Advances in Optics and Computer Vision

Journal homepage: http://www.aocv.ir

AOCV

Extract corner points using Krawtchouk polynomials

Leila Rahbarimanesh

Department of Computer, South Tehran Branch, Islamic Azad University, Tehran, Iran
Corresponding Author, Email: l.rahbarimanesh@gmail.com

ARTICLE INFO

Article history:
- *Received: 2017-10-3*
- *Revised: 2017-12-16*
- *Accepted: 2017-12-28*
- *Available online: 2018-1-1*

Keywords:
Edge-detection
Corner points extraction
Anisotropic directional derivatives
Polynomials

ABSTRACT

The corner has important features and useful information from the objects in the image. The extraction of corner points is most often used in the preprocessing of image analysis. Therefore, it is considered as one of the most important and fundamental stages of image processing. In addition, it plays an essential role in machine vision and image enhancement, image compression, image matching, image tracking, and so on. In this article, corner points are detected in images using polynomials. A discrete orthogonal polynomial is a useful digital image processing tool for extracting object contours in various applications. Therefore, in the corner detection step algorithm, the Krawtchouk polynomial is used in the edge detection step. In the next step, an anisotropic direction derivative is calculated at each pixel in the contour, and finally the removal of the non-maximum suppression and the threshold in each contour is used to find the corners. The results of the experiments show that the proposed method has a good performance compared to other common algorithms.

1. Introduction

Corner Detection is based on images of dynamic and up-to-date topics in the image processing and machine vision. The corners contain important image information and are widely used in various image processing operations. A corner can be defined as the intersection of two edges. Also, a corner can be defined as a point where the two edges are strong and dominant in a neighborhood of points. By definition, the corners are points of the image that changes in the degrees of gray in their neighbors are severe and therefore can be distinguished well from their neighboring points [1].

The methods for extracting corner points can be divided into two main categories [19]:

- *Intensity-based method: A predetermined scale to estimate points of interest directly estimate the pixel values of the image.*
- *Contour-based method: First, it provides a flat curve using edge detectors, then the highest point that has a curvature in the curve.*

Of course, in some articles, another category is also called template-based methods. A pattern-based model finds corners by setting the image signal into a predefined model. As outlined in [2], corner detection methods use different representation patterns to match the image based on the pattern. The relationship between the pattern and the image is used to identify the corner. However, this kind of method has some drawbacks. For example, the representative model cannot cover all possible corner conditions. Therefore, the diagnostic function depends on the choice of the appropriate pattern. Moreover, the relationship between the pattern and the image is determined. Corner detection techniques are based on contour based edge detection. In this set of methods, the edges are identified in the first image. After that, the corners are identified along the path. The first corner detector, or indeed the start of corner detectors, was Moravec Corner detection. Moravec has proposed a corner detector for points of interest, taking into account a local window in the image and determining the average variation in the intensity obtained by moving the window with small steps in four directions [20]. Harris has modified Moravec's algorithm, which is done by estimating autocorrelation from first-order derivatives [4]. Since the derivative is used in this method, this method is sensitive to noise. Smith and Brady have presented a straightforward method known as Susan and not dependent on the derivative of the image. usan areas reach a minimum when they are at the corner. The susan algorithm is not sensitive to noise, and it has a high speed in that it uses simple operators [1].

Some of the famous corner detection methods include: Susan Detector, Harris Detector, Wavelet Detector. The original Harris detector was presented in [4] and the corner measure is constructed from the local auto correlation matrix of the gradients in different directions [12]. Auto correlation function measures the self-similarity of a signal. After calculating this matrix, there are three situations. Note that this is a 2 x 2 matrix, and so there are two eigenvalues for this matrix. If both of the eigenvalues are large, that means the feature point is a corner. [4].

The main idea of the Harris detector is to calculate eigen values and eigen vectors of a small area [4, 5]. Then, using the largest of the two eigen values to compute some of the functions. Finally, it uses the value of the function and one threshold for corner detection. The susan corners detector was presented in [6]. susan corner detector does not need to be derived. This is why it can detect corners well in the presence of noise [6]. The main idea of the Susan detector is to use a mask to count the number of pixels with the same intensity as the center of the pixel [3]. By comparing the number of pixels with the same intensity with the center pixel and a threshold, the detector can determine if the pixel is the center of one corner. Information on a wavelet-based detector is presented in [3]. Since the Susan detector and the Harris detector are both the most well-known corner detectors, the authors in [7] compared the performance of these two detectors in terms of complexity, stability, and execution time. The simulation results showed that the Harris detector is better than the Susan detector. As described in [7], Susan uses a fixed overall threshold instead of an adaptive threshold.

In 2010, in a paper [8], Zhang et al. Proposed a new multi-dimensional nonlinear structure based on the corner detection algorithm to improve the efficiency of the Harris corner detection algorithm. Considering both the distance and the gradient of the distance from the neighboring pixels, a nonlinear two-way tensor structure is constructed to examine the local pattern of the image. In addition, a multi-scale filter design has been developed to express the partial structures of real corners based on their different characteristics on different scales. The comparison between suggested methods shows that it has better performance in both the detection and focusing accuracy of the corners.

In 2011, Zhang et al. [9] presented a feature detector named Harris Laplace's Improvement to obtain a repeatability above the original Harris Laplace. In this new method, all detected points are tracked and grouped at each scale starting with the largest scale on the scale of the scale, which makes each group represent a local structure at first stages Then the point in each group is simultaneously directed to the maximum points of the measured corner and the normalized scale of the Laplace function is chosen. Finally, these points are described in terms of the fixed-value conversion scale (SIFT). Experimental results show that the proposed method has a higher repeatability than the original Harris Laplace. Meanwhile, the new method was investigated by registration the image. Compared to (SIFT), the accuracy of the recorded precision was obtained more precisely than several sensors of remote sensing images using advanced methods. In this method, the Harris points are tracked in all the images smoothly on the scale of space and simultaneously. the points detected from all scale levels in scale–space were assigned into different groups according to certain rules. Then the most characteristic point was selected as final feature point in each group and the other points were discarded. So the redundant points can be removed during the detection step to decrease the complexity of following computation [9]. Experimental results confirmed that the improved detector has higher repeatability and higher accuracy than the original Harris Laplace.

Many Harris corner detectors only use black and white data from an image, however, color information is wasted from an image. In 2013, using color information from the image, a New Harris corner Point Detector was proposed in [10]. In this method, the Harris corner detector uses a gray image using gray level information and a color image using the RGB data. After the corner points are identified in both black and white and color images, cross correlation, and consensus samples are randomly used to find matching points of the corner. This paper is organized as follows. In the second part, the proposed method and a series of related concepts are presented. Experiments and performance comparison are reported in third section. Finally, the Conclusion of the paper is presented in the fourth section.

2. Proposed Method

This section describes the proposed algorithm for extracting corner points in the images and describes the various sections. This method tries to detect corners in images using polynomials. In this paper, the method ANDD[12], edge detection algorithm using polynomial[15] is used. A discrete orthogonal polynomial is a useful digital image processing tool for extracting object contours in various applications. Thus, in this paper, at the edge detection stage an alternative method that extends beyond the classical first-order differential operators is expanded using the orthogonal Krawtchouk polynomial properties to reach a first-order differential operator [15]. Therefore, image smoothing with a two-dimensional Gaussian filter is not required to adjust the derived type [15]. The related concepts and the full description of the proposed algorithm are described below.

2-1. Anisotropic Directional Derivatives

The edges and corners are anisotropic features in the images. Conventional edges and corners detection methods often use gradients to create anisotropic intensity difference around an edge or corner pixel [12]. Generally, the gradients are derived from anisotropic Gaussian filter, which follows two first-order partial-order operator [11]. The use of an isotropic Gaussian smoothing operator and gradient is a dilemma. Small-scale Gaussian filters with good positioning and resolution are good edges, but are sensitive to noise, while Gaussian filters are robust against noise, but in localization (position) and clarity of the edges is weak. In this section, description of Gaussian Anisotropic Directional Derivatives has been presented [12]. The anisotropic directional derivative (ANDD) is calculated by a Gaussian smoothing filter through a directional derivative operator. An anisotropic directional derivative (ANDD) describes not only the intensity of the change around a pixel, but also the corner detection potential [11]. This makes it possible to provide a corner detector based on the ANDDS. In this section, the concept and features of the ANDDS are briefly examined and their ability to identify the corners is shown.

2-1-1. Anisotropic Gaussian kernels and an anisotropic Gaussian directional derivative

A long two-dimension Gaussian function is expressed as follows [12]:

$$g_{\sigma,p}(x)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2\sigma^2}X^T\begin{pmatrix}\rho^{-2}&0\\0&\rho^2\end{pmatrix}X\right) \quad (1)$$

$$\rho\geq 1,\sigma>0,X=[x,y]^T$$

In the above relation, σ is the scale, $\rho \geq 1$ is anisotropic factor and $X=[x,y]^T$.

Its partial derivative with respect to the second variable is as follows:

$$\frac{\partial g_{\sigma,\rho}}{\partial y}(x)=-\frac{\rho^2 y}{\sigma^2}g_{\sigma,p}(x) \quad (2)$$

By turning the argument in formula (1) under an angle θ, an anisotropic Gaussian kernel is obtained as follows:

$$\varphi_{\sigma,\rho,\theta}(x)=g_{\sigma,\rho}(R_\theta X),R_\theta=\begin{pmatrix}\cos\theta&-\sin\theta\\\sin\theta&\cos\theta\end{pmatrix} \quad (3)$$

Where θ is the rotation angle and R_θ is the rotational matrix. , An ANDD filter is extracted in the $\pi/2+\theta$ direction as follows:

$$\psi_{\sigma,\rho,\theta}(x)=\frac{\partial g_{\sigma,\rho}}{\partial y}(R_\theta X) \quad (4)$$

The anisotropic directional derivative of an image I (x) in the direction of $\pi/2+\theta$ with a convolutional operator is obtained by the following equation:

$$\partial I_{\sigma,\rho}(x;\theta)\equiv\frac{\partial}{\partial\theta}(I*\varphi_{\partial,\rho,\theta}(x))=I*\psi_{\partial,\rho,\theta}(x) \quad (5)$$

Which is the characteristic of the variation of intensity around a pixel x.

In Fig. 1 Gaussian anisotropic kernel and ANDD filter are plotted in eight directions. Some corner detectors based on the intensity of use and gradient operator on the image smoothed by the isotropic Gaussian kernel are calculated. Isotropic Gaussian smoothing ensures that the image is indeterminate at the edges and corners, which are often singular singularities of an image function. However, smoothing also blurs local structures at the edges and at the corners. Therefore, the oriented derivatives in all directions are determined by the second partial derivative, and partial derivatives are difficult to distinguish between different types of corners. The recent edge or corner detectors use an Oriented multi-dimensional banked filter instead of a gradient operator to identify the anisotropic local structures on the edges and corners. Similarly, the ANND filter can also be interpreted as a directed bank filter and can extract anisotropic intensity variations around edges and corners pixels.

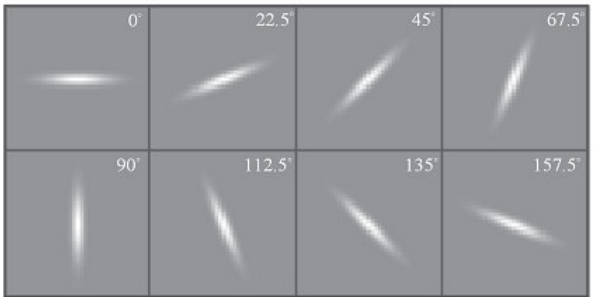

Figure 1- Anisotropic Gaussian kernel in eight directions [12]

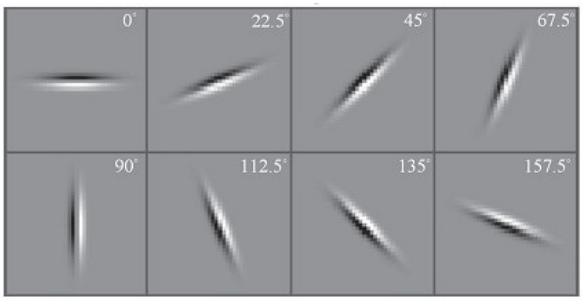

Figure 2- is an anisotropic direction derivative (ANDD) filter in eight directions [12]

The first two rows in Figures 1 and 2 represent the anisotropic Gaussian kernel in eight directions. Each kernel can suppress the noise while maintaining its edges along its axis of direction.

Representation of the directional derivative uses a unique smoothing kernel in each direction. Anisotropic smoothing not only suppresses noise, but also maintains an anisotropic structure of corners and edges. The ability of anisotropic Gaussian smoothing kernels to be clearly determined by their σ scale is independent of the

anisotropic factor ρ and the angle θ. The anisotropic directional derivative per pixel describes the variation of the directional intensity around the pixel. An anisotropic directional derivative can be used to identify different types of corners and smooth edge pixels. In addition, the anisotropic directional derivative is a periodic function with period π. As a result, an anisotropic directional derivative at a distance [0, π] is sufficient to describe the variation of the intensity direction around a pixel.

2-1-2. Discrete anisotropic direction derivative filters

Images are two-dimensional discrete signals in the integer z^2 networks. Therefore, anisotropic direction derivative filters should be transmitted to their discrete forms. The discrete anisotropic Gaussian kernels and anisotropic direction derivative filters are created according to the equation (6):

$$\varphi_{\sigma,\rho,k}(n) = g_{\sigma,\rho}(R_k n)$$
$$\psi_{\sigma,\rho,k}(n) = \frac{\partial g_{\sigma,\rho}}{\partial y}(R_k n)$$
$$R_k = \begin{pmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{pmatrix}$$
$$n = \begin{pmatrix} n_x \\ n_y \end{pmatrix} \in \mathbb{Z}^2 \qquad (6)$$
$$\theta_k = \frac{2\pi(k-1)}{k}$$
$$k = 1, 2, \ldots, K$$

where K is the number of direction and the vector of the integer $n = [n_x, n_y]$ represents the integer coordinates. From the discrete ANDD filter, the ANDD discrete representation is calculated as follows:

$$\partial\nabla(n;k) = \sum_{m_x}\sum_{m_y} I(n-m)\psi_{\sigma,\rho,k}(m)$$
$$n = [n_x, n_y]^T \in \mathbb{Z}^2 \qquad (7)$$
$$m = [m_x, m_y]^T \in \mathbb{Z}^2$$

Where m and n denote the coordinates in the integer 2-dimensional Z^2 network. The area representation of the discrete normal ANDD, the remaining representatives of the ANDD, and the remaining area is calculated as follows:

$$\hat{\partial}\nabla(n;k) = \frac{\partial\nabla(n;k)}{\max_\theta\{|\partial\nabla(n;k)|\}}$$
$$k_0 = \arg_k \max\{|\partial\nabla(n;k)|\} \qquad (8)$$
$$R - \hat{\partial}\nabla(n;k) \equiv \hat{\partial}\nabla(n;k) - sign\{\hat{\partial}\nabla(n;k_0)\}\xi k_0\Delta\theta, \rho(k\Delta\theta)$$
$$R - area(n) = \frac{\Delta\theta}{2}\sum_{k=1}^{p}(R - \hat{\partial}\nabla(n;k))^2 \qquad (9)$$
$$\Delta\theta = \frac{2\pi}{p}$$

The flowchart of the proposed algorithm is as follows:

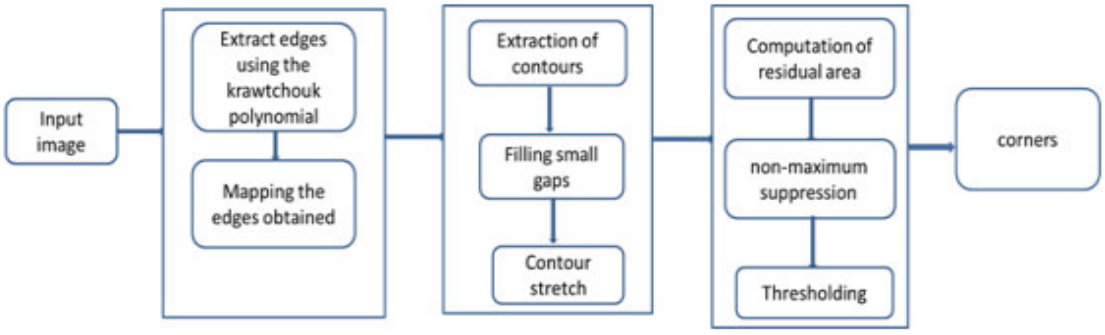

Figure 3-Flowchart Proposed Algorithm

The following is an explanation of each part of the flowchart of the proposed algorithm.

2-2.Expression of the proposed algorithm using Krawtchouk polynomial

Part I:

- *Detect edges of the image based on Krawtchouk's polynomial:*

The edge detection algorithm is defined as:

1. Calculate gradient value matrix

The P_x and P_y matrices are calculated based on the Krawtchouk polynomial. At each point (i, j), the gradient coefficient G (i, j) is defined as follows:

$$G(i,j) = \sqrt{P_x^2(i,j) + P_y^2(i,j)} \qquad (10)$$

2. Find the first threshold value and the strong points of the edge

The first level of the adaptive threshold is obtained as follows:

$$\tau_{h_1} = mean\left(G(i,j)\right) + k \times \text{standard deviation}\left(G(i,j)\right) \qquad (11)$$
$$k \epsilon R^+$$

3. Calculate the second level of the threshold and the weak points of the edge

Only the point (i_1, j_1)*for each mean* $G(i,j)) < (i_1, j_1) < \tau_{h_1}$ *mean is investigated and the second threshold* $\tau_{h_2} < \tau_{h_1}$ *using the following formula Obtained:*

$$\tau_{h_2} = mean\left(G\left(i_1, j_1\right)\right) + k \times \text{standard deviation}(G\left(i_1, j_1\right)) \quad (12)$$

If $G(i_1, j_1) > \tau_{h_2}$*, then the point* (i,j)*is identified as a weak edge point.*

4. declration edge points

4.1. *Every strong point of the edge is considered as an edge point.*

4.2. *A weak edge point is considered as an edge if at least some of 8 pixels in the neighboring area are the strong edge point.*

5. Performing morphological operations

The matrix obtained from the edge points using the above algorithm is a matrix with discontinuous and thick edge points. In order to avoid this result, morphological operations must be performed, which can be defined as the combination of two dilation and erosion operations. Thus, the image of the final edge of E is obtained by performing in the first place the thinning operation and finally the linked operation.

Part II:

This block involves extracting the contours and connecting them to each other.

1-contour extraction*: Extract from the edge map of the lines and contours. Each contour is represented by a chain code as follows:*

$$C = P_1, P_2, ..., P_n \quad where\ \mathrm{P}_i = (x_i, y_i), i = 1, 2, ..., n \quad (13)$$

Where the $P_i = (x_i, y_i)$ *is, the i pixel position is in the chain code. The* $P_1 = P_n$ *is for loop contour and* $P_1 \neq P_n$ *for open contour.*

2-Filling and connecting: *For the end of p in Ω, if in its neighborhoods of a given size (for example, 5 × 5) contains the endpoint, the nearest one of the p's is selected as Q and the intervals between p, Q are filled. When P and Q are identical in the contour direction, the filled contour is tagged as a ring contour, and P and Q are removed from it. When P and Q are in two contours, they are connected to an open contour, and P and Q are removed from Ω.*

3-Stretching Open Contours*: If the neighboring end of an open contour contains edge pixels in other contours, it extends to the nearest pixel edge in the other contour, which causes many of the missing corners selected in the detection of edges and contour extraction.*

Part III:

This block contains decision making for the corners through the residual area. The proposed corner detector uses the remaining area as the corner measure to select the corner of each contour, consisting of three stages: 1) Calculation of corner measurement 2) Non-Maximum suppression and 3) Threshold.

1) Calculation of corner measurement*: Assume c = {p1, p2, ..., pn} is a contour extracted. At each pixel on the contour C, calculate the normalized ANDD representation and corner measurement, residual R-area(pi) is calculated as follows:*

$$\hat{\partial}\nabla(n;k) = \frac{\partial\nabla(n;k)}{\max_k\{|\partial\nabla(n;k)|\}}$$

$$k_0 = \arg_k \max\{|\partial\nabla(n;k)|\}$$

$$R - \hat{\partial}\nabla(n;k) \equiv \hat{\partial}\nabla(n;k) - sign\{\hat{\partial}\nabla(n;k_0)\}\xi k_0\Delta\theta, \rho(k\Delta\theta)$$

$$R - area(n) = \frac{\Delta\theta}{2}\sum_{k=1}^{p}(R - \hat{\partial}\nabla(n;k))^2 \quad (14)$$

$$\Delta\theta = \frac{2\pi}{p}$$

$$C = P_1, P_2, ..., P_n$$

$$where\ \mathrm{P}_i = (x_i, y_i), i = 1, 2, ..., n$$

2) Non-maximum suppression: *Only the local maximum of corners in the contour is shown as the candidate corner. Let the width of the window for the non-maximum suppression be 2L+1.a pixel p in the contour c is a candidate corner if:*

$$R - area(P_i) > \max\{R - area(p_{i\pm l}), l = 1, 2, ..., L\} \quad (15)$$

3) Threshold: *is a predefined threshold for corner decision making that is empirically selected. Its value is 0.12. A pixel is considered as a corner if:*

$$R - area(P) \geq \eta \quad (16)$$

2-3. Description of the Krawtchouk polynomial

In the method of detecting the edges of images using polynomials, a new form is obtained to obtain a derivative estimate at each point of the image using a discrete orthogonal polynomial family, called a Krawtchouk polynomial, which are orthogonal to the distribution of binaries [13,15]. In the remainder of this section, some definitions and features of the Monic Krawtchouk polynomial are presented in one variable [15].

Definition1:Let

$N \epsilon \mathbb{N}, \Lambda_N = \{0,1,2,\ldots,N\},\ \alpha \in [-1,1]$*And* $\omega_{N,\alpha}(x)$ *is the weight function.*

$$\omega_{N,\alpha}(x)=\binom{N}{x}\alpha^{x}(1-\alpha)^{N-x},\ for\ all\ x\epsilon\Lambda_N \tag{17}$$

It is said that $k_n^{\alpha}(x,N)=x^n+\cdots$ *with* $n\leq N$, *is the nth monic Krawtchouk polynomial with respect to the pair* $(\Lambda_N,\omega_{N,\alpha})$ *if:*

$$k_n^{\alpha}(.,N),x^j{}_{N,\alpha}=\sum_{i=0}^{N}k_n^{\alpha}(i,N)x^j\omega_{N,\alpha}(i)=0 \tag{18}$$

for all $j=0,1,\ldots,N$, *where* $\langle .,. \rangle_{N,\alpha}=\langle .,. \rangle_{\Lambda_N,\omega_{N,\alpha}}$

Obviously, the binomial theorem is normalized to the weight formula 17.the nth monic Krawtchouk polynomial in one variable can be generated.

$$k_n^{\alpha}(x,N)=\sum_{j=0}^{N}\binom{n}{j}\alpha^{n-j}(1-\alpha)^{j}(x-N)_{n-j}(x-j+1)_j \tag{19}$$

where $(a)_j$ *denotes the Pochhammer symbol or shifted factorial . The normal state is obtained from a polynomial of degree integer in Formula 19.*

$$\left\|k_n^{\alpha}(.,N)\right\|_{N,\alpha}^{2}=\sum_{i=0}^{N}k_n^{\alpha}(i,N)^2\omega_{N,\alpha}(i)=\binom{N}{n}n!^2(\alpha-\alpha^2)^n \tag{20}$$

If f is a function of a variable, the first-order difference is the $\Delta_+f(x)=f(x+1)-f(x)$, $\Delta f(x)=\frac{1}{2}(f(x+1)-f(x-1))$. *So that the difference Δf from the first order is usually the central difference formula on the two nodes [16]. In the function f with two variables, the partial difference of the first order is defined as follows:*

$$\Delta_x f(x,y)=\frac{f(x+1,y)-f(x-1,y)}{2}$$
$$\Delta_y f(x,y)=\frac{f(x,y+1)-f(x,y-1)}{2} \tag{21}$$

The Krawtchouk polynomials are used because they allow closed definitions to be derived for discrete derivatives of polynomials with one or two variables.

For the 2D monic Krawtchouk polynomials, the following difference formulas are as follows [17]:

$$\Delta_x K_{n,m}^{\alpha_1,\alpha_2}(x,y)=\left(\Delta_x k_n^{\alpha_1}(x,N_1)\right)k_m^{\alpha_2}(y,N_2) \tag{22}$$

$$\Delta_y K_{n,m}^{\alpha_1,\alpha_2}(x,y)=\left(\Delta_y k_m^{\alpha_1}(y,N_2)\right)k_m^{\alpha_1}(y,N_1) \tag{23}$$

The standard Theory of the approximation of the function , for $M_1\in\Lambda_{N_1}\backslash\{0\}$ $M_2\in\Lambda_{N_2}\backslash\{0\}$ *is a polynomial of total degree* $(M_1-1)\times(M_2-1)$.

$$P_{M_1,M_2}(x,y)=\sum_{n=0}^{M_1-1}\sum_{m=0}^{M_2-1}\beta_{m,n}K_{n,m}^{\alpha_1,\alpha_2}(x,y) \tag{24}$$

$$\beta_{m,n}=\frac{<I,K_{n,m}^{\alpha_1,\alpha_2}>_{2D}}{<K_{n,m}^{\alpha_1,\alpha_2},K_{n,m}^{\alpha_1,\alpha_2}>_{2D}} \tag{25}$$

So that:

$$min_{Q\in P_{m_1,m_2}}\left\|I-Q\right\|_{2D}=\left\|I-P_{M1,M2}\right\|_{2D} \tag{26}$$

P_{M_1,M_2} *is a polynomial of the least squares I approximation in* P_{M_1,M_2}, *and I (x, y) is approximately equal to* $P_{M_1,M_2}(x,y)$. *[18] Moreover, if* $M_1=N_1+1$ *and* $M_2=N_2+1$, *then* $I=P_{N_1,N_2}$.

2-3-1. Calculate Discrete Derivatives Using Blocks

In order to detect edge points, the whole image of L is analyzed using the block $L_{i,j}$, *with fixed size* $(N_1+1)\times(N_2+1)$. *So that* $n_1<N_1$ *and* $n_2<N_2$. *All blocks are assumed to be of the same size [15]. First, some definitions are mentioned. If* $a_{i,j}$ *is a real matrix of size* $u\times v$, *then* A^T *refers to the transient matrix. And vec (A) is the vector column (u v) and is defined as:*

$vec(A)=(a_{11},\ldots,a_{u1},a_{12},\ldots,a_{u2},\ldots a_{1v},\ldots,a_{uv})^T$

So that: $\overline{a_n}=(a_n(0),\ldots,a_n(n_1)$, $\overline{b_m}=(b_m(0),\ldots,b_m(n_2)$ *respectively two rows of vector with order* (n_1+1) *and* (n_2+1), *so that:*

$$a_n(v)=\frac{k_n^{\alpha_1}(v,n_1)\omega_{n_1,\alpha_1}(v)}{\left\|k_n^{\alpha_1}\right\|_{n_1,\alpha_1}^{2}}$$
$$b_m(v)=\frac{k_m^{\alpha_2}(v,n_2)\omega_{n_2,\alpha_2}(v)}{\left\|k_m^{\alpha_2}\right\|_{n_2,\alpha_2}^{2}} \tag{27}$$

The matrix $C_{n_1,n_2}(n,m)=\overline{a_n^Tb_m}$ *has the size* $(n_1+1)\times(n_2+1)$ *and depends only on the size of the blocks.*

Let $\beta_{m,n}(i,j)$ *be the coefficient given by equation (24)with respect to* $l=l_{i,j}$. *This coefficient can be calculated as follows:*

$$\beta_{m,n}(i,j)=vec\left(L_{i,j}\right),vec\left(C_{n_1,n_2}(n,m)\right)_2 \tag{28}$$

So that $\langle .,. \rangle_2$ *is the result of the distance of the Euclidean inner product on* $R^{(n_1+1)(n_2+1)}$. $B_{m,n}$ *is a matrix with coefficients* $\beta_{m,n}(i,j)$ *with* $i=0,\ldots,n_1$ و $j=0,\ldots,n_2$. *From formula (24) we can conclude that* $B_{m,n}=L*C_{n_1,n_2}(n,m)$. *The * symbol refers to the discrete convolution of the two-dimensional matrix [14]. By using a discrete Fourier transform, the convolution of this matrix can be optimized and the processor time can be significantly improved.*

For each pixel (i,j) *of the image L, the partial derivative of the discrete formula is calculated by formula (21) with the approximation of formula (24). Just keep in mind that the information is stored in neighborhoods* $L_{i,j}$. *When the processing is complete, two matrices* P_x *and* P_y *are obtained with the same size L, such that each cell (i, j) is a partial derivative with respect to x or y. Therefore, this matrix allows for a good estimate of the gradient coefficient at each point (i, j) of the image L. In order to have numerical tests, the parameters of Krawtchouk polynomial are considered as follows:*

$$n_1=n_2=n_t=4,\alpha_1=\alpha_2=\alpha=\frac{1}{2}$$

Discrete derivative calculations are approximated by taking the block information $L_{i,j}$ with a degree of 5 × 5 and with center (i,j):

$$L_{i,j}=\begin{bmatrix} I(i-2,j-2) & I(i-2,j-1) & I(i-2,j) & I(i-2,j+1) & I(i-2,j+2) \\ I(i-1,j-2) & I(i-1,j-1) & I(i-1,j) & I(i-1,j+1) & I(i-1,j+2) \\ I(i,j-2) & I(i,j-1) & I(i,j) & I(i,j+1) & I(i,j+2) \\ I(i+1,j-2) & I(i+1,j-1) & I(i+1,j) & I(i+1,j+1) & I(i+1,j+2) \\ I(i+2,j-2) & I(i+2,j-1) & I(i+2,j) & I(i+2,j+1) & I(i+2,j+2) \end{bmatrix}$$

From formula (24) and taking $M_1 = M_2 = M_t$ and for $i = 0,\dots,N_1$ and $j=0,\dots,N_2$, the polynomial block approximation is as follows:

$$P^{(i,j)}_{M_t,M_t}(x,y) \approx I(i+x-2, j+y-2) \quad (x,y)\epsilon\Lambda_{n_t}\times\Lambda_{n_t} \tag{29}$$

For each constant point (i,j), the polynomial $P^{(i,j)}_{M_t,M_t}(x,y)$ is obtained using formula (24), and the coefficients $\beta_{m,n}(i,j)$ is obtained using formula (28).Now considering the fact that for x = y = 2 in the approximation of formula (29), the point (i, j) is considered as the center of the block $L_{i,j}$, and then the first order partial differences of $P^{(i,j)}_{M_t,M_t}(x,y)$ is calculated using the central difference formula for the Krawtchouk polynomial using formula (3) and formula (22),(23). For example, for $M_t = 2$ it is:

$$\Delta_x P^{(i,j)}_{M_t,M_t}(2,2) = \beta_{1,0}(i,j) - \beta_{1,2}(i,j)$$
$$\Delta_y P^{(i,j)}_{M_t,M_t}(2,2) = \beta_{0,1}(i,j) - \beta_{2,1}(i,j) \tag{30}$$

$$M_t = 4$$

$$\Delta_x P^{(i,j)}_{M_t,M_t}(2,2) = \beta_{1,0}(i,j) - \beta_{1,2}(i,j) + \frac{3}{2}\left(\beta_{3,2}(i,j) - \beta_{3.0}(i,j)\right)$$
$$\Delta_y P^{(i,j)}_{M_t,M_t}(2,2) = \beta_{0,1}(i,j) - \beta_{2,1}(i,j) + \frac{3}{2}\left(\beta_{2,3}(i,j) - \beta_{0,3}(i,j)\right) \tag{31}$$

It is seen from Formulas (30) and (31) that it is not necessary to calculate all matrices $\beta_{m,n}$ for calculating P_x and P_y. in fact, Using Formula (31), the matrices P_x and P_y are obtained[15] and the partial derivative with respect to x and y is respectively equal to:

$$P_x = \beta_{1,0} - \beta_{1,2} + \frac{3}{2}\left(\beta_{3,2} - \beta_{3.0}\right)$$
$$P_y = \beta_{0,1} - \beta_{2,1} + \frac{3}{2}\left(\beta_{2,3} - \beta_{0,3}\right) \tag{32}$$

To ensure that all image boundary pixels are examined, missing pixels are replaced within the Convolution operation. By reversing the values inside the boundaries of the image L, they are obtained against the boundary pixels.

3-Experimental results

In this section, the datasets used for the tests are first introduced. The proposed method is then examined using the proposed criteria such as precision. Finally, the result of applying the algorithm is compared with the method ANDD.

*To evaluate the proposed method, a dataset called 'Image Database and Corner Detection' and some standard images that are commonly used to evaluate corner detection in papers are used[1]. Each of the images has dimensions 256 * 256 and 512 * 512 or different dimensions and gray. In the following, criteria for evaluation on the data set are used and their results are presented in the form of a table. In the following, a sample of the images used by the dataset brought.*

Figure 4- a sample of dataset images

The outputs of the algorithm of the proposed method and ANDD method on the data set are given in the following figure.

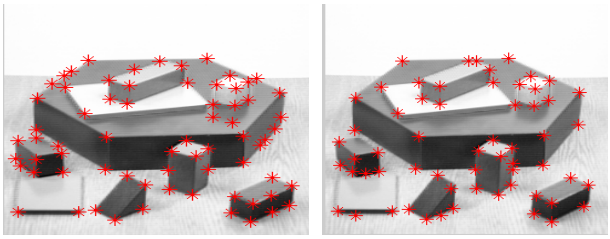

Figure 5- shows this example of the output image of the data set. The images on the left are related to the ANDD method and the right images of the proposed method.

3-1. evaluation criteria

In this section, the precision of the proposed algorithm is evaluated. In order to assess the qualities of the corners, the following criteria are used:

One of the criteria that can be used to evaluate the proposed method is to check the Precision of these two algorithms under the following equation:

$$precision = \frac{TP}{TP+FP} \tag{33}$$

TP: Points that are correctly identified (points that were corners and correctly identified).

FP: Points that are incorrectly identified are correct (points not cornered but mistakenly identified by the corner)

The results of the evaluation of precision on the set of images examined are given in the following table.

[1] http://www.dabi.temple.edu/~shape/MPEG7/dataset.html

Table 1. Evaluation of the average precision of the proposed method and ANDD method

images	Precision of proposed method	Precision of ANDD
image1	1	0.66
image2	0.6	0.53
image3	0.41	0.39
image4	0.4	0.36
image5	0.6	0.55
image6	0.43	0.21
image7	0.19	0.12
image8	0.29	0.27
image9	1	0.56
image10	1	0.66
image11	0.9	0.52
image12	0.8	0.29
image13	0.92	0.66
image14	1	0.76
image15	1	0.54
image16	1	0.5
image17	1	0.75
image18	1	0.7
image19	1	0.29
image20	0.8	0.76
Average precision	**0.767**	**0.504**

As shown in the table above, the average precision of the proposed method is better than the ANDD method, which indicates the superiority of the proposed method.

4. Conclusion

In this article, firstly, the study of studies on the extraction of corner points in the articles was discussed and the methods presented are classified into two general categories, based on the contour and based on the intensity. Comparisons between these two methods show the superiority of the contour-based methods. For this reason, the contour-based method is the basis for the study of this paper. Subsequently, the problem was presented conceptually and accurately with scientific definitions, taking into account the basic concepts. To solve the problem, the idea of using the Krawtchouk polynomial was proposed.

First, a series of concepts were expressed in relation to the proposed algorithm. Then each of the steps in the algorithm is fully explained. In the proposed method, cravings were used to obtain the image edges of a Krawtchouk polynomial. The evaluation of the experimental results between the proposed method and the method ANDD shows the superiority of the proposed method in terms of precision.

Reference

[1] D. Parks and J.-P. Gravel, "Corner detection", International Journal of Computer Vision, 2004.

[2] G. Xinting, Z. Wenbo, F. Sattar, R. Venkateswarlu, and E. Sung, "Scale-space Based Corner Detection of Gray Level Images Using Plessey Operator," in Proc. of the Fifth International Conference on Information, Communications and Signal Processing, pp. 683-687, 2005.

[3] G. Xinting, F. Sattar, and R. Venkateswarlu, "Multiscale Corner Detection of Gray Level Images Based on Log-Gabor Wavelet Transform," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, pp. 868-875, 2007.

[4] C. Harris and M. Stephens, "A combined corner and edge detector," in Proceedings of the 4th Alvey Vision Conference, pp. 147–151, 1988.

[5] A. Willis and S. Yunfeng, "An algebraic model for fast corner detection," in Proc. of the IEEE 12th International Conference on Computer Vision, pp. 2296-2302, 2009.

[6] S. M. Smith and J. M. Brady, "Susan - a new approach to low level image processing," International Journal of Computer Vision, vol. 23, no. 1, pp. 45–78, 1997.

[7] Z. Li-hui, C. Jie, Z. Juan, and D. Li-hua, "The Comparison of Two Typical Corner Detection Algorithms," in Proc. of the Second International Symposium on Intelligent Information Technology Application, pp. 211-215, 2008.

[8] L.Zhang,D.Zhang," A Multi-scale Bilateral Structure Tensor Based Corner Detector", Springer-Verlag Berlin Heidelberg, 2010.

[9] J.Zhang,Q.chen," A highly repeatable feature detector: improved Harris–Laplace", Multimed Tools Appl 52:175–186, 2011.

[10] B. Sirisha and B. Sandhya, "Evaluation of distinctive color features from harris corner key points," in Proc. of the 2013 IEEE 3rd International Advance Computing Conference, pp.1287-1292, Feb. 2013.

[11] PL Shui, WC Zhang," Noise-robust edge detector combining isotropic and anisotropic Gaussian kernels", Pattern Recognition 45 (2), 806-820, 2012.

[12] PL Shui, WC Zhang," Corner detection and classification using anisotropic directional derivative representations", IEEE Transactions on Image Processing 22 (8), 3204-3218, 2013.

[13] S. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach, NY, 1978.

[14] J.H. Mathews, K.K. Fink, Numerical Methods Using MATLAB, Prentice-Hall Inc., Upper Saddle River, NJ, 1999

[15] PA Daniel Rivero-Castillo , Héctor Pijeira, Edge Detection Based on Krawtchouk Polynomials, Computational and Applied Mathematics 284, 244-250, 2015.

[16] G. Szego, Orthogonal Polynomials, in: Amer. Math. Soc. Colloq. Publ., vol. 23, Providence, RI, 1975.

[17] J.R. Rice, The Approximation of Functions, Vol. 2: Nonlinear and Multivariate Theory, Addison-Wesley Publ. Co., Reading, Mass., 1969.

[18] G. Hoggar, Mathematics of Digital Images, Cambridge Univ. Press, Cambridge, 2006.

[19] F. Mokhtarian, F Mohanna, "Performance evaluation of corner detectors using consistency and accuracy measures", in Computer Vision and Image Understanding,Volume 102, Issue 1, Pages 81-94, April 2006.

[20] H. P. Moravec, "Toward automatic visual obstacle avoidance," in Proc. 5th Int. Joint Conf. Artif. Intell., Cambridge, MA, USA, p. 584, 1977.