol.* **2011**, *40*, 122–131. [\[CrossRef\]](#)
4. Akheel, T.S.; Shree, V.U.; Mastani, S.A. Stochastic gradient descent linear collaborative discriminant regression classification based face recognition. *Evol. Intell.* **2021**, *15*, 1729–1743. [\[CrossRef\]](#)
5. Ahmed, S.; Frikha, M.; Hussein, T.D.H.; Rahebi, J. Optimum feature selection with particle swarm optimization to face recognition system using Gabor wavelet transform and deep learning. *BioMed Res. Int.* **2021**, *2021*, 6621540. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Zhao, C.; Li, X.; Dong, Y. Learning blur invariant binary descriptor for face recognition. *Neurocomputing* **2020**, *404*, 34–40. [\[CrossRef\]](#)
7. Chen, Z.; Wu, X.J.; Yin, H.F.; Kittler, J. Noise-robust dictionary learning with slack block-diagonal structure for face recognition. *Pattern Recognit.* **2020**, *100*, 107118. [\[CrossRef\]](#)
8. Zhao, G.; Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 915–928. [\[CrossRef\]](#)
9. Kumar, V.V.; Murty, G.S.; Kumar, P.S. Classification of facial expressions based on transitions derived from third order neighborhood LBP. *Glob. J. Comput. Sci. Technol.* **2014**, *14*, 1–12.
10. Zhang, Y.; Hu, C.; Lu, X. IL-GAN: Illumination-invariant representation learning for single sample face recognition. *J. Vis. Commun. Image Represent.* **2019**, *59*, 501–513. [\[CrossRef\]](#)
11. Jian, M.; Dong, J.; Lam, K.M. FSAM: A fast self-adaptive method for correcting non-uniform illumination for 3D reconstruction. *Comput. Ind.* **2013**, *64*, 1229–1236. [\[CrossRef\]](#)
12. Luciano, L.; Hamza, A.B. Deep learning with geodesic moments for 3D shape classification. *Pattern Recognit. Lett.* **2018**, *105*, 182–190. [\[CrossRef\]](#)
13. Jian, M.; Yin, Y.; Dong, J.; Zhang, W. Comprehensive assessment of non-uniform illumination for 3D heightmap reconstruction in outdoor environments. *Comput. Ind.* **2018**, *99*, 110–118. [\[CrossRef\]](#)

14. Al-Ayyoub, M.; AlZu'bi, S.; Jararweh, Y.; Shehab, M.A.; Gupta, B.B. Accelerating 3D medical volume segmentation using GPUs. *Multimed. Tools Appl.* **2018**, *77*, 4939–4958. [\[CrossRef\]](#)
15. AlZu'bi, S.; Shehab, M.; Al-Ayyoub, M.; Jararweh, Y.; Gupta, B. Parallel implementation for 3d medical volume fuzzy segmentation. *Pattern Recognit. Lett.* **2020**, *130*, 312–318. [\[CrossRef\]](#)
16. Lakhili, Z.; El Alami, A.; Mesbah, A.; Berrahou, A.; Qjidaa, H. Robust classification of 3D objects using discrete orthogonal moments and deep neural networks. *Multimed. Tools Appl.* **2020**, *79*, 18883–18907. [\[CrossRef\]](#)
17. Sadjadi, F.A.; Hall, E.L. Three-Dimensional Moment Invariants. *IEEE Trans. Pattern Anal. Mach. Intell.* **1980**, *PAMI-2*, 127–136. [\[CrossRef\]](#)
18. Guo, X. Three dimensional moment invariants under rigid transformation. In *International Conference on Computer Analysis of Images and Patterns*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 518–522.
19. Suk, T.; Flusser, J. Tensor method for constructing 3D moment invariants. In *International Conference on Computer Analysis of Images and Patterns*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 212–219.
20. Xu, D.; Li, H. 3-D affine moment invariants generated by geometric primitives. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, 20–24 August 2006; Volume 2, pp. 544–547.
21. Xu, D.; Li, H. Geometric moment invariants. *Pattern Recognit.* **2008**, *41*, 240–249. [\[CrossRef\]](#)
22. Mesbah, A.; Berrahou, A.; Hammouchi, H.; Berbia, H.; Qjidaa, H.; Daoudi, M. Non-rigid 3D model classification using 3D Hahn Moment convolutional neural networks. In *Proceedings of the EG Workshop 3D Object Retrieval*, Delft, The Netherlands, 16 April 2018.
23. Amakdouf, H.; El Mallahi, M.; Zouhri, A.; Tahiri, A.; Qjidaa, H. Classification and recognition of 3D image of charlier moments using a multilayer perceptron architecture. *Procedia Comput. Sci.* **2018**, *127*, 226–235. [\[CrossRef\]](#)
24. Lakhili, Z.; El Alami, A.; Mesbah, A.; Berrahou, A.; Qjidaa, H. Deformable 3D shape classification using 3D Racah moments and deep neural networks. *Procedia Comput. Sci.* **2019**, *148*, 12–20. [\[CrossRef\]](#)
25. Mademlis, A.; Axenopoulos, A.; Daras, P.; Tzovaras, D.; Strintzis, M.G. 3D content-based search based on 3D Krawtchouk moments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, Chapel Hill, NC, USA, 14–16 June 2006; pp. 743–749.
26. Batioua, I.; Benouini, R.; Zenkouar, K.; Zahi, A.; Hakim, E.F. 3D image analysis by separable discrete orthogonal moments based on Krawtchouk and Tchebichef polynomials. *Pattern Recognit.* **2017**, *71*, 264–277. [\[CrossRef\]](#)
27. Abdhussain, S.H.; Mahmmoud, B.M. Fast and efficient recursive algorithm of Meixner polynomials. *J. Real-Time Image Process.* **2021**, *18*, 2225–2237. [\[CrossRef\]](#)
28. Sayyouri, M.; Karmouni, H.; Hmimid, A.; Azzayani, A.; Qjidaa, H. A fast and accurate computation of 2D and 3D generalized Laguerre moments for images analysis. *Multimed. Tools Appl.* **2021**, *80*, 7887–7910. [\[CrossRef\]](#)
29. Farokhi, S.; Shamsuddin, S.M.; Sheikh, U.U.; Flusser, J.; Khansari, M.; Jafari-Khouzani, K. Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform. *Digit. Signal Process.* **2014**, *31*, 13–27. [\[CrossRef\]](#)
30. Benouini, R.; Batioua, I.; Zenkouar, K.; Najah, S.; Qjidaa, H. Efficient 3D object classification by using direct Krawtchouk moment invariants. *Multimed. Tools Appl.* **2018**, *77*, 27517–27542. [\[CrossRef\]](#)
31. Hu, M.K. Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theory* **1962**, *8*, 179–187.
32. Radeaf, H.S.; Mahmmoud, B.M.; Abdhussain, S.H.; Al-Jumaily, D. A steganography based on orthogonal moments. In *Proceedings of the International Conference on Information and Communication Technology—ICICT '19*, Baghdad, Iraq, 15–16 April 2019; ACM Press: New York, NY, USA, 2019; pp. 147–153. [\[CrossRef\]](#)
33. 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. [\[CrossRef\]](#)
34. Xu, S.; Hao, Q.; Ma, B.; Wang, C.; Li, J. Accurate computation of fractional-order exponential moments. *Secur. Commun. Netw.* **2020**, *2020*, 8822126. [\[CrossRef\]](#)
35. Daoui, A.; Yamni, M.; Karmouni, H.; Sayyouri, M.; Qjidaa, H. 2D and 3D medical image analysis by discrete orthogonal moments. *Procedia Comput. Sci.* **2019**, *148*, 428–437.
36. Xia, Z.; Wang, X.; Zhou, W.; Li, R.; Wang, C.; Zhang, C. Color medical image lossless watermarking using chaotic system and accurate quaternion polar harmonic transforms. *Signal Process.* **2019**, *157*, 108–118. [\[CrossRef\]](#)
37. Jain, A.K.; Li, S.Z. *Handbook of Face Recognition*; Springer: London, UK, 2011; Volume 1.
38. AL-Utaibi, K.A.; Abdhussain, S.H.; Mahmmoud, B.M.; Naser, M.A.; Alsabah, M.; Sait, S.M. Reliable Recurrence Algorithm for High-Order Krawtchouk Polynomials. *Entropy* **2021**, *23*, 1162. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Liu, B.D.; Shen, B.; Gui, L.; Wang, Y.X.; Li, X.; Yan, F.; Wang, Y.J. Face recognition using class specific dictionary learning for sparse representation and collaborative representation. *Neurocomputing* **2016**, *204*, 198–210. [\[CrossRef\]](#)
40. Shakeel, M.S.; Lam, K.M. Deep-feature encoding-based discriminative model for age-invariant face recognition. *Pattern Recognit.* **2019**, *93*, 442–457. [\[CrossRef\]](#)
41. Liu, K.; Zheng, M.; Liu, Y.; Yang, J.; Yao, Y. Deep Autoencoder Thermography for Defect Detection of Carbon Fiber Composites. *IEEE Trans. Ind. Inform.* **2022**. [\[CrossRef\]](#)
42. Gao, S.; Dai, Y.; Li, Y.; Jiang, Y.; Liu, Y. Augmented flame image soft sensor for combustion oxygen content prediction. *Meas. Sci. Technol.* **2022**, *34*, 015401. [\[CrossRef\]](#)

43. Liu, K.; Yu, Q.; Liu, Y.; Yang, J.; Yao, Y. Convolutional Graph Thermography for Subsurface Defect Detection in Polymer Composites. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [\[CrossRef\]](#)
44. Hosny, K.M.; Abd Elaziz, M.; Darwish, M.M. Color face recognition using novel fractional-order multi-channel exponent moments. *Neural Comput. Appl.* **2021**, *33*, 5419–5435. [\[CrossRef\]](#)
45. Abdulhussain, S.H.; Mahmmoud, B.M.; AlGhadhban, A.; Flusser, J. Face Recognition Algorithm Based on Fast Computation of Orthogonal Moments. *Mathematics* **2022**, *10*, 2721. [\[CrossRef\]](#)
46. Mehdipour Ghazi, M.; Kemal Ekenel, H. A comprehensive analysis of deep learning based representation for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 34–41.
47. Guo, S.; Chen, S.; Li, Y. Face recognition based on convolutional neural network and support vector machine. In Proceedings of the 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 1–3 August 2016; pp. 1787–1792.
48. Asad, M.; Hussain, A.; Mir, U. Low complexity hybrid holistic–landmark based approach for face recognition. *Multimed. Tools Appl.* **2021**, *80*, 30199–30212. [\[CrossRef\]](#)
49. Abdulhussain, S.H.; Ramli, A.R.; Mahmmoud, B.M.; Saripan, M.I.; Al-Haddad, S.A.R.; Jassim, W.A. A New Hybrid form of Krawtchouk and Tchebichef Polynomials: Design and Application. *J. Math. Imaging Vis.* **2019**, *61*, 555–570. [\[CrossRef\]](#)
50. Hmimid, A.; Sayyouri, M.; Qjidaa, H. Fast computation of separable two-dimensional discrete invariant moments for image classification. *Pattern Recognit.* **2015**, *48*, 509–521. [\[CrossRef\]](#)
51. Abdulhussain, S.H.; Ramli, A.R.; Hussain, A.J.; Mahmmoud, B.M.; Jassim, W.A. Orthogonal polynomial embedded image kernel. In Proceedings of the Proceedings of the International Conference on Information and Communication Technology—ICICT '19, Baghdad, Iraq, 15–16 April 2019; ACM Press: New York, NY, USA, 2019; pp. 215–221. [\[CrossRef\]](#)
52. Jassim, W.A.; Raveendran, P.; Mukundan, R. New orthogonal polynomials for speech signal and image processing. *IET Signal Process.* **2012**, *6*, 713–723. [\[CrossRef\]](#)
53. Flusser, J.; Zitova, B.; Suk, T. *Moments and Moment Invariants in Pattern Recognition*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
54. Rahman, S.M.; Howlader, T.; Hatzinakos, D. On the selection of 2D Krawtchouk moments for face recognition. *Pattern Recognit.* **2016**, *54*, 83–93. [\[CrossRef\]](#)
55. Teh, C.H.; Chin, R.T. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 496–513. [\[CrossRef\]](#)
56. Chen, B.; Yu, M.; Su, Q.; Shim, H.J.; Shi, Y.Q. Fractional Quaternion Zernike Moments for Robust Color Image Copy-Move Forgery Detection. *IEEE Access* **2018**, *6*, 56637–56646. [\[CrossRef\]](#)
57. Kaur, P.; Pannu, H.S.; Malhi, A.K. Plant disease recognition using fractional-order Zernike moments and SVM classifier. *Neural Comput. Appl.* **2019**, *31*, 8749–8768. [\[CrossRef\]](#)
58. Teague, M.R. Image analysis via the general theory of moments. *Josa* **1980**, *70*, 920–930. [\[CrossRef\]](#)
59. Abdulhussain, S.H.; Mahmmoud, B.M.; Baker, T.; Al-Jumeily, D. Fast and accurate computation of high-order Tchebichef polynomials. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7311. [\[CrossRef\]](#)
60. Mahmmoud, B.M.; Abdulhussain, S.H.; Suk, T.; Hussain, A. Fast Computation of Hahn Polynomials for High Order Moments. *IEEE Access* **2022**, *10*, 48719–48732. [\[CrossRef\]](#)
61. Yang, B.; Dai, M. Image analysis by Gaussian–Hermite moments. *Signal Process.* **2011**, *91*, 2290–2303. [\[CrossRef\]](#)
62. Mukundan, R.; Ong, S.; Lee, P.A. Image analysis by Tchebichef moments. *IEEE Trans. Image Process.* **2001**, *10*, 1357–1364. [\[CrossRef\]](#)
63. Curtidor, A.; Baydyk, T.; Kussul, E. Analysis of Random Local Descriptors in Face Recognition. *Electronics* **2021**, *10*, 1358. [\[CrossRef\]](#)
64. Song, G.; He, D.; Chen, P.; Tian, J.; Zhou, B.; Luo, L. Fusion of Global and Local Gaussian–Hermite Moments for Face Recognition. In *Image and Graphics Technologies and Applications*; Wang, Y., Huang, Q., Peng, Y., Eds.; Springer: Singapore, 2019; pp. 172–183.
65. Paul, S.K.; Bouakaz, S.; Rahman, C.M.; Uddin, M.S. Component-based face recognition using statistical pattern matching analysis. *Pattern Anal. Appl.* **2021**, *24*, 299–319. [\[CrossRef\]](#)
66. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [\[CrossRef\]](#)
67. Ekenel, H.K.; Stiefelwagen, R. Local appearance based face recognition using discrete cosine transform. In Proceedings of the 2005 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005; pp. 1–5.
68. Abdulhussain, S.H.; Mahmmoud, B.M.; Flusser, J.; AL-Utaibi, K.A.; Sait, S.M. Fast Overlapping Block Processing Algorithm for Feature Extraction. *Symmetry* **2022**, *14*, 715. [\[CrossRef\]](#)
69. Kamaruzaman, F.; Shafie, A.A. Recognizing faces with normalized local Gabor features and spiking neuron patterns. *Pattern Recognit.* **2016**, *53*, 102–115. [\[CrossRef\]](#)
70. Ahonen, T.; Hadid, A.; Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [\[CrossRef\]](#) [\[PubMed\]](#)
71. Shen, L.; Bai, L. A review on Gabor wavelets for face recognition. *Pattern Anal. Appl.* **2006**, *9*, 273–292. [\[CrossRef\]](#)
72. Ahonen, T.; Hadid, A.; Pietikainen, M. Face recognition with local binary patterns. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 469–481.
73. Muqet, M.A.; Holambe, R.S. Local binary patterns based on directional wavelet transform for expression and pose-invariant face recognition. *Appl. Comput. Inform.* **2019**, *15*, 163–171. [\[CrossRef\]](#)

74. Shrinivasa, S.; Prabhakar, C. Scene image classification based on visual words concatenation of local and global features. *Multimed. Tools Appl.* **2022**, *81*, 1237–1256. [[CrossRef](#)]
75. Onan, A. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 2098–2117. [[CrossRef](#)]
76. Kim, J.Y.; Cho, S.B. Obfuscated Malware Detection Using Deep Generative Model based on Global/Local Features. *Comput. Secur.* **2022**, *112*, 102501. [[CrossRef](#)]
77. Siddiqi, K.; Zhang, J.; Macrini, D.; Shokoufandeh, A.; Bouix, S.; Dickinson, S. Retrieving articulated 3-D models using medial surfaces. *Mach. Vis. Appl.* **2008**, *19*, 261–275. [[CrossRef](#)]
78. Abdul-Hadi, A.M.; Abdulhussain, S.H.; Mahmmod, B.M. On the computational aspects of Charlier polynomials. *Cogent Eng.* **2020**, *7*, 1763553. [[CrossRef](#)]
79. Koekoek, R.; Lesky, P.A.; Swarttouw, R.F. *Hypergeometric Orthogonal Polynomials and Their q -Analogues*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010.
80. Mahmmod, B.M.; bin Ramli, A.R.; Abdulhussain, S.H.; Al-Haddad, S.A.R.; Jassim, W.A. Signal compression and enhancement using a new orthogonal-polynomial-based discrete transform. *IET Signal Process.* **2018**, *12*, 129–142. [[CrossRef](#)]
81. Abdul-Haleem, M.G. Offline Handwritten Signature Verification Based on Local Ridges Features and Haar Wavelet Transform. *Iraqi J. Sci.* **2022**, *63*, 855–865. [[CrossRef](#)]
82. Tippaya, S.; Sitjongsataporn, S.; Tan, T.; Khan, M.M.; Chamnongthai, K. Multi-modal visual features-based video shot boundary detection. *IEEE Access* **2017**, *5*, 12563–12575. [[CrossRef](#)]
83. Rivera-Lopez, J.S.; Camacho-Bello, C.; Vargas-Vargas, H.; Escamilla-Noriega, A. Fast computation of 3D Tchebichef moments for higher orders. *J.-Real-Time Image Process.* **2022**, *19*, 15–27. [[CrossRef](#)]
84. Byun, H.; Lee, S.W. A survey on pattern recognition applications of support vector machines. *Int. J. Pattern Recognit. Artif. Intell.* **2003**, *17*, 459–486. [[CrossRef](#)]
85. Awad, M.; Motai, Y. Dynamic classification for video stream using support vector machine. *Appl. Soft Comput.* **2008**, *8*, 1314–1325. [[CrossRef](#)]
86. Chang, C.C.; Lin, C.J. LIBSVM. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [[CrossRef](#)]