Доклади на Българската академия на науките
Comptes rendus de l'Académie bulgare des Sciences

Tome 75, No 8, 2022

MATHEMATICS
Artificial intelligence

DIGITAL IMAGE FILTERING WITH NEW GENERALIZED ORDER-*k* JACOBSTHAL NUMBERS

Hossen Mohammed Mahdi Ahmed, Ahmet Daşdemir*

Received on March 27, 2022
Presented by P. Popivanov, Member of BAS, on May 31, 2022

Abstract

In this paper, a new generalized order-k Jacobsthal and Jacobsthal–Lucas numbers are investigated and used in the digital image processing for filtering of the image. The Gaussian, Morphological operation with "Motion", Morphological operation with "Disk" specification, Poisson method and Jacobsthal method is used for the filtering. The Structural similarity, Multiscale structural similarity and Peak SNR are used for evaluation of the results. For implementation of the proposed method the Matlab 2021a version is used.

Key words: generalized order-*k* Jacobsthal, Jacobsthal–Lucas numbers, image processing, filtering

2020 Mathematics Subject Classification: 11B37

1. Introduction. Edge detection in digital images is a type of image processing that includes various mathematical methods developed to determine the sharp changes in pixel values, or in other words, where the continuity is lost in the image. These sharp changes in the image form typical clusters of curves and lines called edges. Edge detection in two-dimensional images is a type of two-dimensional signal processing. In fact, edge detection in the image is a similar issue to change detection, which is defined as finding points that distort continuity, known as step detection in one-dimensional signals, or finding points where the signal shows discontinuity across time. Edge detection emerges as a basic tool in the fields of image processing, machine vision, computer vision, and especially feature detection and feature extraction [1].

DOI:10.7546/CRABS.2022.08.02

Finding places that show discontinuity in the image aims to detect the important aspects of the image data and the changes in the features in the image space.

The following reasons can be listed in the context of general acceptance as causing discontinuity in the image:

1. Discontinuities in depth.
2. Discontinuities in surface orientation.
3. Changes in material properties.
4. Lighting differences in the image [2,3].

Ideally, applying edge detection to the image would result in clusters of combined curves showing the edges of objects or discontinuities in the surface orientation. In this way, the edge detection process can filter the significant amount of unnecessary data, which is far from serving the purpose, while preserving the important structural features present in the image. Getting the desired output in the edge detection stage is a step that will contribute to the continuation of the progressive data processing more easily. However, in real life, it is not always possible to detect such ideal edges, even in images with average complexity. Therefore, edge detection methods, which have important areas of use, should be focused on and the existing techniques should be improved.

Commonly used and well-known edge detection techniques include Canny, Prewitt, Sobel and Roberts Cross edge detection methods. Each of them can be used in areas such as feature detection, feature extraction, computer vision by optimizing with other edge detection algorithm components, according to the usage area and the image properties subject to processing.

Edge detection algorithms are created by using different components together. These components serve the purpose of detecting the edges of the image to be processed correctly, as well as preventing the image contents that do not have edge feature from being detected as edges. It is possible to list the components of the algorithms discussed in this study as follows:

1. Noise cancelling pre-filtering.
2. Edge detection masking (detection of discontinuity between pixel neighbourhoods).
3. Thresholding.
4. Thinning.

De-noise pre-filtering is applied to remove as much as possible the noise signal existing on the image before the post-processing stages of the digital image. At this stage, while preserving the data of the edge patterns in the image, it is aimed to detect both the defective edge and the noise signals that may cause the existing edge to be detected.

In the phase of edge detection masking based on differential value determination over pixel neighbourhoods, sharp changes in the image or in other words discontinuities are tried to be detected, and the pixel values of the edge coordinates are weighted according to the severity of the change in the image.

In the thresholding phase, the threshold value determined according to the weighted pixel values or pixels with values below a certain size according to the threshold value set are considered not edge coordinates, and the edge detection image consisting of pixels with binary values is obtained.

It may be preferred to use the thinning step according to the application area or not. However, for application areas such as feature detection or extraction where template matching operations are performed, refining stage is generally preferred as it provides ease of operation and better accuracy. At this stage, the thickness of the template determined as the edge is thinned to a single pixel thickness.

The details of the steps mentioned above are given below under separate headings.

2. Noise filtering. In order to increase the accuracy of the edge detection process, noise filtering is applied as a pre-treatment to remove the noise in the image as much as possible. The nature of the noise is an important parameter in filter selection. For example, if the noise in the image is in high frequency feature, it is purified with a low pass filter [4].

Various filters are used for noise reduction in images. Average filter and median filter are frequently used filters for noise reduction [5]. The Gaussian filtering method is among the effective filtering methods that are used in the cleaning of noises, whose exact form is unknown but assumed to be in normal distribution [6]. Information on these three different filter types listed above is given below.

2.1. Median filter. Median filter is a nonlinear digital filtering technique that is widely used to eliminate noise on the image or signal. It has a pre-treatment feature that is applied to clean the noise in the image or signal before further processing. It has a wide range of applications, especially in image processing applications such as edge detection, since it has the ability to accurately remove noise while preserving edge data under certain conditions. The basic technique of using the median filter in digital images is to create a subset of this pixel for each pixel and the values of neighbouring pixels equidistant to it, and to determine the middle value in this cluster according to the order of magnitude [7–9]. In order to explain the process of determining the new value with median filtering, one $I(x, y)$ pixel and eight neighbouring pixels of this pixel are discussed below (Fig. 1).

Figure 1 shows the gray level values of the $I(x, y)$ pixel in the I image and neighbouring pixels in primary proximity. Here, a subset A, in which the said pixel values are considered the element for the median filtering process, is formed as $A = \{120, 130, 135, 200, 230, 233, 238, 242, 253\}$. When the nine elements of this

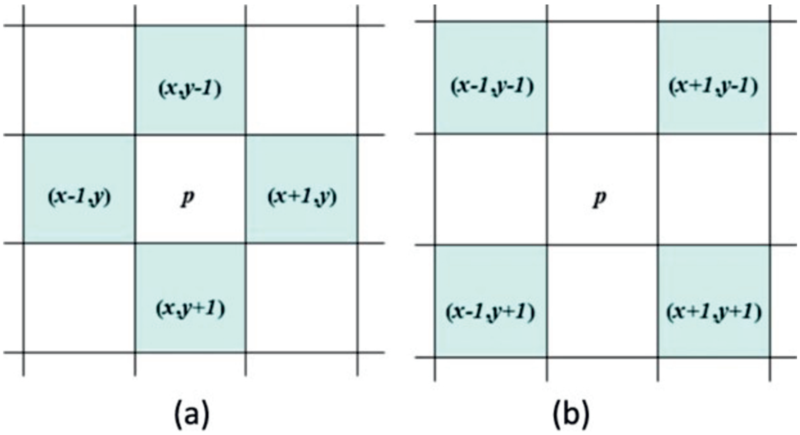


Fig. 1. $I(x, y)$ pixel and primary neighbourhoods of the I image [10]

subset are sorted according to their size, the fifth value 230, which is the median value, is assigned as the new value of the $I(x, y)$ pixel after filtering.

2.2. Average filter. Mean filter, as its name suggests, is a type of filtering in which the arithmetic average of the pixel values subject to the process is taken. Today, it is preferred in many applications as it is an easy and at the same time effective noise softening tool in most conditions. The neighbourhood approach used is similar to the median filter. If the average filter is applied for the above example, the arithmetic average of the elements of the set A will be taken and as a result, the value of approximately 198 will be obtained as the new value of the $I(x, y)$ pixel as a result of the filtering.

The Gaussian filter is a two-dimensional convolution operator used to remove noise from digital images. Although it resembles the average filter in this respect, thanks to the Gaussian distribution in the form of a bell curve, less weight is given to distant neighbourhoods, making it possible to achieve more accurate results in normally distributed noise, one of the most common types of noise, as well as positive results in other noise forms. In addition, it is less affected by the downsides of sharp noise forms than the average filter [11]. The Gaussian function applied to each pixel in the digital image and which is the impulse response of the filter in question is expressed as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{1}$$

where x and y are cartesian plane coordinates, while σ denotes standard deviation.

2.3. Masking with the edge detection operator. This subsection is about the stage, where edge detection, which gives the name to the most basic process of edge detection, is performed directly. In the scope of the study, since the edge detection methods known as Canny, Prewitt, Sobel and Roberts Cross will be focused on, the differential value detection methods calculated on the pixel neighbourhoods related to them will be mentioned.

The basic approach in these methods is to analyze the discontinuity in the image or sharp changes in the pixel values in order to detect the important aspects of the image data and the changes in the image space features. In order to determine the changes in question, the image is processed with the differential value detection approach that predicts the difference in value according to neighbouring pixels, separately for each pixel. For this, in these methods, unique masks are used for each of them in order to detect the change between neighbouring pixels. These masks are applied to each pixel in the image using other pixel values in the neighbourhood of the pixel. Thus, an edge detection image with the same gray colour level values as the image used as input at this stage is obtained. However, in these edge detection methods, there is a pair of masks to be applied separately for each of the x and y axes in the coordinate plane. Both of the pair of masks used in determining the differential value in both axes in the image form the mask for edge detection method. Then, from the differential values calculated separately for the two axes, a single differential value is determined for the relevant pixel. In other words, the magnitude of the combination of two vectors in the same direction with the axes is calculated.

For a P_0 pixel on the image, the values obtained after masking in the x-axis and y-axis directions are D_x and D_y, respectively, and the gradient value D (edge detection value) calculated for the $I(x, y)$ pixel in the I image is given below:

$$\nabla I(x, y) = \begin{bmatrix} D_x \\ D_y \end{bmatrix}, \tag{2}$$

$$D = \|\nabla I(x, y)\|, \tag{3}$$

$$D = \sqrt{D_x^2 + D_y^2}. \tag{4}$$

Here ∇ is the gradient operator, $\nabla I(x, y)$ is the gradient of the image I at x and y location, and $\|\cdot\|$ is the Euclidian norm. The mask forms of these techniques are explained as follows.

2.4. Canny edge detection mask. Canny edge detection is an edge detection operator developed by CANNY in 1986 [12]. The 2×2 masking used in this method can be specified in the form of x and y axis components as in Fig. 2.

The calculation of the gradient value pair for the Canny edge detection masking process for an $I(x, y)$ pixel on an I image is as follows:

$$D_x = I(x, y) - I(x + 1, y) - aI(x + 1, y + 1), \tag{5}$$

$$D_y = I(x, y) - I(x, y + 1) - aI(x + 1, y + 1). \tag{6}$$

While the value of a here can be zero, changes can be made to this constant according to the application area. Although there are many modification studies

1	−1
0	−a

a)

1	0
−1	−a

b)

Fig. 2. Canny edge detection masking demonstration:
a) x-axis mask, b) y-axis mask

for the Canny edge detection technique, a value of $a = 0.707 \approx 1/\sqrt{2}$ is generally preferred.

2.5. Roberts Cross edge detection mask. Roberts Cross edge detection is one of the first edge detection methods and was first introduced by Lawrence Roberts in 1963. The most attractive aspect of this method is its simplicity, that is, it has a mask structure that is small in size and only takes an integer value. However, with the improvements in computer speeds, this advantage has lost its importance and has become negligible, and this method has the disadvantage of being very sensitive about being affected by noise [13]. However, this method can still be used with some modifications.

The pair of masks belonging to the Roberts Cross edge detection method is given in Fig. 3.

The calculation of the gradient value pair for the Roberts Cross edge detection method for the $I(x, y)$ pixel on an I image is as follows:

$$D_x = I(x, y) - I(x + 1, y + 1), \tag{7}$$

$$D_y = I(x + 1, y) - I(x, y + 1). \tag{8}$$

1	0
0	−1

a)

0	1
−1	0

b)

Fig. 3. Roberts Cross edge detection masks:
a) x-axis mask, b) y-axis mask

2.6. Prewitt operator mask. The Prewitt operator is used in the detection of edge patterns in digital images in a wide range of applications. The difference from the Roberts Cross edge detection is the dimensions and forms of the masks used. In this method, 3×3 size masks are used, based on multiplying the element in the middle of the mask geometrically with the pixel value whose edge value will be calculated.

Calculation of the gradient value pair of Prewitt edge detection method for $I(x, y)$ pixel on an I image is as follows:

$$\begin{aligned} D_x = {} & I(x+1, y-1) + I(x+1, y) + I(x+1, y+1) \\ & - I(x-1, y-1) - I(x-1, y) - I(x-1, y+1), \end{aligned} \tag{9}$$

$$\begin{aligned} D_y = {} & I(x-1, y-1) + I(x, y-1) + I(x+1, y-1) \\ & - I(x-1, y+1) - I(x, y+1) - I(x+1, y+1). \end{aligned} \tag{10}$$

2.7. Jacobsthal numbers. Let us define k-sequence of order-k generalization (KSOKJ) as follows: For $n > 0$, $1 \leq i \leq k$

$$J_n^i = J_{n-1}^i + J_{n-2}^i + \cdots + J_{n-k+1}^i + 2J_{n-k}^i \tag{11}$$

with initial conditions for $1 - k \leq n \leq 0$

$$J_{k,n}^i = \begin{cases} 1 & \text{if } i + n = 1, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

where J_n^i is the n-thB term of the i-th sequence for k=2 and i=1.

The generalized order-k Jacobsthal sequence is reduced to the usual Jacobsthal sequence.

By the definition of generalized Jacobsthal numbers, we can write the following Vector Recurrence relation

$$\begin{bmatrix} J_{n+1}^i \\ J_n^i \\ J_{n-1}^i \\ \vdots \\ J_{n-k+2}^i \end{bmatrix} = A \begin{bmatrix} J_n^i \\ J_{n-1}^i \\ J_{n-2}^i \\ \vdots \\ J_{n-k+1}^i \end{bmatrix}, \tag{13}$$

where A is a k-square companion matrix as follows:

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 2 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{14}$$

Let us define a k-square matrix $J_n^{\sim} = [J_{ij}]_{k.k}$ to deal with the k-sequences of the generalized order-k Jacobsthal numbers, as

$$J_n^{\sim} = \begin{bmatrix} J_{k,n}^1 & J_{k,n}^2 & \cdots & J_{k,n}^k \\ J_{k,n-1}^1 & J_{k,n-1}^2 & \cdots & J_{k,n-1}^k \\ \vdots & \vdots & \ddots & \vdots \\ J_{k,n-k+1}^1 & J_{k,n-k+1}^2 & \cdots & J_{k,n-k+1}^k \end{bmatrix}. \tag{15}$$

3. Evaluation criteria. For evaluation of the proposed method the Structural similarity (SSIM), Multiscale structural similarity (MS-SSIM) and Peak SNR are examined.

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2. \tag{16}$$

The PSNR is illustrated in the following equation

$$\text{PSNR} = 10 \cdot \log_{10}\frac{\text{MAX}_I^2}{\text{MSE}} = 20 \cdot \log_{10}\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} = 20 \cdot \log_{10}\text{MAX}_I - 10 \cdot \log_{10}\text{MSE}. \tag{17}$$

The SNR is shown in thye following equation

$$\text{SNR}_{\text{dB}} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right), \tag{18}$$

where

$$P_{\text{signal}} = \sum_{i=1}^{m}\sum_{j=1}^{n}(I(i,j))^2 \tag{19}$$

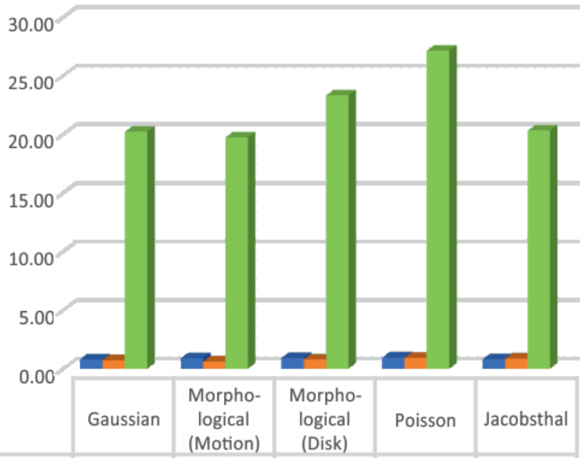
and

$$P_{\text{Noise}} = \sum_{i=1}^{m}\sum_{j=1}^{n}(N(i,j))^2. \tag{20}$$

The name of this equation is Parseval's Equation.

The Structural similarity (SSIM), Multiscale structural similarity (MS-SSIM) and Peak SNR results are graphically shown in Fig. 4.

4. Conclusion. In mathematics, the Jacobsthal numbers are an integer sequence named after the German mathematician Ernst Jacobsthal. Like the related Fibonacci numbers, they are a specific type of Lucas sequence $U_n(P, Q)$ for which $P_0 = 2$, $P_1 = 1$, are defined by a similar recurrence relation: in simple terms, the sequence starts with 0 and 1, then each following number is found by adding the number before it to twice the number before that. The first Jacobsthal numbers



	Gaussian	Morpho-logical (Motion)	Morpho-logical (Disk)	Poisson	Jacobsthal
Structural similarity (SSIM)	0.80	0.90	0.93	0.95	0.83
Multiscale structural similarity (MS-SSIM)	0.74	0.62	0.80	0.93	0.86
Peak SNR	20.20	19.73	23.32	27.13	20.30

Fig. 4. The Structural similarity (SSIM), Multiscale structural similarity (MS-SSIM) and Peak SNR results shown graphically

are: 0, 1, 1, 3, 5, 11, 21, 43, 85, 171, 341, 683, 1365, 2731, 5461, 10 923, 21 845, 43 691, 87 381, 17 4763, ... In this paper a new generalized order-k Jacobsthal and Jacobsthal–Lucas numbers are investigated and used in the digital image processing for filtering of the image. The Gaussian, Morphological operation with “Motion”, Morphological operation with “Disk” specification, Poisson method and Jacobsthal method are used for the filtering. The Structural similarity, Multiscale structural similarity and Peak SNR are used for evaluation of the results. For implementation of the proposed method the Matlab 2021a version is used.

REFERENCES

[1] Umbaugh S. E. (2010) Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, CRC Press.

[2] Barrow H. G., J. M. Tenenbaum (1981) Interpreting line drawings as three-dimensional surfaces, Artif. Intell., **17**(1–3), 75–116.

[3] Xiang Y., F. Wang, L. Wan, H. You (2017) SAR-PC: Edge detection in SAR images via an advanced phase congruency model, Remote Sens., **9**(3), 209.

[4] Alizadeh M., M. M. Moghaddam, S. H. HosseinNia (2021) A novel zero delay low pass filter: Application to precision positioning systems, ISA Trans., **111**, 231–248.

[5] Olsen S. I. (1993) Estimation of noise in images: An evaluation, CVGIP Graph. Model. Image Process., **55**(4), 319–323.

[6] Shin D.-H., R.-H. Park, S. Yang, J.-H. Jung (2005) Block-based noise estimation using adaptive Gaussian filtering, IEEE Trans. Consum. Electron., **51**(1), 218–226.

[7] Huang T., G. Yang, G. Tang (1979) A fast two-dimensional median filtering algorithm, IEEE Trans. Acoust., **27**(1), 13–18.

[8] Arias-Castro E., D. L. Donoho (2009) Does median filtering truly preserve edges better than linear filtering?, Ann. Stat., **37**(3), 1172–1206.

[9] Arce G. R. (2005) Nonlinear Signal Processing: A Statistical Approach, John Wiley & Sons.

[10] Alvarez-Sanchez T., J. A. Alvarez-Cedillo, J. Sandoval-Gutierrez (2018) Many-core parallel algorithm to correct the gaussian noise of an image, In: Int. Conf. Supercomputing in Mexico, 70–86.

[11] Hall M. (2007) Smooth operator: Smoothing seismic interpretations and attributes, Lead. Edge, **26**(1), 16–20.

[12] Canny J. (1986) A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell., **PAMI-8**(6), 679–698.

[13] Davis L. S. (1975) A survey of edge detection techniques, Comput. Graph. Image Process., **4**(3), 248–270.

Statistics Department
Sebha University
Sebha, Libya
e-mail: `alremal68@gmail.com`

**Department of Mathematics*
Kastamonu University
Kastamonu, Turkey
e-mail: `ahmetdasdemir37@gmail.com`