IJE TRANSACTIONS B: Applications Vol. 35, No. 08, (August 2022) 1588-1598

International Journal of Engineering

Journal Homepage: www.ije.ir

A New Framework for Canny Edge Detector in Hexagonal Lattice

M. Firouzi[a], S. Fadaei*[a], A. Rashno[b]

[a] *Department of Electrical Engineering, Faculty of Engineering, Yasouj University, Yasouj, Iran*
[b] *Department of Computer Engineering, Engineering Faculty, Lorestan University, Khorramabad, Iran*

PAPER INFO

Paper history:
Received 05 March 2022
Received in revised form 25 April 2022
Accepted 27 April 2022

Keywords:
Edge Detection
Square Lattice
Hexagonal Lattice
Hexagonal Gradient

ABSTRACT

Feature extraction is widely used in image processing applications such as face recognition, character recognition, fingerprint identification and medicine. Edge features is among the most important features for such applications. Canny edge detector is the most popular one and has many benefits in comparison with other methods. Since pixels in hexagonal domain have many benefits in comparison with square domain, this paper presents an efficient Canny edge detector in hexagonal domain. The proposed method includes square to hexagonal transformation and edge detection based on a new algorithm. The proposed method has been evaluated on synthetic and real image datasets with different signal to noise ratios (SNRs). Detected edges in synthetic images show that the proposed hexagonal edge detector outperforms existing methods in 44 cases out of 60 cases with respect to figure of merit (FoM). Finally, results of real images demonstrate the superiority of the proposed method in qualitative analysis of sub-images.

doi: *10.5829/ije.2022.35.08b.15*

1. INTRODUCTION

Image processing can be helpful and plays an efficient role to solve problems such as medical field, robot vision, video processing and pattern recognition. To process, save, represent and translate an image, the first step is image sampling. As shown in Figure 1, there are three tessellation schemes with no overlaps and no gaps including hexagons, squares and regular triangles [1, 2]. Other tessellation methods lead to either overlap or gap between adjacent pixels.

Based on Nyquist frequency, a band-limited signal can be reconstructed using its samples if the sampling frequency is equal or greater than twice the highest frequency of the signal. The hexagonal sampling is the best sampling scheme for band-limited 2D signals. In comparison to other methods, it requires fewer samples to reconstruct original signal. Therefore, the hexagonal sampling needs 13.4% lower samples in comparison with square sampling [3].

Lower samples in hexagonal lattice leads to higher efficiency in sampling. The sampling efficiency in hexagonal domain is 90.7% compared with 78.5% in square domain [4]. The sampling efficiency is the ratio of nonzero area spectrums of the signal over the total area of a period of signal spectrum. On the other hand, quantization is inevitable in sampling and it is proven that for a specific resolution of sensors, the quantization error in hexagonal sampling is lower than square one [5, 6]. Another important issue that makes hexagonal lattice better than other grids is the consistent connectivity property. Neighboring pixels of a central pixel have the same distance from that central pixel. This property in hexagonal lattice makes image processing algorithms on this grid simpler and more efficient. For example, the thinning algorithm on this network is simpler because fewer connection modes are involved in the calculations [7-9].

Due to the advantages of hexagonal network, the implementation of image processing algorithms in this

*Corresponding Author Institutional Email: s.fadaei@yu.ac.ir (S. Fadaei)

Please cite this article as: M. Firouzi, S. Fadaei, A. Rashno, A New Framework for Canny Edge Detector in Hexagonal Lattice, ***International Journal of Engineering, Transactions B: Applications***, Vol. 35, No. 08, (2022) 1588-1598

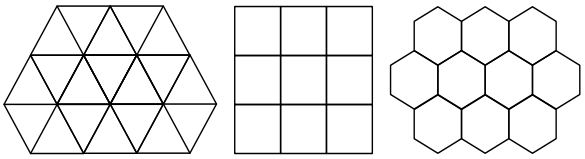


Figure 1. Three tessellation scheme; (a) regular triangles, (b) squares and (c) hexagons

network leads to better results, so researchers try to implement image processing algorithms on hexagonal lattice. A new framework for hexagonal image processing using hexagonal pixel-perfect approximations in subpixel resolution is proposed to convert square pixels to hexagonal one [10]. Hexagonal grid in image processing is studied and its advantages such as higher circular symmetry, equidistant, smaller quantization error, greater angular resolution have been investigated [11]. An approach based on bilinear interpolation was used to convert square image to hexagonal one [12]. A complete study of hexagonal image processing was provided [13, 14]. It was mentioned that arrangement of photoreceptors in the human retina is hexagonal [14]. Image compression based on wavelet was performed on both square and hexagonal images and their performances were compared using two criteria Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE). The implementation results showed that compression in hexagonal lattice is better than square one [14].

The construction of symmetric FIR filter banks for image processing on hexagonal lattice was investigated [15]. A new filter was introduced to denoise hexagonal images and it was shown that results in hexagonal lattice is better than square one [16]. Hexagonal lattice provides 13.4 percent fewer samples than square lattice [17]. A new algorithm was proposed to convert square to hexagonal image and it was shown that variable grids occur in Hexagonal Discrete Fourier Transform (HDFT) [17]. Two-dimensional Modulation Transfer Function (MTF) model of hexagonal lattice detector was deduced based on its definition. MTF is the tool widely used to quantify the efficiency of an electro-optical imaging system [18].

A new approach was proposed to design operators for hexagonal image processing and by developing these operators, a framework for obtaining large-scale neighborhood operators and edge maps at different scales was presented [19]. A new addressing for hexagonal imaging was proposed to simplify implementation and keeps all excellent properties of hexagonal grid [20]. An approach for corner detection on hexagonal images presented and it is shown that its accuracy is comparable to corner detectors applied to square images [21]. Essential operators of translation and rotation were applied to hexagonal images [22]. The computational complexity of edge detection on hexagonal image was reduced using an edge map pyramid approach [23]. Several algorithms were presented for uniform image separation based on virtual hexagonal structure are presented [24].

Feature extraction is very helpful in many applications of image processing such as face recognition, character recognition, content-based image retrieval (CBIR), and medical diagnosis [25-29]. Indeed, to reduce the number of required resources to describe an image, feature extraction is done. Edge detection is one of the most important and widely used feature extraction algorithms and several methods such as Middleton, Staunton, Li and Davies have proposed [30-33]. He et al. [34] has converted the image to hexagonal lattice using virtual hexagonal structure method and edge detection was applied to hexagonal image. The edge detection operator was done through three steps: filtering of noise by a bilateral filter, edge detection using Sobel operator and edge thinning using thresholds. Converting from square to hexagonal lattice was implemented and then the edges of hexagonal image was extracted using Canny edge detector [35].

Edge detection on hexagonal and square image was compared and it was proven that the computational complexity of hexagonal lattice is less than square lattice as well as better qualitative performance in hexagonal lattice [30]. Using the finite element framework, a new method to develop multiscale gradient operators in hexagonal lattice is presented [36]. Some binary morphological operators on hexagonal lattice were introduced to manipulate edge detection by Mostafa et al. [37]. The proposed operators by Mostafa et al. [37] showed that combination of 0˚, 60˚ and 120˚ directional structuring elements on hexagonal grid gives better results than square grid. An approach for hexagonal gradient operators was developed on hexagonal grid within finite element framework [38]. Square lattice was converted to hexagonal domain by two-dimensional to one-dimensional interpolation transformation [39]. Hexagonal image was created from square image using hexagonal deep neural networks [40]. As one of hexagonal advantages, less computational cost of hexagonal Gabor filter in comparison with square one was proven by Varghese and Saroja [41]. Hexagonal domain was developed to use in CNNs with hexagonal convolution defination [42]. Hexagonal lattice was also investigated for better angular symetry, isotropy and consistent connectivity by singular value decomposition definition in hexagonal domain [43]. Finally, bilateral filter was developed for hexagonal edge detection by He et al. [44].

Although hexagonal lattice is the best tessellation scheme, approximately all the edge detection algorithms have been introduced in square lattice. Since the edge

detection plays important role in image processing and its precision and quality can affect results of image processing applications, a hexagonal edge detection method is proposed and investigated in this paper. The proposed hexagonal edge detector is a pure extension of Canny edge detector in square lattice. For this task, images are first transferred from square to hexagonal lattice. Then, gradient magnitude and direction of pixels in hexagonal domain are calculated. Hexagonal level quantization and non-maximum suppression are presented in the next step. Then, thresholds selection and double thresholding are applied to edges. Finally, Edge tracking is done by hysteresis and untrue edges are removed. The organization of this paper is as follows: Section 2 presents Motivation of this research. Proposed method with details is presented in section 3. Implementation results is reported in section 4 and the paper is concluded in section 5.

2. MOTIVATION

Many advantages of image processing applications and operators in hexagonal lattice have been reported in the literature. As stated in our previous framework for hexagonal image processing [10], benefits such as perfect hexagonal shape, accurate intensity level of hexagonal pixels and high resolutions in hexagonal space can be used in real applications. These benefits also revealed that hexagonal lattice has better angular resolution 60 in comparison with 90 in square domain which leads to better edge detection results [10]. Since edge detection operator is among the most significant operators for industrial image processing techniques [34], mentioned advantages of hexagonal lattice motivated us to present an edge detector in this domain. Canny is a conventional edge detector in many applications, therefore a framework for transferring Canny edge detector from square to hexagonal lattice is proposed in this research and referred as hexagonal Canny edge detector.

3. PROPOSED METHOD

In one side, edge detection is an important topic in image processing and among edge detection methods, Canny operator is the most popular one which is widely used in the literature. On the other side, researches have shown that the image processing in hexagonal grid leads to better results than square grid. In this paper, an optimized hexagonal Canny edge detection method is proposed. The flowchart of the proposed method is shown in Figure 2.

Images are first converted to hexagonal grid followed by hexagonal image filtering by a gaussian filter. Then, magnitude and direction of gradients are computed in three axis *x*, *y* and *z*. In the next stage, the non-maximum suppression is applied to the magnitude and direction of gradients. Finally, threshold selection, double thresholding, untrue edge tracking and edge removing are done sequentially to reach edges in hexagonal domain.

3. 1. Square to Hexagonal Transformation

Since the edge processing should be applied to hexagonal image, the square images are first transferred to hexagonal image. To do this, several algorithms have been proposed in the literature. The most significant method is that each square pixel is divided into 7×7 sub-pixels and hexagonal pixels are composed from square sub-pixels [11]. The intensity of sub-pixels is equal to the intensity of the original square pixel. This division leads to a new virtual image, 49 times larger than original image. Finally, each hexagonal pixel is formed from averaging on 56 sub-pixels of the virtual image as shown in Figure 3.

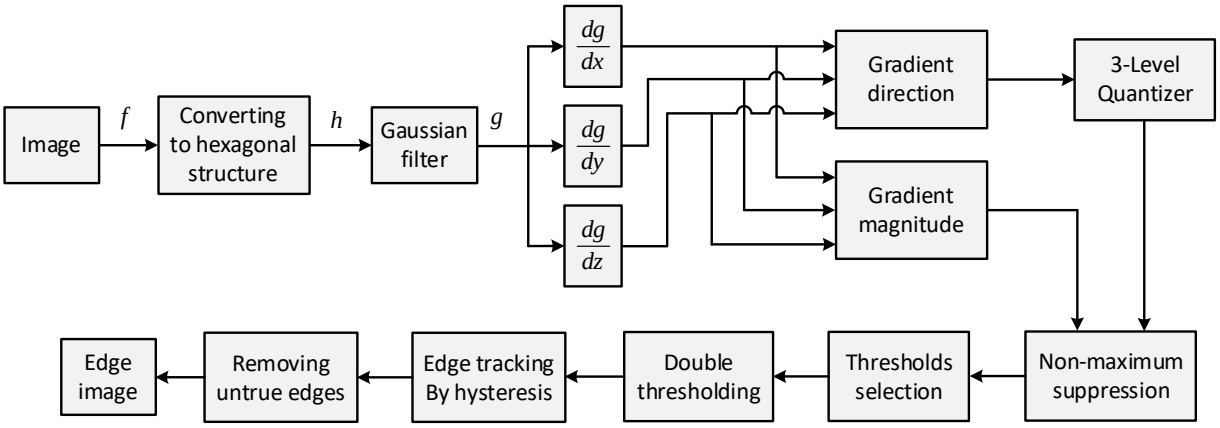


Figure 2. Proposed Canny edge detector in hexagonal domain

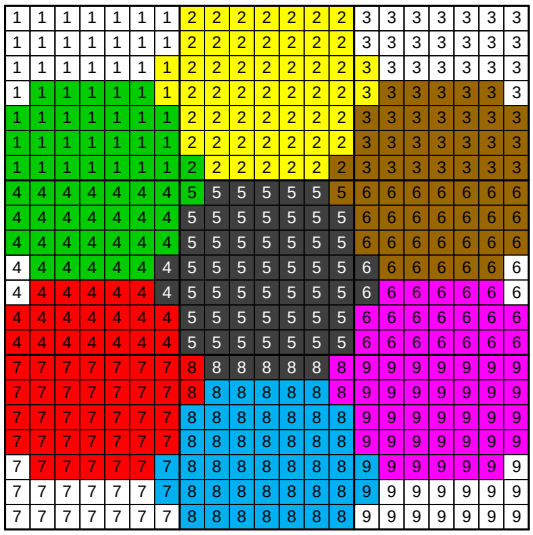


Figure 3. Square to hexagonal image technique [11]

In this research, in order to eliminate noise, the hexagonal image is filtered by a gaussian filter. Gaussian filter size can be varied as 3×3, 5×5, 7×7 and etc. The relation between the size (K) and standard deviation (σ) of gaussian filter is $\sigma = K/6$.

3. 2. Gradient Magnitude and Direction Calculation To determine directional changes of pixel intensities, gradient operation is used. The more changes in the intensity are occurred, the higher gradient values are achieved. Because of the high intensity changes in the edges, the gradient of edge regions is bigger than the uniform regions of the image. Therefore, the gradient is very powerful tool for edge detectors. As depicted in Figure 4a, in square domain, the gradient is a 2D vector with the components given by the derivation in x and y directions. Compared with the square images, the gradient in hexagonal domain is a 3D vector in which its components are obtained from derivation in r, s and t directions (see Figure 4(b)).

To calculate gradient in three directions, kernel matrices shown in Figure 5 are considered. Since the hexagonal image is not in square grid, a platform should be defined to locate hexagonal pixels. Figure 6 shows the proposed platform in which the filled circles are on locations of hexagonal pixels. Indeed, the hexagonal image in Figure 6(a) is mapped to the matrix in Figure 6(b).

As shown in Figure 5, and based on defined platform in Figure 6, the gradient in three directions for the pixel on location (i, j) is computed as:

$$g_r(i,j) = OP_r \times G_n(i,j) \quad (1)$$

$$g_s(i,j) = OP_s \times G_n(i,j) \quad (2)$$

$$g_t(i,j) = OP_t \times G_n(i,j) \quad (3)$$

where OP_r, OP_s, OP_t and $G_n(i,j)$ are defined as:

$$OP_r = \begin{bmatrix} 0 & -2 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix}, OP_s = \begin{bmatrix} 0 & -1 & 0 \\ -2 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 2 \\ 0 & 1 & 0 \end{bmatrix}, OP_t = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (4)$$

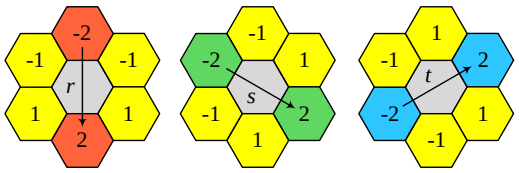
$$G_n(i.j) = \begin{bmatrix} 0 & I(i-2.j) & 0 \\ I(i-1.j-1) & 0 & I(i-1.j+1) \\ 0 & 0 & 0 \\ I(i+1.j-1) & 0 & I(i+1.j+1) \\ 0 & I(i+2.j) & 0 \end{bmatrix} \quad (5)$$

where $I(i,j)$ represents the intensity of pixel (i,j). The gradient in directions x and y is:

$$g_r^x(i,j) = g_r(i,j) \times \cos(\beta) \quad (6\text{-a})$$

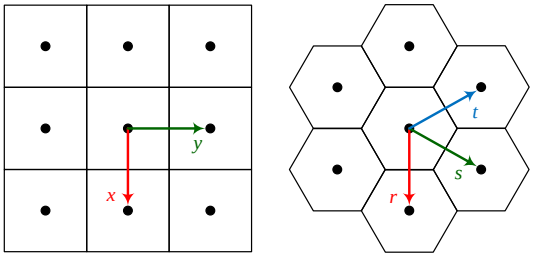
$$g_s^x(i,j) = g_s(i,j) \times \cos(\beta) \quad (6\text{-b})$$

$$g_t^x(i,j) = g_t(i,j) \times \cos(\beta) \quad (6\text{-c})$$



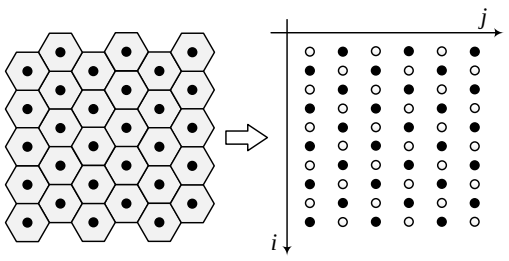
(a) (b) (c)

Figure 5. Gradient calculation in three directions: (a) r, (b) s and (c) t



(a) (b)

Figure 4. Gradient directions in: (a) Square and (b) Hexagonal



(a) (b)

Figure 6. Platform to locate hexagonal pixels

$$g_r^y(i,j) = g_r(i,j) \times sin(\beta) \quad (6\text{-d})$$

$$g_s^y(i,j) = g_s(i,j) \times sin(\beta) \quad (6\text{-e})$$

$$g_t^y(i,j) = g_t(i,j) \times sin(\beta) \quad (6\text{-f})$$

where β is defined as:

$$\beta = \begin{cases} 0\,, & for\ dierection\ r \\ \pi/3\,, & for\ direction\ s \\ 2\pi/3\,, & for\ direction\ t \end{cases} \quad (7)$$

The resultant vector in *x* and *y* directions is:

$$G_x(i,j) = g_r^x(i,j) + g_s^x(i,j) + g_t^x(i,j) \quad (8)$$

$$G_y(i,j) = g_r^y(i,j) + g_s^y(i,j) + g_t^y(i,j) \quad (9)$$

Finally, gradient magnitude and gradient direction of pixel (i,j) are $G_M(i,j)$ and $\theta(i,j)$, respectively, computed as follows:

$$G_M(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)} \quad (10)$$

$$\theta(i,j) = tan^{-1}\big(G_y(i,j)/G_x(i,j)\big) \quad (11)$$

where, θ is the gradient direction matrix.

3. 3. Level Quantization and Non-maximum Suppression Since the direction of gradient has different values, it should be quantized. Based on the Figure 4b, the direction of gradient is quantized into 3 main directions and takes values between $-\pi/2$ and $\pi/2$ added by $\pi/2$ leads to interval $0 \le G_D(i,j) < \pi$. As shown in Figure 7, the quantization is done in three directions demonstrated by red, green and blue arrows.

It is clear in Figure 7, the quantization of direction is calculated as:

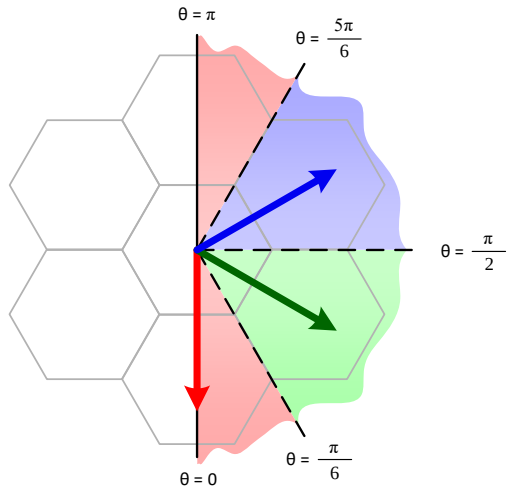


Figure 7. 3-Level direction quantization

$$G_D(i.j) = \begin{cases} 0, & 0 \le \theta(i,j) < \pi/6\ or\ 5\pi/6 \le \theta(i,j) < \pi \\ \pi/3, & \pi/6 \le \theta(i,j) < \pi/2 \\ 2\pi/3, & \pi/2 \le \theta(i,j) < 5\pi/6 \end{cases} \quad (12)$$

where G_D is the quantization of gradient direction matrix. After applying gradient calculation and angle quantization, non maximum suppression is used to select more significant edges which makes edges thinner.

Indeed, non-maximum suppression helps suppressing all gradient values except the local maxima and determines locations with the maximum change of intensity levels. To do this, for direction of the current pixel, if the gradient magnitude of the current pixel is greater than the gradient magnitude of that pixels in the negative and positive gradient directions, the pixel is preserved as edge candidate. Otherwise, the pixel is suppressed.

As shown in Figure 8a, if quantized direction of the central pixel is 0 ($G_D = 0$) and its gradient magnitude is greater than two adjacent gray pixels, the central pixel is used as edge candidate (set to 1), else, the central pixel is removed from edge candidates (set to 0). Figsures 8b and 8c are similar to Figure 8a but for directions $G_D = \pi/3$ and $G_D = 2\pi/3$. Based on defined platform in Figure 6, the edge candidates are determined as follows:

$$E(i,j) = \begin{cases} 1, G_D(i,j) = 0, G_M(i-2,j) < G_M(i,j), \\ \quad G_M(i,j) > G_M(i+2,j) \\ 1, G_D(i,j) = \pi/3, G_M(i-1,j-1) < G_M(i,j), \\ \quad G_M(i,j) < G_M(i+1,j+1) \\ 1, G_D(i,j) = 2\pi/3, G_M(i+1,j-1) < G_M(i,j), \\ \quad G_M(i,j) < G_M(i-1,j+1) \\ 0, Otherwise \end{cases} \quad (13)$$

where, E is the edge candidate matrix.

3. 4. Thresholds Selection and Double Thresholding Thresholding is an important process in a wide range of image processing applications. There are a variety of methods to calculate thresholds such as fixed thresholding, optimal thresholding, Otsu method and etc. In this research, thresholding is done by Otsu method which has better results than other methods because it minimizes intra-class variance or maximizes inter-class variance. Edge detection results depend on two parameters: window size of Gaussian filter and the threshold value. To achieve better results, while the size of window is increased, the value of threshold should be decreased.

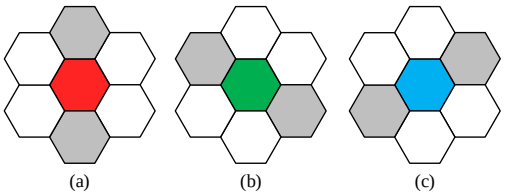


Figure 8. Non-maximum suppression for different directions: (a) $G_D = 0$, (b) $G_D = \pi/3$ and (c) $G_D = 2\pi/3$

After non-maximum suppression and remove some unreal edges, the remaining edge locations have more accurate representation of edges in the image. However, some remained edge pixels are because of noise, diversity of gradient domain value and etc. In order to remove these pixels, double thresholding is used in which two thresholds are defined: T_{low} and T_{high}. If gradient magnitude of an edge pixel is higher than the T_{high}, it is introduced as a strong edge pixel and should certainly be marked as real edge. If gradient magnitude of an edge pixel is lower than the T_{low}, it is removed. If gradient magnitude of an edge pixel is between T_{low} and T_{high}, it is introduced as a weak edge pixel. Since T_{low} and T_{high} are dependent on content of the image, the low and high thresholds should be defined based on the image content. As mentioned before, by inspiring Otsu thresholding method, a threshold can be defined as follows:

$$\sigma_b^2(t) = \sigma^2(t) - \sigma_w^2(t) = w_0(t)\mu_0^2(t) + w_1(t)\mu_1^2(t) - \mu_T^2(t) \quad (14)$$

Which $w_0(t) = \sum_{i=0}^{t-1} p(i)$, $w_1(t) = \sum_{i=t}^{L-1} p(i)$, $\mu_0(t) = \sum_{i=0}^{t-1} ip(i)/w_0(t)$, $\mu_0(t) = \sum_{i=t}^{L-1} ip(i)/w_1(t)$ and $\mu_T = \sum_{i=0}^{L-1} ip(i)$. It is clear that there are L intensity levels. The optimal threshold corresponds to maximum value of $\sigma_b^2(t)$ and is determined as T_{Otsu}. T_{high} and T_{low} defines as below:

$$T_{high} = \frac{7}{8} \times T_{Otsu}\,, \quad T_{low} = \frac{\sqrt{3}}{2} \times T_{high} \quad (15)$$

The coefficients of T_{low} and T_{high} are obtained by supervision.

3. 5. Edge Tracking by Hysteresis And Removing Untrue Edges Remained edges after non-maximum suppression are divided into two groups: strong and weak edges (see Figure 9).

We know that all weak edges are not real edges, as these edges can be extracted from either real edges or noise. To achieve better accuracy, the weak edges extracted form noise should be removed. Usually a weak edge pixel caused by noise is unconnected to a strong edge pixel while true edges are connected. To do this, as shown in Figure 10a, a 6-neighbor set is considered to search for strong edge pixels around weak edge pixel. For weak edge pixel P_c, if one or more pixels of $P_1, P_2, \ldots, P_6$ are not strong edges, the pixel P_c is removed.

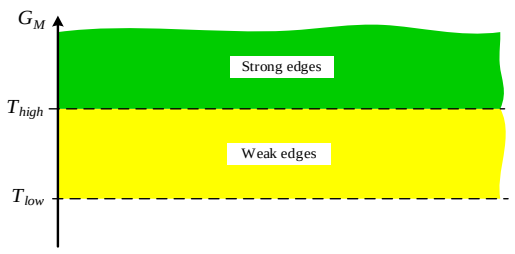


Figure 9. Strong and weak edges

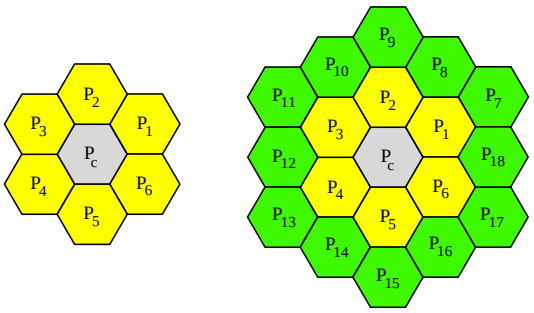


Figure 10. Definition of neighboring in hexagonal lattice: (a) 6-neighbor, (b) 18-neighbor

Finally, to remove edge pixels with no neighbor edges, a new algorithm is proposed. As shown in Figure 10, two neighboring in hexagonal lattice is defined and we have introduced ε_1 and ε_2 as below:

$$\varepsilon_1 = \sum_{i=1}^{6} P_i\,, \quad \varepsilon_2 = \sum_{i=1}^{18} P_i \quad (16)$$

As depicted in Figure 10b, ε_1 and ε_2 are sum of yellow and green pixels, respectively and P_i takes 0 (not edge) and 1 (edge) resulted from previous stage. If $\varepsilon_1 \geq 2$ or $\varepsilon_1/\varepsilon_2 \neq 1$ then the pixel p_c is preserved as edge, otherwise p_c is removed.

4. IMPLEMENTATION RESULTS

To evaluate performance of the proposed method, comparisons have been made between four existing methods and the proposed method. For this task, edge detection results of the proposed method and other methods are reported in synthetic and real images.

4. 1. Edge Detection Results on Synthetic Images For quantitative evaluation, 6 synthetic images horizontal, oriented, curved, square, chess and wave edges are used as shown in Figure 11.

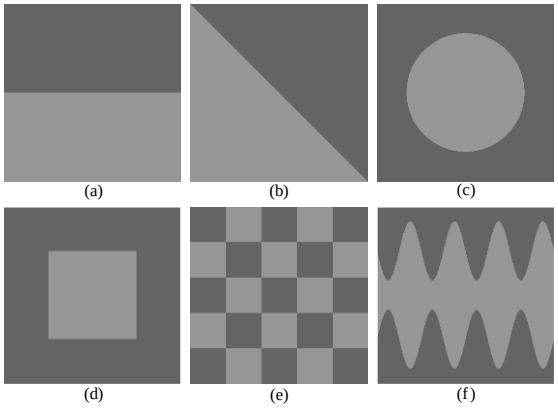


Figure 11. Synthetic images for edge detection evaluation

For synthetic evaluation, Figure of Merit (FoM) is used [39]. FoM considers three important errors including: missing valid edge pixels, failure to localize edge pixels and classification of noise fluctuations as edge pixels. FoM is defined as:

$$\text{FoM} = \frac{1}{\max(I_I, I_A)} \sum_{i=1}^{I_A} \frac{1}{1+\alpha d_i^2} \quad (17)$$

where I_A is the number of detected edge pixels, I_I is the number of ideal edge pixels, d_i is the separation distance of a detected edge point normal to a line of ideal edge points, and α is a scaling factor and usually set to 1/9.

Figure of Merit is normalized to interval [0 1], where 0 shows that none of the edge points were found correctly and 1 shows that all edge points were detected perfectly. To compare the proposed method with the other methods, we consider each synthetic image in Figure 11. It is necessary to mention that FoM criterion can only be used for synthetic images since in these images real edges are known.

To simulate a real environment for quantitative evaluation of all methods, the FoM measure is applied to images with different signal-to-noise ratios (SNRs), where $SNR = A^2/\sigma_n^2$, A is the difference between high and low intensities of synthetic images and σ_n^2 is the variance of noise. To generate synthetic images, A is configured to 50 with SNRs = 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 for all synthetic images in Figure 11. For each synthetic image and each SNR, 10 sets of test images are generated. In the next step, FoM of each test image is calculated and the FoM of a specific image and SNR is computed by averaging over these 15 test images. Therefore, each FoM is obtained from 10 test images. All test images are $6 \times 15 \times 10 = 900$. FoM results of methods Prewitt [30], He et al. [34], Sonya et al. [39], He et al. [45] and the proposed method applied to 6 synthetic images in 10 mentioned SNRs are shown in Table 1.

Results show that the proposed method has increased accuracy in the majority of cases. The proposed method outperforms other methods in Curved, Chess and Wave edge images over all SNRs. In the Chess edge images, the proposed method outperforms other methods in all SNRs. In the Horizontal, Curved and Wave edge images, the proposed method achieves better accuracy in 9 SNRs out of 10. In the Oriented edge image, the proposed method outperforms other methods in 6 SNRs among 10 ones and in the rest SNRs the FoM of the proposed method is very close to Sonya. Sonya achieves better FoM in 9 SNRs out of 10 on the Square edge image. Therefore, Sonya achieves better accuracy in Square edge image. To sum up, the proposed method outperforms other methods in 44 cases out of 60 (73.33% of cases).

4. 2. Edge Detection on Real Images For further evaluation of the proposed edge detector, 15 real images and 15 images taken from Coil-100 dataset [45] are used as shown in Figure 12.

To compare visually, edge detection results of all methods are shown in Figure 13. In each row, the left image is input image and the others are results of Middleton and Sivaswamy [30], He et al. [34], Sonya et al. [38], He et al. [44] and the proposed method, respectively.

TABLE 1. FoM results for all methods in different SNRs and Synthetic images shown in Figure 11

Image	Method	SNR=10	SNR=20	SNR=30	SNR=40	SNR=50	SNR=60	SNR=70	SNR=80	SNR=90	SNR=100
Horizontal	Proposed	**0.96099**	**0.96390**	**0.96115**	**0.96370**	**0.96561**	**0.96673**	**0.96655**	**0.96606**	0.96610	**0.96744**
	He [44]	0.52438	0.94502	0.95592	0.95671	0.95599	0.95601	0.96074	0.95921	0.96140	0.96243
	Sonya [38]	0.66342	0.95620	0.96072	0.96156	0.96396	0.96624	0.96618	0.96501	**0.97049**	0.96739
	He [34]	0.41895	0.70703	0.69897	0.71551	0.72674	0.71581	0.71994	0.73294	0.71669	0.72526
	Middleton [30]	0.45180	0.93496	0.94372	0.94499	0.94203	0.94205	0.94402	0.94367	0.94427	0.94665
Oriented	Proposed	**0.97649**	**0.98534**	0.98870	0.98998	**0.99100**	0.99114	0.99220	**0.99243**	**0.99282**	**0.99319**
	He [44]	0.69861	0.98127	0.98319	0.98297	0.98576	0.98790	0.99006	0.98794	0.98983	0.98899
	Sonya [38]	0.72752	0.98442	**0.98889**	**0.99073**	0.98935	**0.99164**	**0.99324**	0.99123	0.99203	0.99280
	He [34]	0.59332	0.92251	0.90948	0.92973	0.93646	0.93019	0.94358	0.94004	0.94408	0.94577
	Middleton [30]	0.63586	0.97089	0.97582	0.97832	0.98028	0.97915	0.98278	0.98367	0.98413	0.98484
Curved	Proposed	**0.97255**	**0.97876**	**0.98201**	**0.98278**	0.98335	**0.98393**	**0.98558**	**0.98598**	**0.98584**	**0.98724**
	He [44]	0.67431	0.96834	0.97439	0.98130	**0.98349**	0.98357	0.98310	0.98486	0.98533	0.98499
	Sonya [38]	0.62811	0.96866	0.97686	0.97654	0.97772	0.98017	0.97992	0.97980	0.98029	0.98069
	He [34]	0.48692	0.95903	0.96227	0.97320	0.96093	0.97728	0.97301	0.97934	0.97419	0.97852
	Middleton [30]	0.54296	0.95696	0.96085	0.95960	0.94371	0.97124	0.95502	0.97749	0.96486	0.97253
Square	Proposed	**0.91313**	0.96481	0.97426	0.97581	0.97806	0.97663	0.97694	0.97681	0.97893	0.97794
	He [44]	0.68941	0.97439	0.98017	0.98238	0.98508	0.98395	0.98483	0.98576	0.98631	0.98517

	Sonya [38]	0.71867	**0.98655**	**0.99139**	**0.99322**	**0.99444**	**0.99523**	**0.99676**	**0.99632**	**0.99727**	**0.99782**
	He [34]	0.61212	0.80975	0.81267	0.83886	0.84299	0.85720	0.86391	0.87249	0.87276	0.87786
	Middleton [30]	0.64802	0.96671	0.97003	0.97165	0.97264	0.97274	0.97389	0.97449	0.97520	0.97562
	Proposed	**0.96527**	**0.96924**	**0.97286**	**0.97259**	**0.97329**	**0.97465**	**0.97457**	**0.97609**	**0.97608**	**0.97594**
Chess	He [44]	0.87642	0.96829	0.97070	0.97193	0.97006	0.97081	0.97216	0.97274	0.97228	0.97301
	Sonya [38]	0.90638	0.92554	0.92758	0.92897	0.93165	0.93597	0.94201	0.94233	0.94417	0.94433
	He [34]	0.82936	0.80940	0.81811	0.83242	0.84313	0.84152	0.84751	0.85252	0.85684	0.85611
	Middleton [30]	0.85503	0.95528	0.95925	0.95901	0.96103	0.96088	0.96217	0.96317	0.96255	0.96350
	Proposed	**0.96661**	**0.97076**	**0.97233**	0.97471	**0.97607**	**0.97595**	**0.97710**	**0.97766**	**0.97789**	**0.97819**
	He [44]	0.87631	0.96893	0.97135	**0.97486**	0.97493	0.97235	0.97482	0.97399	0.97467	0.97620
Wave	Sonya [38]	0.87150	0.96694	0.97087	0.97303	0.97471	0.97423	0.97694	0.97652	0.97684	0.97725
	He [34]	0.78957	0.95220	0.94489	0.95508	0.96093	0.96593	0.96248	0.96235	0.97079	0.97147
	Middleton [30]	0.81584	0.96233	0.94929	0.95974	0.96001	0.96851	0.96418	0.95981	0.97223	0.97189

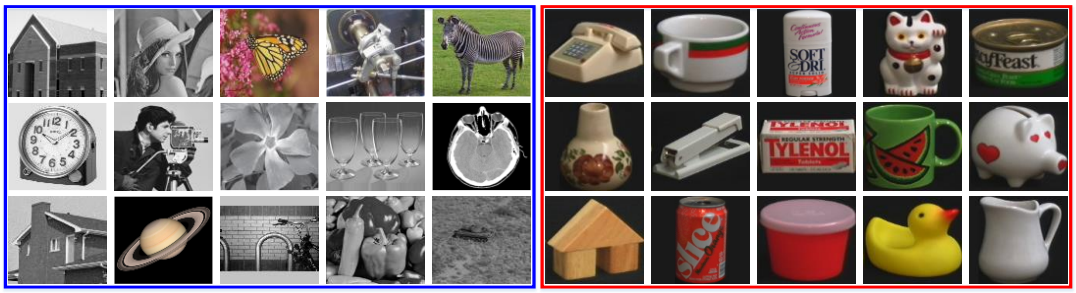


Figure 12. Real images (in blue box) and images from Coil-100 dataset (in red box) for quality evaluation

Figure 13. Edge detection results applied to images in Figure 12

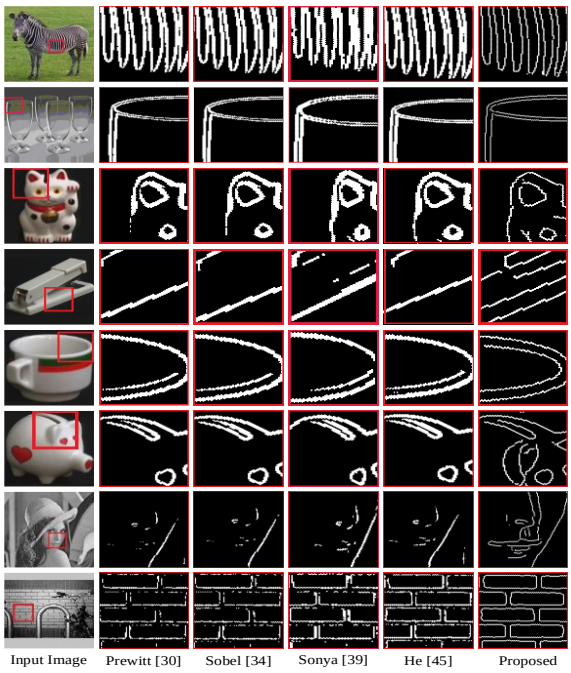


Figure 14. sub-image of 8 real images to illustrate edge detection quality of all methods in details

To illustrate edge detection quality of all methods in details, sub-image of 8 real images are selected and shown in Figure 14. It is obvious that detected edges in the proposed method are more accurate than the other methods. Indeed, the proposed method extracts the edges more accurately and edges are thinner than the other methods.

Edges are identified based on T_{low} and T_{high} thresholds in Equation (15). Pixels lower than T_{low} are excluded from edges and pixels greater than T_{high} are labeled as edge. Pixels between these thresholds are further evaluated in the next steps. Higher values of T_{low} leads to loss real edge pixels and lower values of T_{high} assigns edge label to non-edge pixels. Therefore, redundant edges appeared in images stem from threshold values. In the proposed methods, thresholds have been configured carefully, although even appropriate values of thresholds cannot exclude all redundant edges from edge pixels.

5. CONCLUSION

This paper presented a new edge detector in hexagonal lattice which includes square to hexagonal transformation, gradient magnitude and direction calculation, level quantization, non-maximum suppression, thresholds selection, double thresholding and edge tracking.

To evaluate detected edges, the proposed method was evaluated on two datasets of synthetic and real images. Results showed the superiority of the proposed method quantitatively and qualitatively. Suggested works can be directed toward implementing other image processing algorithms such as, corner detection, object segmentation and image filtering in hexagonal domain.

5. REFERENCES

1. Middleton, L. and Sivaswamy, J., "Hexagonal image processing: A practical approach, Springer Science & Business Media, (2006).
2. Coxeter, H.S.M., "Introduction to geometry", (1961).
3. Petersen, D.P. and Middleton, D., "Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces", ***Information and Control***, Vol. 5, No. 4, (1962), 279-323, doi: 10.1016/S0019-9958(62)90633-2.
4. Mersereau, R.M., "The processing of hexagonally sampled two-dimensional signals", ***Proceedings of the IEEE***, Vol. 67, No. 6, (1979), 930-949, doi: 10.1109/PROC.1979.11356.
5. Kamgar-Parsi, B., "Evaluation of quantization error in computer vision", ***IEEE Transactions on Pattern Analysis & Machine Intelligence***, Vol. 11, No. 09, (1989), 929-940, doi: 10.1109/34.35496.
6. Kamgar-Parsi, B., "Quantization error in hexagonal sensory configurations", ***IEEE Transactions on Pattern Analysis & Machine Intelligence***, Vol. 14, No. 06, (1992), 665-671, doi: 10.1109/34.141556.
7. Scotney, B., Coleman, S. and Gardiner, B., "Biologically motivated feature extraction using the spiral architecture", in 2011 18th IEEE International Conference on Image Processing, IEEE., (2011), 221-224.
8. Staunton, R.C., "An analysis of hexagonal thinning algorithms and skeletal shape representation", ***Pattern recognition***, Vol. 29, No. 7, (1996), 1131-1146, doi: 10.1016/0031-3203(94)00155-3.
9. Staunton, R., "One-pass parallel hexagonal thinning algorithm", ***IEE Proceedings-Vision, Image and Signal Processing***, Vol. 148, No. 1, (2001), 45-53, doi: 10.1049/cp:19990443.
10. Fadaei, S. and Rashno, A., "A framework for hexagonal image processing using hexagonal pixel-perfect approximations in subpixel resolution", ***IEEE Transactions on Image Processing***, Vol. 30, (2021), 4555-4570, doi: 10.1109/TIP.2021.3073328.
11. Asharindavida, F., Hundewale, N. and Aljahdali, S., "Study on hexagonal grid in image processing", ***Proc. ICIKM***, Vol. 45, (2012), 282-288, doi: 10.1.1.707.7492.
12. He, X., Jia, W. and Wu, Q., "An approach of canny edge detection with virtual hexagonal image structure", in 2008 10th International Conference on Control, Automation, Robotics and Vision, IEEE., (2008), 879-882.
13. Staunton, R.C., "The processing of hexagonally sampled images", ***Advances in Imaging and Electron Physics***, Vol. 119, (2001), 191-265, doi: 10.1016/S1076-5670(01)80088-4.
14. Jeevan, K. and Krishnakumar, S., "Compression of images represented in hexagonal lattice using wavelet and gabor filter", in 2014 International Conference on Contemporary Computing and Informatics (IC3I), IEEE., (2014), 609-613.
15. Jiang, Q., "Fir filter banks for hexagonal data processing", ***IEEE Transactions on Image Processing***, Vol. 17, No. 9, (2008), 1512-1521, doi: 10.1109/TIP.2008.2001401.

16. Nourian, M.B. and Aahmadzadeh, M., "Image de-noising with virtual hexagonal image structure", in 2013 First Iranian Conference on Pattern Recognition and Image Analysis (PRIA), IEEE., (2013), 1-5.
17. Li, X., "Implementation of a simulated display for hexagonal image processing", ***Displays***, Vol. 50, (2017), 63-69, doi: 10.1016/j.displa.2017.09.005.
18. Wang, F., Ni, J. and Guo, R., "Modulation transfer function of an imaging system with a hexagonal pixel array detector", ***Optik***, Vol. 179, (2019), 986-993, doi: 10.1016/j.ijleo.2018.11.035.
19. Gardiner, B., Coleman, S.A. and Scotney, B.W., "Multiscale edge detection using a finite element framework for hexagonal pixel-based images", ***IEEE Transactions on Image Processing***, Vol. 25, No. 4, (2016), 1849-1861, doi: 10.1109/TIP.2016.2529720.
20. Li, X., "Storage and addressing scheme for practical hexagonal image processing", ***Journal of Electronic Imaging***, Vol. 22, No. 1, (2013), 010502, doi: 10.1117/1.JEI.22.1.010502.
21. Liu, S.J., Coleman, S., Kerr, D., Scotney, B. and Gardiner, B., "Corner detection on hexagonal pixel based images", in 2011 18th IEEE International Conference on Image Processing, IEEE., (2011), 1025-1028.
22. He, X., Jia, W., Hur, N., Wu, Q. and Kim, J., "Image translation and rotation on hexagonal structure", in The Sixth IEEE International Conference on Computer and Information Technology (CIT'06), IEEE., (2006), 141-141.
23. Gardiner, B., Coleman, S. and Scotney, B., "Fast edge map pyramids for hexagonal image structures", in 2009 13th International Machine Vision and Image Processing Conference, IEEE., (2009), 41-46.
24. He, X., Wang, H., Hur, N., Jia, W., Wu, Q., Kim, J. and Hintz, T., "Uniformly partitioning images on virtual hexagonal structure", in 2006 9th International Conference on Control, Automation, Robotics and Vision, IEEE., (2006), 1-6.
25. Fadaei, S., Amirfattahi, R. and Ahmadzadeh, M.R., "New content-based image retrieval system based on optimised integration of dcd, wavelet and curvelet features", ***IET Image Processing***, Vol. 11, No. 2, (2017), 89-98, doi: 10.1049/iet-ipr.2016.0542.
26. Duan, Y., Lu, J., Feng, J. and Zhou, J., "Context-aware local binary feature learning for face recognition", ***IEEE transactions on Pattern Analysis and Machine Intelligence***, Vol. 40, No. 5, (2017), 1139-1153, doi: 10.1109/TPAMI.2017.2710183.
27. Fadaei, S., Amirfattahi, R. and Ahmadzadeh, M.R., "Local derivative radial patterns: A new texture descriptor for content-based image retrieval", ***Signal Processing***, Vol. 137, (2017), 274-286, doi: 10.1016/j.sigpro.2017.02.013.
28. Fadaei, S. and Rashno, A., "Content-based image retrieval speedup based on optimized combination of wavelet and zernike features using particle swarm optimization algorithm", ***International Journal of Engineering***, ***Transactions B: Applications***, Vol. 33, No. 5, (2020), 1000-1009, doi: 10.5829/IJE.2020.33.05B.34.
29. Fadaei, S., "New dominant color descriptor features based on weighting of more informative pixels using suitable masks for content-based image retrieval", ***International Journal of Engineering***, ***Transactions B: Applications,*** Vol. 35, No. 8, (2022), doi: 10.5829/IJE.2022.35.08B.01.
30. Middleton, L. and Sivaswamy, J., "Edge detection in a hexagonal-image processing framework", ***Image and Vision Computing***, Vol. 19, No. 14, (2001), 1071-1081, doi: 10.1016/S0262-8856(01)00067-1.
31. Staunton, R.C., "The design of hexagonal sampling structures for image digitization and their use with local operators", ***Image and Vision Computing***, Vol. 7, No. 3, (1989), 162-166, doi: 10.1016/0262-8856(89)90040-1.
32. Li, J., Tang, X. and Jiang, Y., "Comparing study of some edge detection algorithms", ***Information Technology***, Vol. 38, No. 9, (2007), 106-108, doi: 10.1109/ICCCIS51004.2021.9397225.
33. Davies, E., "Circularity—a new principle underlying the design of accurate edge orientation operators", ***Image and Vision Computing***, Vol. 2, No. 3, (1984), 134-142, doi: 10.1016/0262-8856(84)90049-0.
34. He, X., Jia, W., Li, J., Wu, Q. and Hintz, T., "An approach to edge detection on a virtual hexagonal structure", in 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007), IEEE., (2007), 340-345.
35. He, X., Li, J., Wei, D., Jia, W. and Wu, Q., "Canny edge detection on a virtual hexagonal image structure", in 2009 Joint Conferences on Pervasive Computing (JCPC), IEEE., (2009), 167-172.
36. Gardiner, B., Coleman, S. and Scotney, B., "Multi-scale feature extraction in a sub-pixel virtual hexagonal environment", in 2008 International Machine Vision and Image Processing Conference, IEEE., (2008), 111-116.
37. Mostafa, K., Chiang, J. and Her, I., "Edge-detection method using binary morphology on hexagonal images", ***The Imaging Science Journal***, Vol. 63, No. 3, (2015), 168-173, doi: 10.1179/1743131X14Y.0000000098.
38. Coleman, S., Scotney, B. and Gardiner, B., "Tri-directional gradient operators for hexagonal image processing", ***Journal of Visual Communication and Image Representation***, Vol. 38, (2016), 614-626, doi: 10.1016/j.jvcir.2016.04.001.
39. Li, X., "Simplified square to hexagonal lattice conversion based on 1-d multirate processing", ***Signal Processing: Image Communication***, Vol. 99, (2021), 116481, doi: 10.1016/j.image.2021.116481.
40. Schlosser, T., Beuth, F. and Kowerko, D., "Biologically inspired hexagonal deep learning for hexagonal image generation", in 2020 IEEE International Conference on Image Processing (ICIP), IEEE., (2020), 848-852.
41. Varghese, P. and Saroja, G.A.S., "Hexagonal image enhancement using hex-gabor filter for machine vision applications", ***Materials Today: Proceedings***, Vol. 56, (2022), 555-558, doi: 10.1016/j.matpr.2022.02.277.
42. Luo, J., Zhang, W., Su, J. and Xiang, F., "Hexagonal convolutional neural networks for hexagonal grids", ***IEEE Access***, Vol. 7, (2019), 142738-142749, doi: 10.1109/ACCESS.2019.2944766.
43. Varghese, P. and Saroja, G.A.S., "Hexagonal image compressionusing singular value decomposition in python", in 2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), IEEE., (2021), 211-215.
44. He, X., Wei, D., Lam, K.-M., Li, J., Wang, L., Jia, W. and Wu, Q., "Canny edge detection using bilateral filter on real hexagonal structure", in International Conference on Advanced Concepts for Intelligent Vision Systems, Springer., (2010), 233-244.
45. Nene, S.A., Nayar, S.K. and Murase, H., "Columbia object image library (coil-100)", (1996).

Persian Abstract

چکیده

در پردازش تصویر، پیکسل‌های شش‌ضلعی نسبت به پیکسل‌های مربعی مزایای زیادی دارند. از طرفی استخراج ویژگی در بسیاری از کاربردهای پردازش تصویر از جمله تشخیص چهره، تشخیص کاراکتر، اثر انگشت و پزشکی استفاده می‌شود. آشکارسازی لبه یکی از الگوریتم‌های مهم استخراج ویژگی در تصویر است و لبه‌یاب کنی از معروفترین روش‌های استخراج لبه‌های تصویر است که نسبت به دیگر روش‌ها عملکرد مناسب‌تری دارد. در این مقاله یک الگوریتم لبه‌یابی کنی برای شبکه‌های شش‌ضلعی ارایه شده است. در روش پیشنهادی ابتدا تصویر از حالت مربعی به شش‌ضلعی تبدیل شده سپس الگوریتم لبه‌یابی کنی روی شبکه‌ی شش‌ضلعی تعریف می‌شود. روش پیشنهادی به تصاویر مصنوعی و همچنین تصاویر واقعی با سیگنال به نویزهای مختلف اعمال شده است. بر اساس معیار FoM برای تصاویر مصنوعی، روش پیشنهادی در ٤٤ حالت از ٦٠ حالت، بهتر از دیگر روش‌ها عمل نموده است و نتایج لبه‌یابی روی تصاویر واقعی برتری روش پیشنهادی را نسبت به دیگر روش‌ها نشان می‌دهد.