Multimedia Tools and Applications (2020) 79:7021–7048
https://doi.org/10.1007/s11042-019-08487-z

FTSH: a framework for transition from square image processing to hexagonal image processing

Taner Cevik[1] · Mustafa Fettahoglu[2] · Nazife Cevik[3] · Serdar Yilmaz[1]

Received: 1 February 2019 / Revised: 2 September 2019 / Accepted: 19 November 2019 /
Published online: 18 December 2019

Abstract

This paper proposes a novel framework for transition from the ordinary square-pixel-based image processing (SIP) domain to the hexagonal-pixel-based (HIP) domain (FTSH). The conventional image acquisition and processing are based on square pixels. However, HIP can provide promising advantages in many respects, such as degrading the curse of data size and accordingly reducing the processing time. HIP did not achieve satisfactory attraction because all software, including libraries, methods and structures, as well as mathematical operations and methodologies developed to date, are aimed at SIP. In this study, we propose a framework containing the corresponding HIP equivalents of some basic SIP methods and operations. In addition, the results of these basic operations in both SIP and HIP areas are presented comparatively. Since there is no common and standardized framework or library for HIP, this study can be used by other researchers who wish to enter the HIP. Simulation results support the competitive performance of HIP, and this promising performance can be carried far beyond when properly handled and focused.

Keywords Hexagonal image processing · Square image processing · Framework

✉ Taner Cevik
tanercevik@aydin.edu.tr

Mustafa Fettahoglu
mustafafettahoglu@stu.aydin.edu.tr

Nazife Cevik
nazifecevik@arel.edu.tr

Serdar Yilmaz
serdaryilmaz1@aydin.edu.tr

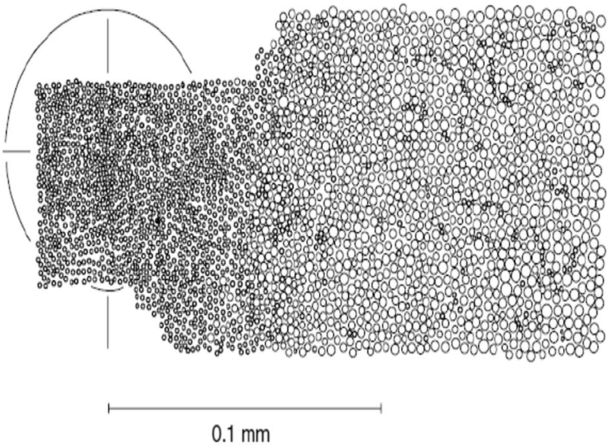
[1] Department of Software Engineering, Istanbul Aydin University, Istanbul, Turkey

[2] Department of Computer Engineering, Istanbul Aydin University, Istanbul, Turkey

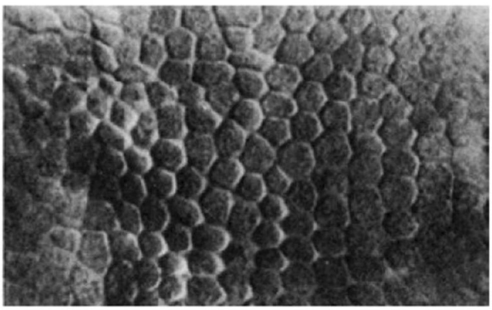
[3] Department of Computer Engineering, Istanbul Arel University, Istanbul, Turkey

1 Introduction

Image processing is the imitation of human visual system by using a computer. In the human visual system, light enters the eye through the pupil behind the cornea and is then projected by the lens into the spherical interior of the back of the eye. In this section, the retina, light is converted into electrical signals. Photoreceptors are found in the deepest and thinnest part of the retina called fovea. There are two types of receptors in the retina, namely rods and cones. Cones that are specialized in high-resolution and color vision, contribute to the day-time vision. These photoreceptors reside in the central region of the fovea. In the contrary, rods are specialized on colorless and dark stimulus, therefore contributes to the night time vision. The settlement of the photoreceptors along the circular retinal surface, is shown in Fig. 1(a). Here, the bigger circles relate to the rods and the littler circles to the cones. The most remarkable point to see is that, the general topology in this chart is generally hexagonal. This is on the grounds that, as will be shown later, all normally deformable round structures are bundled in the best two measurements in a hexagonal example, for example, in honeycombs. A case of an



(a)



(b)

Fig. 1 (a) Settlement of cones and rods along the fovea [39]. (b) A zoomed in view of a portion of the foveal region [10]

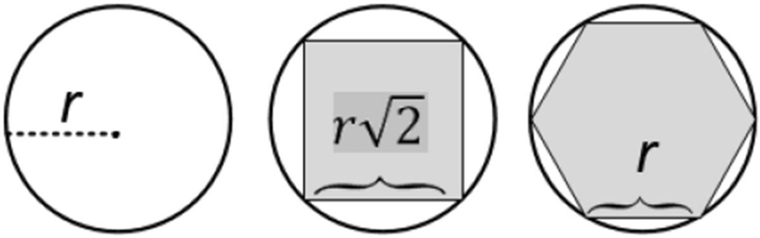


Fig. 2 The illustration of internal coverage by square and hexagonal pixels

all-inclusive segment of the foveal district of the retina area demonstrating this conduct is illustrated in Fig. 1 (b) [35].

Image processing is the art of imitating the human vision and transferring it to computer vision. The data in the physical environment, which is actually light, is continuous and some special sensors are used to obtain this continuous data. These sensors differ in the context of the light spectrum which they are sensitive and are used in square or rectangular arrays. Although light data is continuous, computers can only process digital data. Therefore, continuous light data should be sampled and digitized. Since square or rectangular sensor arrays are used, the latter processes at the computer side have been designed accordingly. Therefore, pixel, which is the smallest data unit of computerized digitized data, is also designed as a square. However, sampling light data on a hexagonal lattice and then maintaining the subsequent processes at hexagonal domain can change many things and yield promising results. The hexagonal geometry has been investigated for several decades. It had been a conjecture that the best way to partition a plane into regions with equal areas can be done by means of hexagons till it was proved by Hales [18, 19]. Besides the natural hexagonal arrangement of photoreceptors in fovea, another hexagonal natural encounter of hexagon geometry is the honeycombs.

The hexagonal lattice has some advantages over the square lattice. Firstly, in accordance with the isopimetric theorem, a hexagonal circle occupies more space than any other closed planar curve except circle. However, it is not possible to cover a plane totally by circles. This means that the sampling density of a hexagonal cage is higher

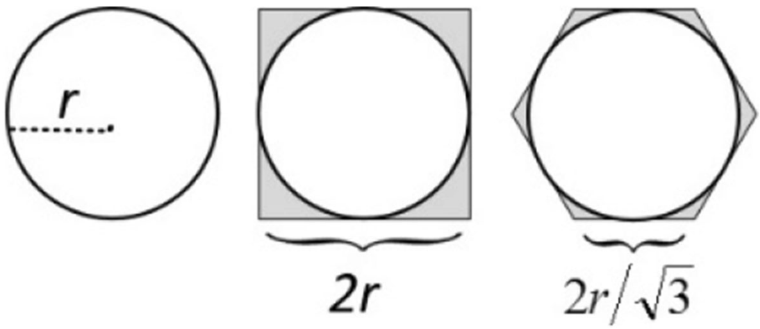


Fig. 3 The illustration of external coverage by square and hexagonal pixels

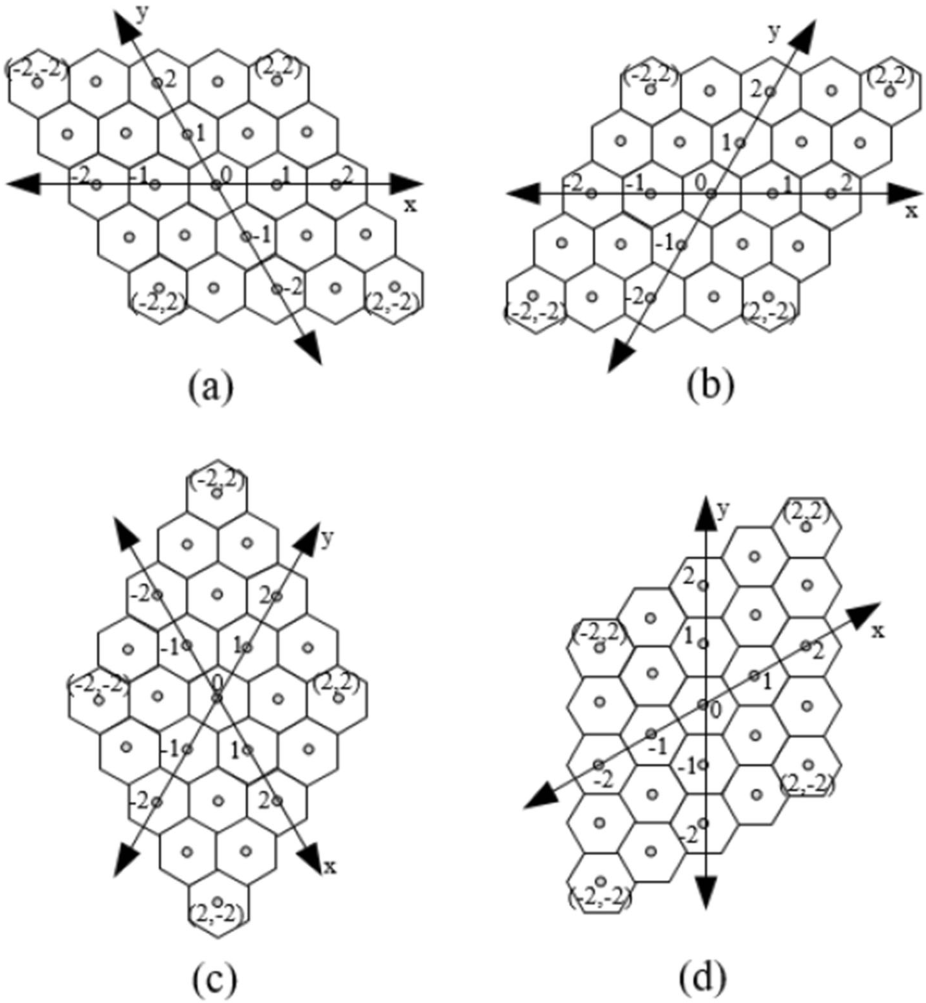


Fig. 4 Variations of addressing schemes on hexagonal grid by using two skewed axes

than that of a square cage. Secondly, in a hexagonal lattice each hexagon has six equidistant neighbors with sharing an edge with each [20]. However, a square has two different types of neighbors. One group of neighbors reside on vertical and horizontal axis, and other group of neighbors reside on diagonal, which are further distant than the first group. The central pixel shares an edge with the first group of neighbors, in contrast, share a corner with the neighbors belonging to the second group. This violates the homogeneity and hardens the process of edge tracking via the neighbors. Furthermore, hexagonal pixels can achieve a better resolution when compared to square pixels. The superiority of hexagonal pixels over square pixels in terms of better higher resolution providence is described as follows:

Let r be the radius of a circle that is the smallest unit area in the spatial domain. The smallest inscribing square and hexagon to cover the area of this circle have the side lengths $r\sqrt{2}$ and r respectively as shown in Fig. 2.

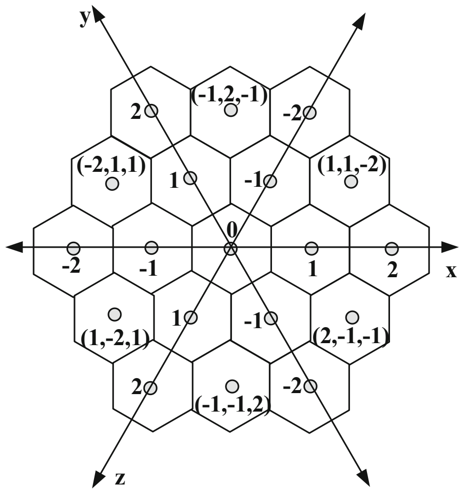


Fig. 5 Three-skewed-axes addressing schemes on hexagonal grid

The comparison in terms of coverage, which also points to the resolution, is performed as follows. Let C_{sqr} and C_{hex} denote the coverage of square and hexagonal pixels respectively and calculated as:

$$C_{sqr} = \frac{A_{sqr}}{A_{cir}} = \frac{\left(r\sqrt{2}\right)^2}{\pi r^2} = \frac{2}{\pi} \tag{1}$$

$$C_{hex} = \frac{A_{hex}}{A_{cir}} = \frac{6\left(r^2\sqrt{3}/2\right)/2}{\pi r^2} = \frac{3\sqrt{3}/2}{\pi} \tag{2}$$

where A_{cir}, A_{sqr} and A_{hex} identify the areas of circle, square and hexagon respectively.

Thus, it can be inferred from Eq. (1–2):

$$Eff_{cov} = \frac{C_{hex}}{C_{sqr}} = \frac{\frac{3\sqrt{3}/2}{\pi}}{\frac{2}{\pi}} = \frac{3\sqrt{3}}{4} = 1.3 \tag{3}$$

where Eff_{cov} denotes the coverage efficiency.

In addition to the internal coverage efficiency, the amendment in the unnecessary consumption through external coverage (Fig. 3) can also be measured as follows:

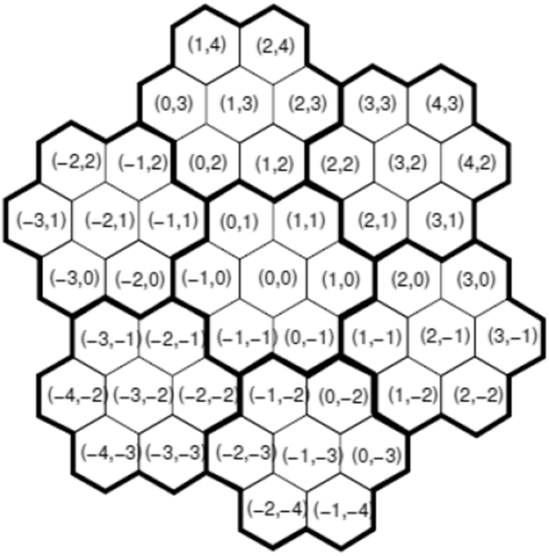
$$A_{rdnt_{square}} = A_{sqr} - A_{cir} = 4r^2 - \pi r^2 = r^2(4-\pi) \tag{4}$$

$$A_{rdnt_{hex}} = A_{hex} - A_{cir} = 2\sqrt{3r^2} - \pi r^2 = r^2\left(2\sqrt{3} - \pi\right) \tag{5}$$

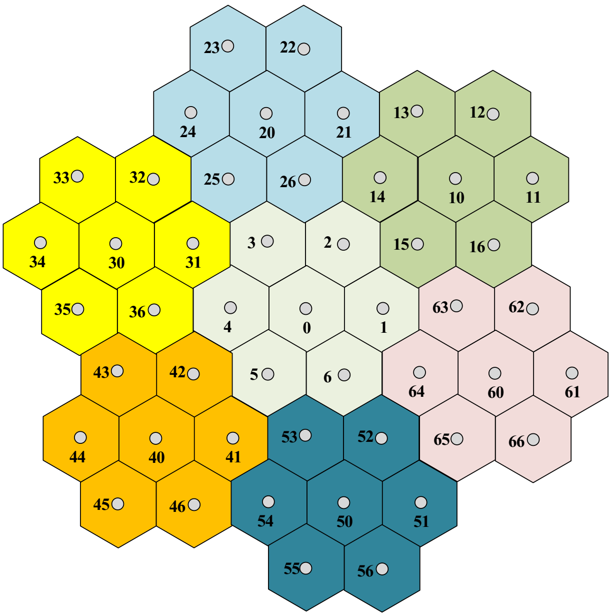
Thus, it can be inferred from Eq.(4–5):

$$Rdnt_{cov} = \frac{A_{rdnt_{square}}}{A_{rdnt_{hex}}} = \frac{r^2(4-\pi)}{r^2\left(2\sqrt{3}-\pi\right)} = \frac{4-\pi}{2\sqrt{3}-\pi} \cong 2.6875 \tag{6}$$

where $Rdnt_{cov}$ denotes the coverage redundancy.



(a)



(b)

Fig. 6 Sample hierarchical addressing schemes (a) Pyramid (b) Second-level aggregate

As identified by Eq. (3) and Eq. (6), using hexagonal pixels increases the coverage efficiency and degrades the coverage redundancy.

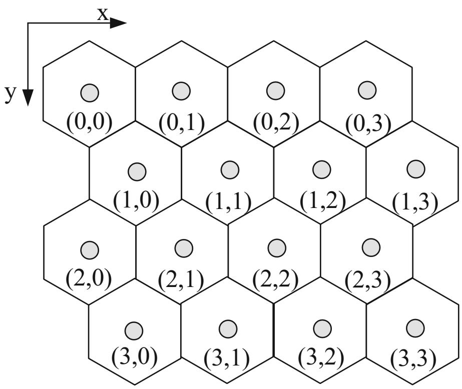


Fig. 7 Ordinary flat addressing scheme

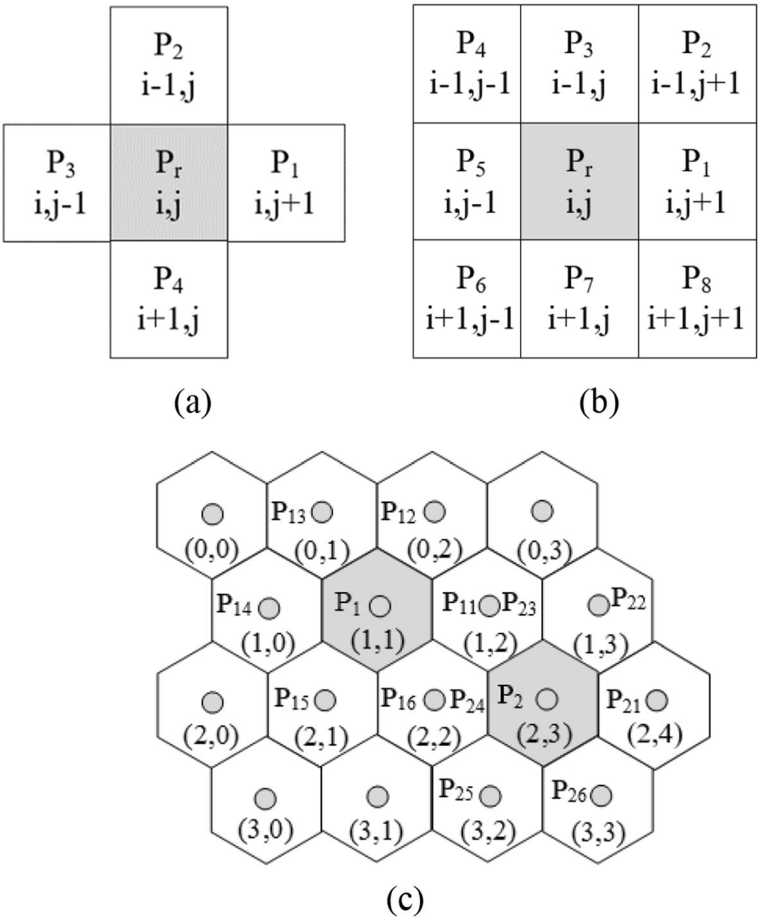


Fig. 8 Neighbourhood definitions **(a)** Square 4-connected neighbourhood **(b)** Square 8-connected neighbourhood **(c)** Hexagonal neighbourhood

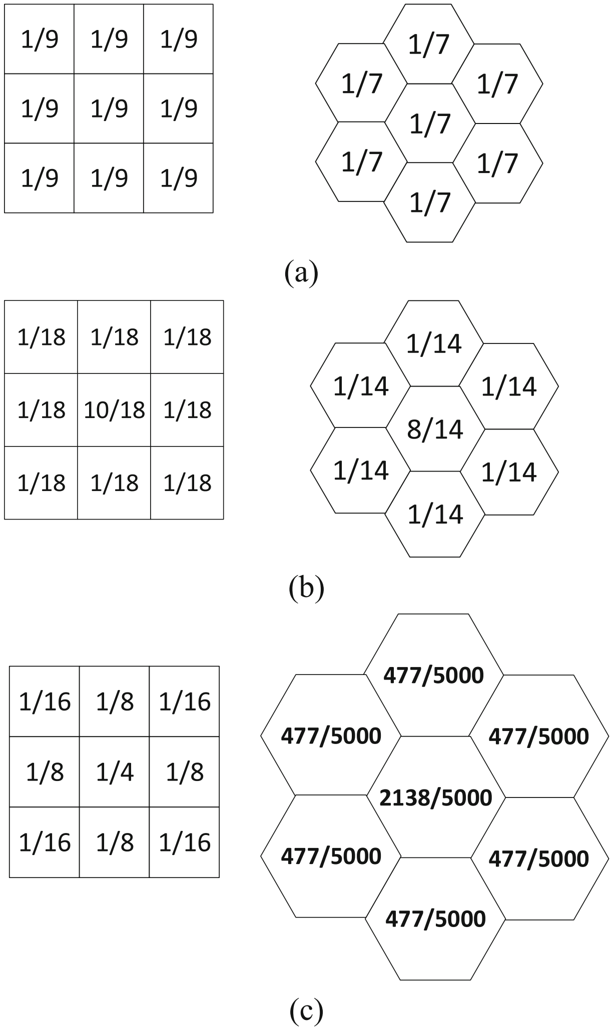


Fig. 9 The mean, weighted average and Gaussian kernels on square and hexagonal domains **(a)** Square and hexagonal mean kernels **(b)** Square and hexagonal weighted average kernels. **(c)** Square and hexagonal Gaussian kernels

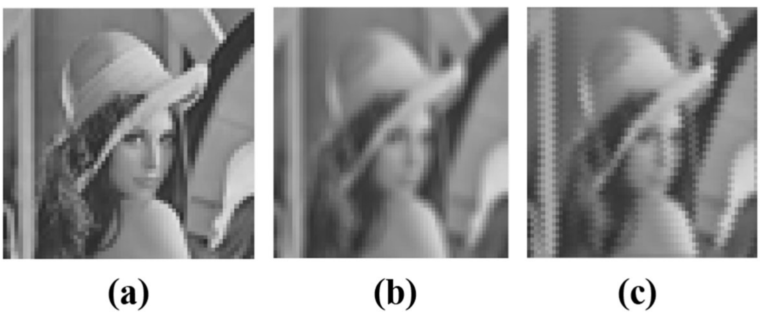


Fig. 10 **(a)** The original Lena image **(b)** Lena blurred by the mean kernel on the square domain **(c)** Lena blurred by the mean kernel on the hexagonal domain

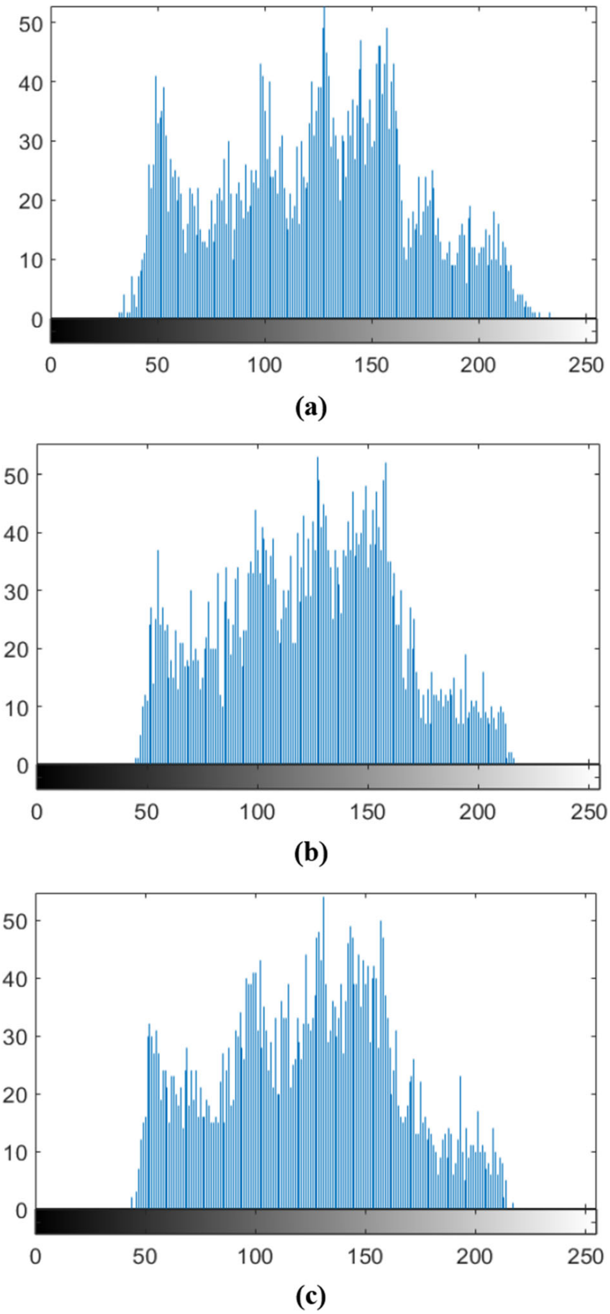


Fig. 11 Histograms of the images in Fig 10 **(a)** Histogram of the original Lena image **(b)** Histogram of Lena blurred by the mean kernel on the square domain **(c)** Histogram of Lena blurred by the mean kernel on the hexagonal domain

The proposed research studies on the use of hexagons during image processing have focused mostly on coordinate mapping from the square area to the hexagon region. The

Fig. 12 **(a)** The original Lena image **(b)** Lena blurred by the weighted average kernel on the square domain **(c)** Lena blurred by the weighted average kernel on the hexagonal domain

most important part, however, is how to handle operations after the mapping phase. Because, the system and its constituents, such as sensing elements, presentation hardware, software, mathematics, etc., are built on the idea of square pixel logic. Therefore, there are no standards or libraries or packages that are globally accepted for hexagonal image processing [4]. Thus, in this study, it is intended to develop a framework on hexagonal domain which performs some basic operations of image processing such as blurring, edge detection, noise filtering and recognition. The results of these procedures and their corresponding square-domain equivalents are also presented and compared. The rest of the paper is organized as follows. Section 2 presents the operations accomplished in hexagonal domain and their results, as well as the methodologies followed during transition from SIP to HIP. Finally, Section 3 concludes the paper.

2 Framework for transition from square image processing to hexagonal image processing (FTSH)

Hexagonal geometry can provide significant improvements in the area of image processing. Studies so far have focused on the difficulty of mapping, and elegant methods have been proposed to solve how the data in square pixels are represented by hexagonal pixels. Though there are generally accepted ideas about the mapping process, there is no agreed standard on how things will be handled after the mapping phase. Thus, FTSH is a starting point to consider and illustrate how some basic operations can be performed in the hexagonal domain.

2.1 Mapping from a square domain to a hexagonal domain

Since the points on a hexagonal grid are not aligned in two orthogonal directions, is not always possible to represent the addresses of the points on a hexagonal grid with integer Cartesian coordinates. As a solution for that can be benefiting from the nature of hexagonal grids and assigning the axis of symmetry of the hexagon as the coordinate axes. One of the easiest way is using two skewed axes that are 60° or 120° apart from each other as depicted in Fig. 4. This way is efficient because, it is possible to address a point on the hexagonal plane by two integer coordinates. A number of combinations can be applied by rotating the skewed axes in any direction, however the coordinate is going to stay same. Many studies proposed in the literature [2, 32, 34, 36, 37, 43, 44, 46, 48–50, 53, 54] have applied one of the addressing schemes illustrated in Fig. 4.

In addition to the two skewed axis addressing schemes, other alternative addressing methods have been proposed. One of these is the three skewed axis scheme [22, 23], using the three symmetric axes of the hexagon as depicted in Fig. 5 rather than two. Although this addressing scheme appears to be advantageous for operations such as rotation that involve high degree of symmetry, an increased burden is suffered in terms of complex data structures and processing time, especially for non-symmetric operations.

Another prominent addressing method is the hierarchical addressing scheme. In such addressing schemes [3, 15–17, 26, 29, 30, 38, 47], hexagons are considered hierarchically, such as in a pyramid architecture or in a sort of collection as shown in Fig. 6.

In FTSH, the idea of Overington [38] is pursued rather than the above-mentioned schemes. That is, the hexagonal grid is treated as a rectangular grid with respect to row-column manner as illustrated in Fig. 7.

The hexagons in the odd-numbered rows are shifted by $a\sqrt{3}/2$ on the horizontal axis as shown in Fig. 7. This forced-shifting does not cause a problem during the mapping process.

Most of the image processing operations require padding and neighborhood definitions. The padding and neighborhood definition operations are easily handled on square domain. However, for pixels located on the sides of the image in the hex grid, special processing is required because of the mandatory shift. The challenges of padding and neighborhood definition operations and the solutions that manage them are briefly described below.

2.2 Padding and neighborhood definition operations on hexagonal domain

Almost all image processing operations such as blurring, sharpening, edge detection, and so on require neighborhood definitions. That is, these operations involve not only the reference pixel itself, but also the intensity levels of the neighboring pixels. Thus, each pixel must have neighboring pixels defined. As is known, there are two types of neighbor for each pixel in the square area, because not all adjacent neighbors of a pixel are at equal distance from the reference pixel. These are basically 4-connected and 8-connected neighbors, as shown in Fig. 8 (a-b). However, there is only one type of neighbor on the hexagonal grid. This is because all adjacent neighbors are equidistant from the reference pixel as depicted in Fig. 8 (c). While addressing the adjacent neighbors, regardless whether 4-connected or 8-connected, of a reference pixel on the square domain, the row and column indices are easily used. Furthermore, this methodology is the same for all pixels in an image on the square grid. However, it is not possible to apply the same methodology for the hexagonal domain. Because the indexes of neighboring pixels are not uniform and differ depending on the row and column of the reference pixel. As given in Algorithm 1, the solution is to treat each type differently, taking into account the position of the reference pixel.

Algorithm 1: Neighbor assignment and padding by zero

```
function HexagonalNgbAssignment(fHex);
  Input: Hexagonal version of the original image f
  Output: fHexPadded
  rows←numberOfRows(fHex);
  cols←numberOfColumns(fHex);
  for i←1 to rows do
    for j←1 to cols do
        if i = 1 && j = 1 then
          ngb1 = fHex(i,j+1); ngb2 = 0; ngb3 = 0;
          ngb4 = 0; ngb5 = 0; ngb6 = fHex(i+1,j);
        else if i = 1 && j > 1 && j < cols
          ngb1 = fHex(i,j+1); ngb2 = 0;
          ngb3 = 0; ngb4 = fHex(i,j-1);
          ngb5 = fHex(i+1,j-1); ngb6 = fHex(i+1,j);
        else if i = 1 && j = cols
          ngb1 = 0; ngb2 = 0;
          ngb3 = 0; ngb4 = fHex(i,j-1);
          ngb5 = fHex (i+1,j-1); ngb6 = fHex(i+1,j);
        else if j=cols && i >1 && i <rows && i is even
          ngb1 = 0; ngb2 = 0;
          ngb3 = fHex(i-1,j); ngb4 = fHex(i,j-1);
          ngb5 = fHex(i+1,j); ngb6 = 0;
        else if j=cols && i > 1 && i < rows && i is odd
          ngb1 = 0; ngb2 = fHex(i-1,j);
          ngb3 = fHex(i-1,j-1); ngb4 = fHex(i,j-1);
          ngb5 = fHex(i+1,j-1); ngb6 = fHex(i+1,j);
        else if i=rows && j=cols && i is even
          ngb1 = 0; ngb2 = 0;
          ngb3 = fHex(i-1,j); ngb4 = fHex(i,j-1);
          ngb5 = 0; ngb6 = 0;
        else if i=rows && j=cols && i is odd
          ngb1 = 0; ngb2 = fHex(i-1,j);
          ngb3 = fHex(i-1,j-1); ngb4 = fHex(i,j-1);
          ngb5 = 0; ngb6 = 0;
        else if i=rows && j > 1 && j<cols && i is even
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j+1);
          ngb3 = fHex(i-1,j); ngb4 = fHex(i,j-1);
          ngb5 = 0; ngb6 = 0;
        else if i=rows && j > 1 && j<cols && i is odd
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j);
          ngb3 = fHex(i-1,j-1); ngb4 = fHex(i,j-1);
          ngb5 = 0; ngb6 = 0;
        else if i=rows && j=1 && i is even
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j+1);
          ngb3 = fHex(i-1,j); ngb4 = 0;
          ngb5 = 0; ngb6 = 0;
        else if i=rows && j=1 && i is odd
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j);
          ngb3 = 0; ngb4 = 0;
          ngb5 = 0; ngb6 = 0;
        else if j=1 && i >1 && i<rows && i is even
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j+1);
          ngb3 = fHex(i-1,j); ngb4 = 0;
          ngb5 = fHex(i+1,j); ngb6 = fHex(i+1,j+1);
        else if j=1 && i >1 && i<rows && i is odd
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j);
          ngb3 = 0; ngb4 = 0;
          ngb5 = 0; ngb6 = fHex(i+1,j);
        else if j=1 && j<cols && i>1 && i<rows &&
          i is even
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j+1);
          ngb3 = fHex(i-1,j); ngb4 = fHex(i,j-1);
          ngb5 = fHex(i+1,j); ngb6 = fHex(i+1,j+1);
        else if j=1 && j<cols && i>1 && i<rows &&
          i is odd
          ngb1 = fHex(i,j+1); ngb2 = fHex(i-1,j);
          ngb3 = fHex(i-1,j-1); ngb4 = fHex(i,j-1);
          ngb5 = fHex(i+1,j-1); ngb6 = fHex(i+1,j);
        fHexPadded(i,j) = fHex(i,j);
    end
end
```

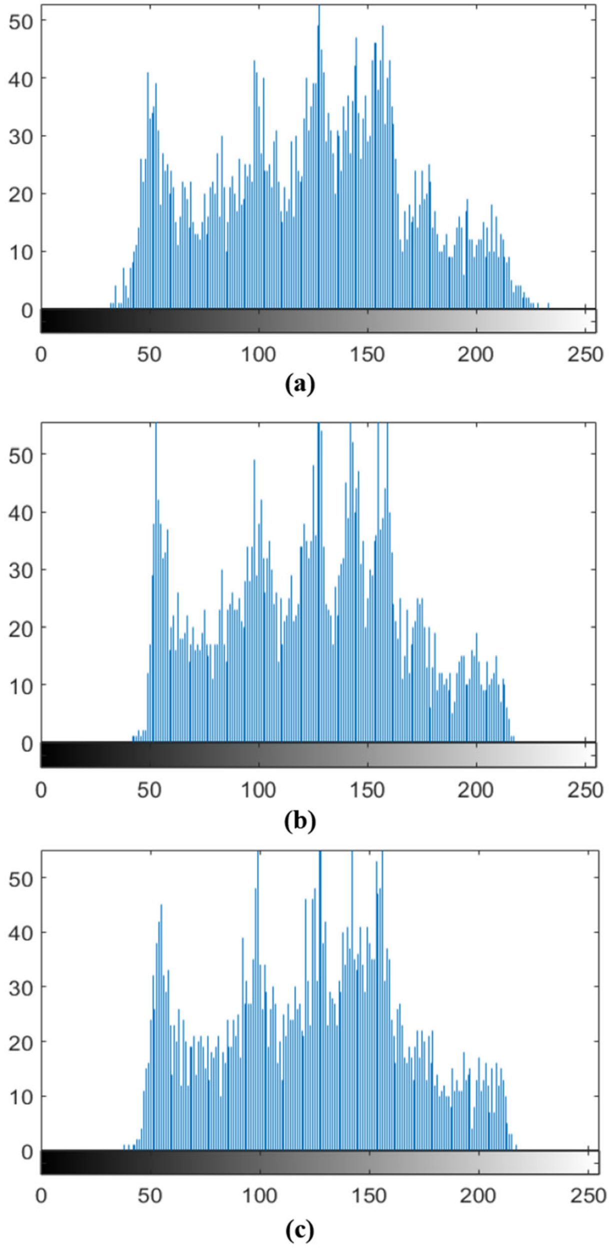


Fig. 13 Histograms of the images in Fig 12 **(a)** Histogram of the original Lena image **(b)** Histogram of Lena blurred by the weighted average kernel on the square domain **(c)** Histogram of Lena blurred by the weighted average kernel on the hexagonal domain

As clearly shown in Fig. 8, although any pixel in the square grid can be uniformly addressed to its neighbors, the indexes in the hexagonal grid vary depending on whether the reference pixel is in odd or even numbered rows. On the hexagonal grid, for pixel P_1 with the indices *(i,j)*, the neighboring pixels are P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16} and their indices are *(i,j + 1),*

Fig. 14 **(a)** The original Lena image **(b)** Lena blurred by the Gaussian kernel on the square domain **(c)** Lena blurred by the Gaussian kernel on the hexagonal domain

(i-1,j+ 1), (i-1,j), (i,j-1), (i+ 1,j), (i+ 1,j+ 1) respectively. However, for pixel P_2 with the indices *(i,j)*, the neighboring pixels are P_{21}, P_{22}, P_{23}, P_{24}, P_{25}, P_{26} and their indices are *(i,j+ 1), (i-1,j), (i-1,j-1), (i,j-1), (i+ 1,j-1), (i+ 1,j)* respectively. The conditional neighborhood definition and assignment for the pixel P_r is given in Eq.(7):

$$\begin{Bmatrix} P_{r1} = I_{i,j+1} \\ P_{r2} = I_{i-1,j+1} \\ P_{r3} = I_{i-1,j} \\ P_{r4} = I_{i,j-1} \\ P_{r5} = I_{i+1,j} \\ P_{r6} = I_{i+1,j+1} \end{matrix} \quad , \quad i \text{ is odd} \end{Bmatrix}$$

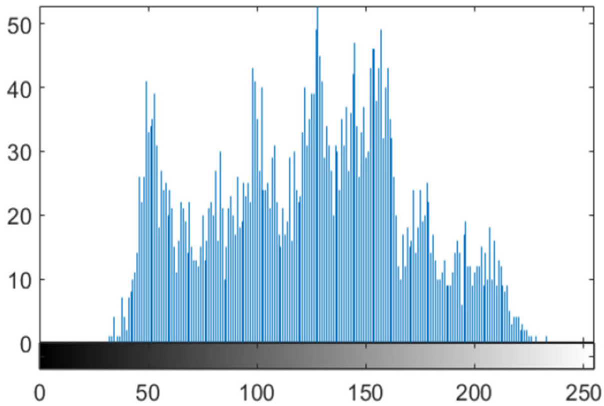
$$\begin{Bmatrix} \begin{matrix} P_{r1} = I_{i,j+1} \\ P_{r2} = I_{i-1,j} \\ P_{r3} = I_{i-1,j-1} \\ P_{r4} = I_{i,j-1} \\ P_{r5} = I_{i+1,j-1} \\ P_{r6} = I_{i+1,j} \end{matrix} \quad , i \text{ is even} \end{Bmatrix} \tag{7}$$

where *i, j* and *I* denote the row index, column index and intensity value of a pixel respectively.

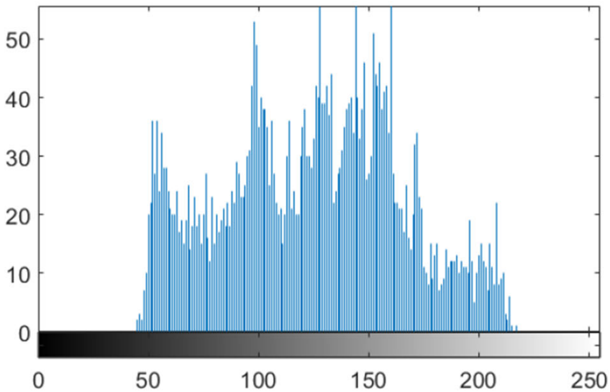
For pixels located at the sides and corners of the image, neighbor assignment becomes more challenging. Because, the location of the pixel should be examined in more varieties than the two alternatives for the pixels in the middle of the image. For the pixels positioned at this edge, the neighbor assignment process involves a padding process, because in the case of zero padding for missing neighbors, a zero value must be assigned. And this operation varies depending on whether the pixel is in the leftmost column or in the top row, or in the middle and in the top row, etc. Algorithm 1 describes all possibilities and actions taken accordingly.

2.3 Blurring operation on hexagonal domain

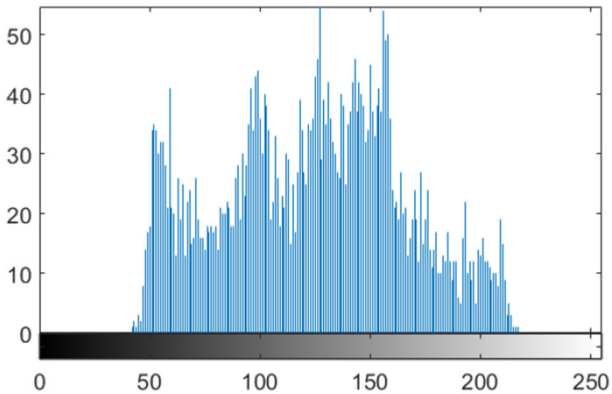
Blur is one of the most common processes in image processing and is actually an example of low pass filtering. Blurring is used before an edge detection or during a noise removal



(a)



(b)



(c)

Fig. 15 Histograms of the images in Fig 14 **(a)** Histogram of the original Lena image **(b)** Histogram of Lena blurred by the Gaussian kernel on the square domain **(c)** Histogram of Lena blurred by the Gaussian kernel on the hexagonal do-main

operation. When blurring is applied to an image, at points where the color transition occurs quickly, they are actually edges, the transition event occurs smoothly, not suddenly. This

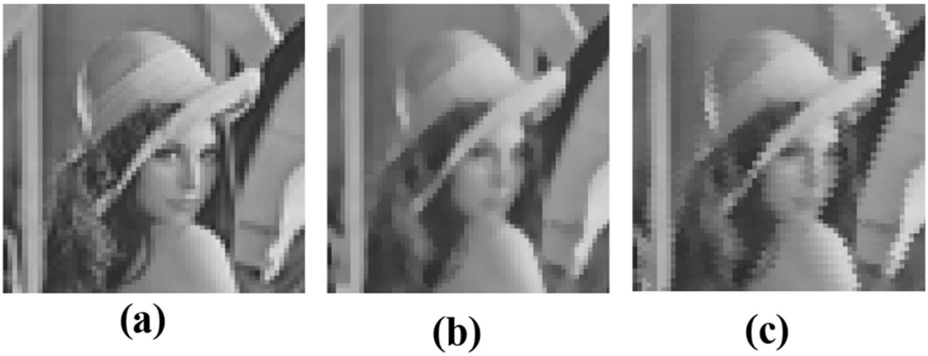


Fig. 16 **(a)** The original Lena image **(b)** Lena blurred by the median operator on the square domain **(c)** Lena blurred by the median operator on the hexagonal domain

eliminates sharp color transitions. This process implicitly eliminates external pixels, which are representative of noise. There are a number of methods used for blurring. Some are linear, some are non-linear. The linear ones involve convolution of the image with a kernel matrix. Mean, weighted mean and Gaussian are representative of linear ones, while median is representative of non-linear filters. Figure 9 shows the mean, weighted average and Gaussian kernels on square and hexagonal domains.

Blurring is sometimes called smoothening or low-pass filtering. Therefore, the mean kernel is also referred to as the low-pass filtering kernel. The result of the blurring operation on a sample image as well as histograms by applying the mean, weighted average, Gaussian and median filters on both the square and hexagonal domains are illustrated in Fig. 10-17 respectively.

One of the application areas of blurring is the noise elimination. Noise appears in the image as high-frequency components. Therefore, the blurring operation discards the high-frequency components and permits the low-frequency ones to pass. Thus, blurring is also called as the low-pass filtering. The application of the basic filters described above is effective in eliminating the effect of noise. Figure 18 illustrates the performance of the abovementioned filters under salt-pepper noise on both the square and hexagonal domains.

2.4 Sharpening on hexagonal domain

Details and edges are highly important in human perception. That is, when an individual looks at an image, the vision system inherently focuses on fine details and edges, which play a key role in perception. The visual quality of the image degrades if the details of the image are reduced or minimized. Image sharpening is used to make the edges and fine details distinct. Image sharpening clarifies these details by enhancing contrast between dark and bright regions. In fact, the details in an image are the high-frequency components. Applying a high-pass filter to an image makes the details more salient. High-pass filtering can be implemented by convolving an image by a high-pass filtering kernel. Figures 17, 18 and 19 show sample image sharpening kernels for square and hexagonal

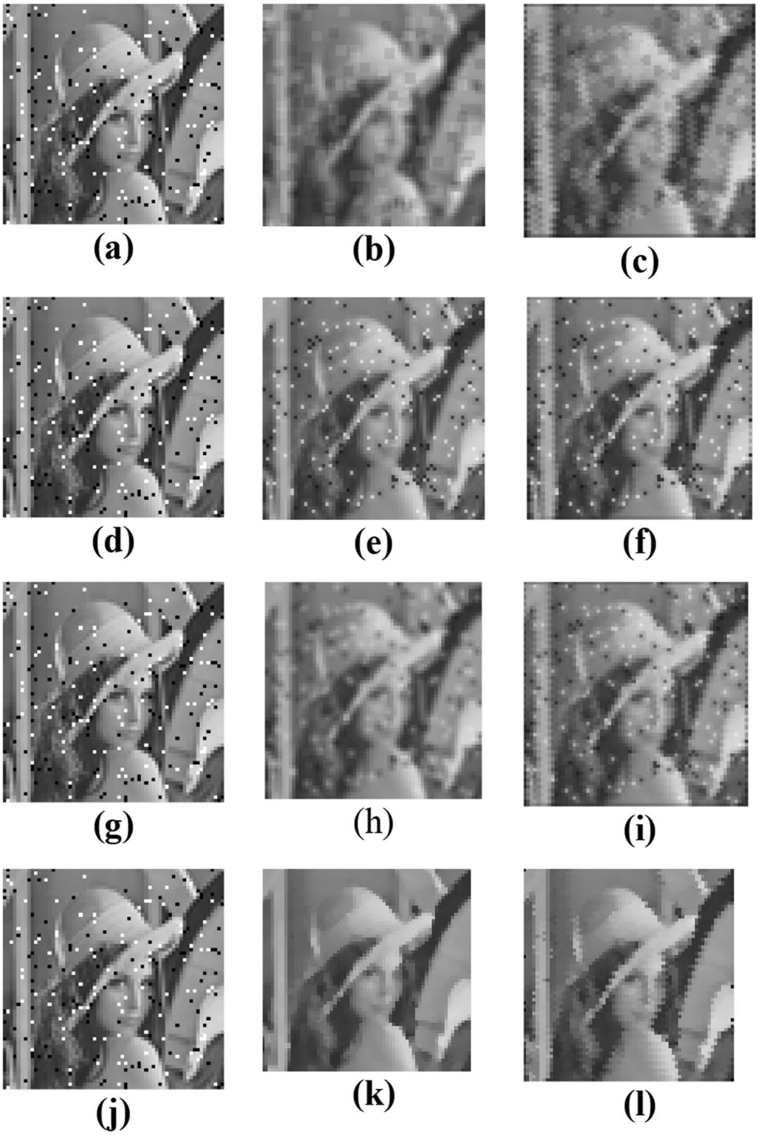
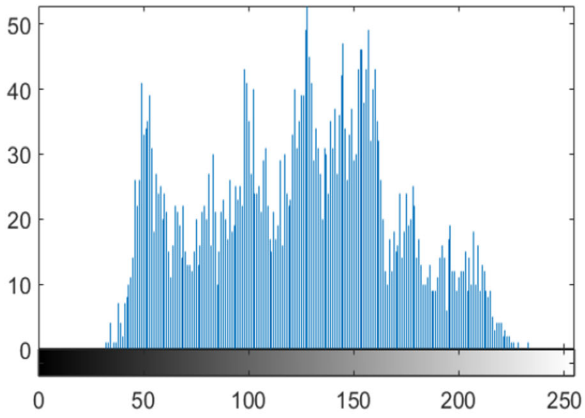


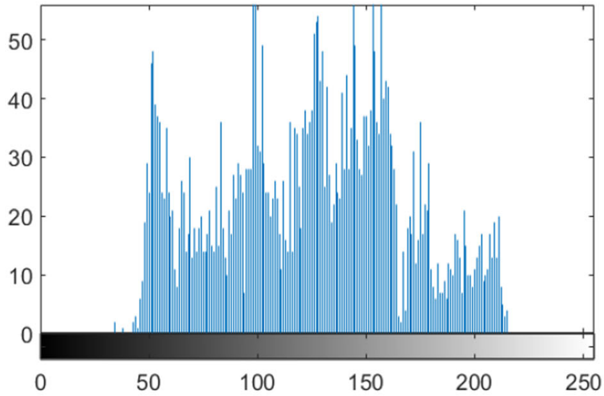
Fig. 17 Histograms of the images in Fig 16 **(a)** Histogram of the original Lena image **(b)** Histogram of Lena blurred by the median operator on the square domain **(c)** Histogram of Lena blurred by the median operator on the hexagonal domain

domains, as well as the results of the sharpening process and histograms of the sharpened images respectively.

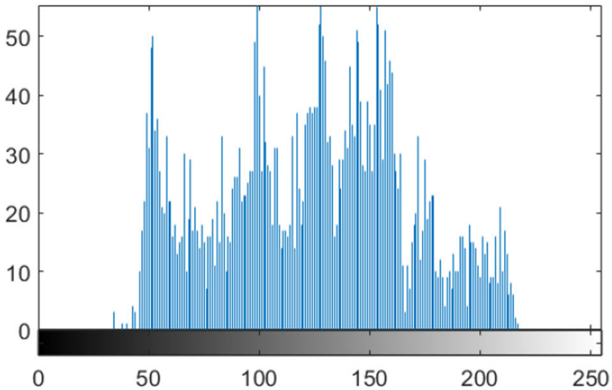
Obviously, the same sharpening process can be achieved by applying six multiplication operations on the hexagonal domain rather than implementing eight multiplication operations on the square domain. The gain (G_C) in terms of reducing the computational



(a)



(b)

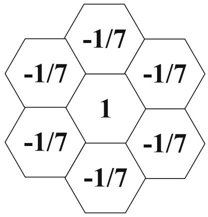


(c)

Fig. 18 **(a,d,g,h)** The salt-pepper noisy Lena image **(b,e,h,k)** The **s**alt-pepper noisy Lena image that is filtered on the square domain by mean, weighted average, Gaussian and median filters respectively **(c,f,i,l)** The salt-pepper noisy Lena image that is filtered on the hexagonal domain by mean, weighted average, Gaussian and median filters respec-tively

Fig. 19 Image sharpening kernels on square and hexagonal domains

-1/9	-1/9	-1/9
-1/9	1	-1/9
-1/9	-1/9	-1/9



complexity for a single pixel is ~1,33. For an image of size $m \times n$, the total G_C is $1, 33 \times m \times n$.

2.5 Edge detection on hexagonal domain

Edge detection is one of the important and fundamental topics in image processing to find boundaries of the objects in an image. Edge detection is especially used for image segmentation and feature extraction to be applied in image processing, computer and machine vision. Features that discriminate objects or regions from each other are extracted from edges. The importance of edge detection has become more than ever depending on the improvements in applications and areas that require more discriminative features from images. So far, a number of edge detection methods have been proposed, including Sobel, Canny, Prewitt and Roberts. Edge detection is based on distinguishing sudden density changes in the horizontal, vertical and diagonal axes. On the square domain, there are three main directions where intensity changes can occur, *0*°, *45*° and *90*°. The counterparts of these directions on the hexagonal plane are *0*°, *60*° and *120*°. Thus, a hexagonal domain edge detector should reveal the intensity changes in these directions. Figure 20 shows two different sets of hexagonal edge detector operators (a,b) as well as the square domain Sobel operator (c). At each set, three operators from left to right are illustrated for *0*°, *60*° and *120*° respectively.

As seen in Fig. 21, competitive results are achieved with sets of hexagonal edge detectors. Multiplication is not required in HexEd2. To complete the edge detection process, three subtraction and two addition operations are sufficient. However, the Sobel edge detection operator requires four multiplications, four additions and six subtractions. Table 1 presents the operational requirements of the above-mentioned edge detectors.

Fig. 20 **(a)** Original image **(b)** Image sharpened on the square domain **(c)** Image sharpened on the hexagonal domain

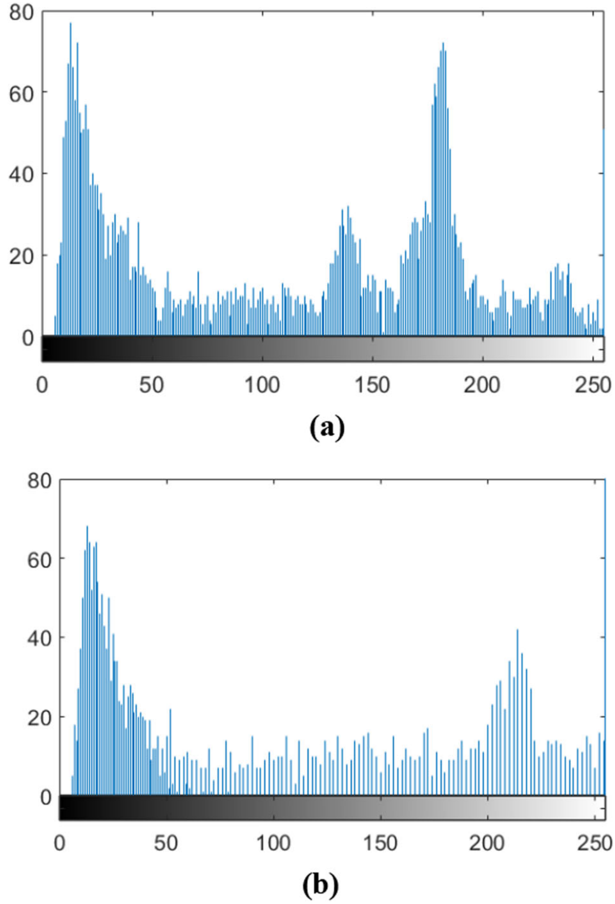


Fig. 21 Histograms of the images in Fig. 18 **(a)** Histogram of the image sharpened on the square domain **(b)** Histogram of the image sharpened on the hexagonal domain

The total arithmetic complexities of Sobel, HexEd1 and HexEd2 are given in Eq. (8–10), respectively.

$$AC_{Sobel} = (4 \times C_A) + (6 \times C_S) + (4 \times C_M) \tag{8}$$

$$AC_{HexEd1} = (6 \times C_A) + (9 \times C_S) + (12 \times C_M) \tag{9}$$

Table 1 Operational requirements of Sobel, HexEd1 and HexEd2 edge detectors

Method	Addition	Subtraction	Multiplication
Sobel	4	6	4
HexEd1	6	9	12
HexEd2	–	3	0

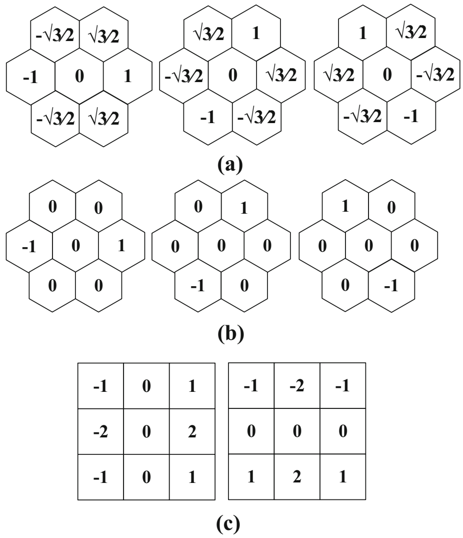


Fig. 22 **(a)** The hexagonal edge detector operator set 1 (HexEd1) **(b)** The hexagonal edge detector operator set 2 (HexEd2) **(c)** The square domain Sobel edge detector set

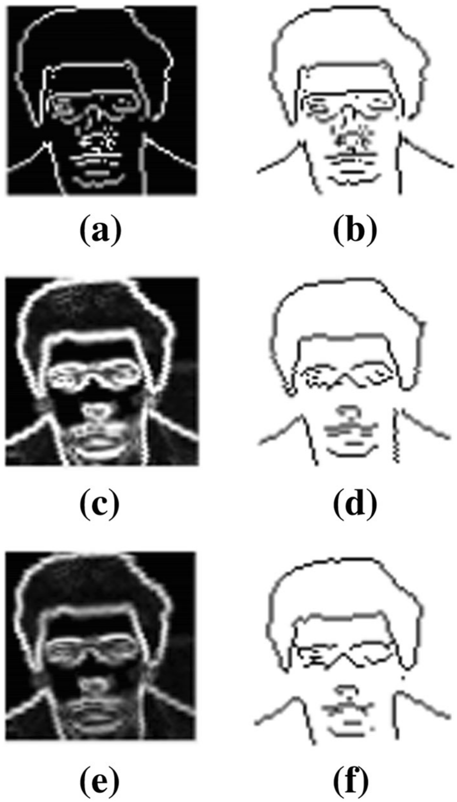


Fig. 23 **(a)** Original image **(b)** Sobel edge detection result on square domain **(c)** Result of the edge detection process by HexEd1 **(d)** Complement of the resulting image given at c **(e)** Result of the edge detection process by HexEd2 **(d)** Complement of the resulting image given at e

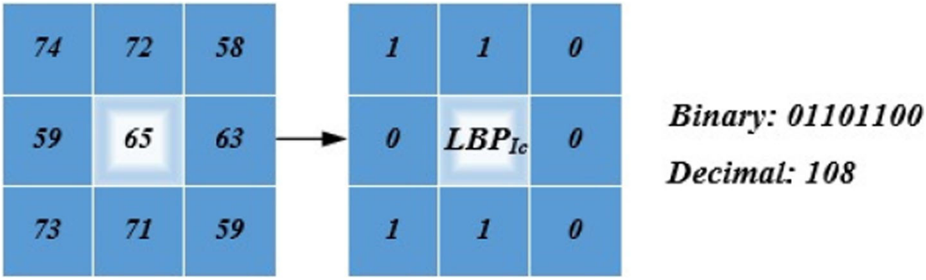


Fig. 24 An exemplary calculation of the basic LBP

$$AC_{HexEd2} = (3 \times C_S) \tag{10}$$

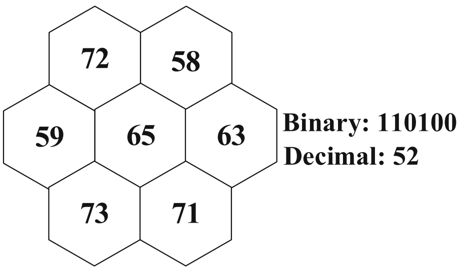
where AC_{Sobel}, AC_{HexEd1}, AC_{HexEd2}, C_A, C_S *and* C_M denote the total arithmetic computational complexity of Sobel, HexEd1, HexEd2, computational complexity of the addition, subtraction and multiplication operations respectively.

2.6 Feature extraction and recognition on hexagonal domain

Recognition is one of the most important areas in which image processing is applied. As is known, machine learning methods that are used to imitate the human perception, relay it to the computer system and handle it autonomously. Learning methods classify images or objects according to their characteristic features. Hence, these features should be specific to those objects to discriminate them efficiently from the others. The face is one of the most important biometrics used in many areas of life, such as surveillance, security and law. Face recognition is the art of discriminating individuals by using the facial data. Its high distinctive performance, as well as the possibility of its collection and processing in real time without any discomfort and physical contact through devices such as cameras, makes the face data one of the leading biometrics [8, 13, 25].

Face recognition descriptors are categorized as holistic [6] and local. Local descriptors pose better performance than the holistic ones regarding robustness against rotation, noise and illumination factors [27]. Plenty of local descriptors (LBP [1], LGBP [58], CS-LBP [21], GV-LBP [27], LDP [24], LJBPW [12], RIMFRA [7], LDGP [9], LPQ [55], LDNP [41, 42], HoG [11], LTP [51], Gabor [33, 57]) have been proposed and appear in the literature.

Fig. 25 An exemplary calculation of LBP on hexagonal domain



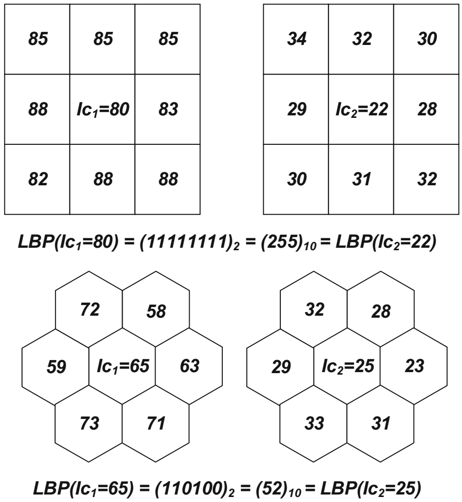


Fig. 26 Identical descriptor values assigned to different patterns on both square and hexagonal domains

LBP is one of the basic and pioneering local descriptors that has inspired many followers. The basic LBP considers the intensity relationship between a pixel and its adjacent neighbors. If a neighboring pixel's intensity value is greater than the reference pixel, than a *0*, otherwise a *1* is assigned to the corresponding digit of the new LBP value, which denotes the intensity magnitude relationships between the neighboring pixels and the reference pixel. LBP of a reference pixel c, considering its P equally apart neighbors on a circle with radius R, is calculated as follows [5]:

$$LBP_{P,R}(c) = \sum_{P=0}^{P-1} s(I_c - I_P)2^p \tag{11}$$

where I_c and I_P denote the intensity values of the reference pixel and the P^{th} neighboring pixel that is considered respectively. The function *s(x)* identifies the coefficient of the corresponding binary digit and defined as:

$$s(x)\begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{12}$$

LBP is invariant to monotonic gray-scale changes due to the invariance of the function *s(x)* against monotonic gray-scale changes [40]. An exemplary demonstration of the basic LBP is given in Fig. 22:

Table 2 The recognition accuracy analysis by means of supervised training conducted on Face94, CAS-PEAL-R1, JAFFE and ORL datasets

Method	Accuracies				
	Face94	YALE	CAS-PEAL-R1	JAFFE	ORL
Hex_LBP	1000	0,803	0,889	0,940	0,713
LBP	1000	0,807	0,912	0,960	0,825

2^p possible different patterns can be calculated. Following the calculation of the LBP values for each pixel, the texture of the image (I_{mxn}) is defined by considering the probability distributions of these LBP values on a histogram, as follows:

$$H(LBP_k) = \sum_{i=1}^{m} \sum_{j=1}^{n} \delta\{k, LBP(i,j)\} \tag{13}$$

where $\delta\{.\}$ denotes the Kroneck product function [56].

An equivalent of LBP on hexagonal domain is given in Fig. 23.

Unlike normal square-domain-LBP which produces $2^8–1 = 255$ different identifiers, hexagonal LBP produces only $2^6–1 = 63$ distinct values. Besides, the basic LBP produce same descriptor values for different patterns unless the intensity value of the reference pixel is involved in during the descriptor calculation stage as illustrated in Fig. 24.

As shown in Fig. 24, even there are two different patterns, the resulting descriptor value is produced for both. If the intensity of the reference pixel is involved in during the calculation of the local descriptor value, the abovementioned challenge is easily overcome. A possible solution for the hexagonal domain is given in Eq. (14).

$$Hex\_LBP_{I_c} = mod\ ((B_{I_c} \times (I_c/63)), 63) \tag{14}$$

Thus, the new descriptor (Hex_LBP) values for the example given in Fig. 24, become as $Hex\_LBP_{65} = 41$, $Hex\_LBP_{25} = 52$ respectively. Successfully handling of this similar descriptor assignment challenge significantly affects the recognition accuracy performance. Because, although the matrices given above represent two different local regions of an image, if it is not considered and handled properly, the same descriptor is going to be produced for these two different patterns.

To analyze the face recognition accuracy performance, a number of simulations are conducted on the well-known basic datasets, namely, Face94 [28], ORL [45], JAFFE [31], Yale (http://vision.ucsd.edu/content/yale-face-database), CAS-PEAL-R1 [14]. To ensure uniformity, some pre-treatment is applied to each image. Each image is initially scaled to 64 × 64. After the scaling step, facial extraction is performed using the Viola Jones [52] algorithm to eliminate the effect of unnecessary background factors. The recognition accuracy performance is measured as presented in Table 2. Obviously, although the range of feature values falls by a quarter, the recognition accuracy performance is very close to the performance of the ordinary LBP.

3 Conclusion

The hexagonal-pixel-based image processing (HIP) is claimed to have significant advantages when compared to the ordinary square-pixel-based image processing (SIP) for decades. However, since all the mathematical, software and hardware background that have been used since the date of beginning of the image processing science have based on square domain, HIP has not gained the attention, which it deserves. There is no standardized library nor package to process an image hexagonally. Therefore, a framework is developed for use in future research on HIP, in which the hexagonal equivalents of some of the basic processes of ordinary image processing are presented in this article. The most prominent benefit of HIP against SIP is the gain provided in terms of computing complexity and memory area. Since the same information and operations that are performed in SIP can be implemented by a less number of steps, the burden of processing and memory occupation is alleviated. As presented in the article, the contours of a face can be extracted by only processing six neighbor relationship rather than eight like done is SIP. Furthermore, operations such as blurring, sharpening and noise elimination can be also handled remarkably by utilizing less number of operations that conclude at tiring the processor less. Lastly, the face recognition operation can be achieved at very close rates to the SIP by expressing each feature by a less number of bits. For future work, it is intended to elaborate on the face recognition in HIP and realize the hexagonal equivalents of the state-of-the-art descriptors that have been already presented for SIP.

References

1. Ahonen T, Hadid A, Pietikainen M (2004) Face recognition with local binary patterns. Proceedings of the 8th European Conference on Computer Vision: 469–481
2. Bell SCM, Holroyd FC, Mason DC (1989) A digital geometry for hexagonal pixels. Image Vis Comput 7: 194–204
3. Burt PJ (1980) Tree and pyramid structures for coding hexagonally sampled binary images. Comput Graph Image Process 14:271–280
4. Cevik N (2019) Face recognition by grey-level co-occurrence matrices in hexagonal digital image processing. Turkish Stud – Inform Technol Appl Sci 14(2):149–165
5. Cevik N, Cevik T (2018) DLGBD: a directional local gradient based descriptor for face recognition. Multimed Tools Appl 78:1–20. https://doi.org/10.1007/s11042-018-6967-4
6. Cevik N, Cevik T (2019) Novel high-performance holistic descriptor for face retrieval. Pattern Anal Applic: 1–13. doi:https://doi.org/10.1007/s10044-019-00803-5
7. Cevik T, Cevik N (2019) RIMFRA: rotation-invariant multi-spectral face retrieval approach by using orthogonal polynomials. Multimed Tools Appl 78:1–31. https://doi.org/10.1007/s11042-019-07816-6
8. Cevik N, Cevik T, Gurhanli A (2019) A novel multispectral face descriptor using orthogonal Walsh codes. IET Image Process 13(7):1097–1104. https://doi.org/10.1049/iet-ipr.2018.6423
9. Chakraborty S (2017) Satish Kumar Singh, pa-van Chakraborty, “local directional gradient pat-tern: a local descriptor for face recognition”. Multimed Tools Appl 76:1201–1216
10. Curcio CA, Sloan KR, Packer O, Hendrickson AE, Kalina RE (1987) Distribution of cones in human and monkey retina: individual variability and radial asymmetry. Science 236(4801):579–582
11. Dahmane M, Meunier J (2011) Emotion recognition using dynamic gridbased HoG features. IEEE Int Conf Autom Face Gesture Recognit Workshops (FG): 884–888
12. Dan Z, Chen Y, Yang Z, Wu G (2014) An improved local binary pattern for texture classification. Optik 125:6320–6324
13. Dubey SR (2017) Local directional relation pattern for unconstrained and robust face retrieval. arXiv: 1709.09518 [cs.CV]
14. Gao W, Cao B, Shan S, Chen X, Zhou D, Zhang X, Zhao D (2008) The CAS-PEAL large-scale Chinese face database and baseline evaluations. IEEE Trans Syst Man Cybernet (Part A) 38(1):149–161
15. Gibson L, Lucas D (1982) Vectorization of raster images using hierarchical methods. Comput Graph Image Process 20:82–89
16. Gibson L, Lucas D (1982) Spatial data processing using generalized balanced ternary. Proceedings of PRIP 82. IEEE Computer Society Conference on Pattern Recognition and Image Processing: 566–571
17. Gibson L, Lucas D (1986) Pyramid algorithms for automated target recognition. Proceedings of the IEEE 1986 National Aerospace and Electronics Conference: 215–219
18. Hales TC (2000) Cannonballs and honeycombs. Not Am Math Soc 47(4):440–449
19. Hales TC (2001) The honeycomb conjecture. Discrete Comput Geometry 25:1–22
20. He X, Jia W (2005) Hexagonal structure for intelligent vision. 1st International Conference on Information and Communication Technologies: 52–64
21. Heikkilä M, Pietikäinen M, Schmid C (2009) Description of interest regions with local binary patterns. Pattern Recogn 42(3):425–436
22. Her I (1995) Geometric transforms on the hexagonal grid. IEEE Trans Image Process 4:1213–1222
23. Her I, Yuan C-T (1994) Resampling on a Pseudohexagonal grid. CVGIP: Graph Models Image Process 56: 336–347
24. Jabid T, Kabir MH, Chae O (2010) Robust facial expression recognition based on local directional pattern. ETRI J 32(5):784–794
25. Jafri R, Arabnia HR (2009) A survey of face recognition techniques. J Inform Process Syst 5(2):41–68

26. Knuth DE (1969) The art of computer programming: Seminumerical Algorithms, Vol. 2. Addison Wesley
27. Lei Z, Liao S, Pietikäinen M, Li SZ (2011) Face recognition by exploring information jointly in space, scale and orientation. IEEE Trans Image Process 20(1):247–256
28. Libor S (2000) Face recognition data
29. Lucas D (1979) A multiplication in N-space. IEEE Trans Image Process 74:1–8
30. Lucas D, Gibson L (1991) Template decomposition and inversion over hexagonally sampled images. Image Algebra Morphol Image Process II 1568:257–263
31. Lyons MJ, Akemastu S, Kamachi M, Gyoba J (1998) Coding facial expressions with gabor wavelets. 3rd IEEE International Conference on Automatic Face and Gesture Recognition: 200-205
32. Mahesh B, Pearlman W (1989) Image coding on a hexagonal pyramid with noise spectrum shaping. Proc Int Soc Optic Eng 1199:764–774
33. Melendez J, Garcia MA, Puig D (2008) Efficient distance-based per-pixel texture classification with Gabor wavelet filters. Pattern Anal Appl 11(3):365–372
34. Mersereau RM (1979) The processing of hexagonally sampled two-dimensional signals. Proc IEEE 67: 930–949
35. Middleton L, Sivaswamy J (2005) Hexagonal image processing, a practical approach. Springer
36. Miller E (1999) Alternative Tilings for improved surface area estimates by local counting algorithms. Comput Vis Image Underst 74(3):193–211
37. Nel A (1989) Hexagonal image processing. COMSIG, IEEE
38. Overington I (1992) Computer vision: a unified, biologically-inspired approach. Elsevier Science Publishing Company
39. Pirenne MH (1967) Vision and the eye, 2nd edn. Chapman and Hall, London
40. Qi X, Xiao R, Li C-G, Qiao Y, Guo J, Tang X (2014) Pairwise rotation invariant co-occurrence local binary pattern. IEEE Trans Pattern Anal Mach Intell 36(11):2199–2213
41. Ramirez Rivera A, Castillo R, Chae O (2013) Local directional number pattern for face analysis: face and expression recognition. IEEE Trans Image Process 22(5):1740–1752
42. Rivera AR, Chae O (2015) Spatiotemporal directional number transitional graph for dynamic texture recognition. IEEE Trans Pattern Anal Mach Intell 37(10):2146–2152
43. Rosenfeld A (1970) Connectivity in digital pictures. J Assoc Comput Mach 17(1):146–160
44. Rosenfeld A, Pfaltz JL (1968) Distance functions on digital pictures. Pattern Recogn 1:33–61
45. Sarasota FL (1994) Proceedings of 2nd IEEE workshop on applications of computer vision
46. Serra J (1986) Introduction to mathematical morphology. Computer Vision, Graphics Image Process 35: 283–305
47. Sheridan P, Hintz T, Alexander D (2000) Pseudo-invariant image transforms on a hexagonal lattice. Image Vis Comput 18:907–917
48. Snyder W, Qi H, Sander W (1999) A coordinate system for hexagonal pixels. Proc Int Soc Optic Eng 3661: 716–727
49. Staunton R (1999) Hexagonal sampling in image processing. Adv Imaging Electro Phys 107:231–307
50. Stevenson RL, Arce GR (1985) Binary display of hexagonally sampled continuous-tone images. J Opt Soc Am A 2:1009–1013
51. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans Image Process 19(6):1635–1650
52. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57:137–154
53. Watson AB, Ahumada AJ (1989) A hexagonal orthogonal-oriented pyramid as a model of image representation in the visual cortex. IEEE Trans Biomed Eng BME-36:97–106
54. Wuthrich CA, Stucki P (1991) An algorithmic comparison between squareand hexagonal-based grids. CVGIP: Graph Models Image Process 53:324–339
55. Yang S, Bhanu B (2011) Facial expression recognition using emotion avatar image. IEEE Int. Conf Autom Face Gesture Recognit Workshops (FG): 866–871
56. Yang B, Chen S (2013) A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image. Neurocomputing 120:365–379
57. Yin QB, Kim JN (2008) Rotation-invariant texture classification using circular Gabor wavelets based local and global features. Chin J Electron 17(4):646–648
58. Zhang WC, Shan SG, Gao W, Zhang HM (2005) Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. Proceedings of the 10th IEEE International Conference and Computer Vision: 786–791

Taner Cevik He received the B.Sc. degree in computer engineering from Istanbul Technical University, Istanbul in 2001, and obtained his Master of Science degree in computer engineering from Fatih University, Istanbul in 2008. Following, completed Ph.D. at Istanbul University in 2012. He joined the Computer Engineering Department, Aydin University in 2017 and continues to work as an Associate Professor. His research interests are Image Processing and Wireless Multimedia Sensor Networks. Mailing Address: Ataturk Mahallesi, Gazi Mustafa Kemal Caddesi Deran Sokak No: 6/5 Büyükçekmece Istanbul Turkey

Mustafa Fettahoglu He received the B.Eng. degree in Applied Informatics from Universität Salzburg, Austria in 2011 and obtained his Diplom-Ingenieur degree in Applied Informatics from Universität Salzburg, Austria in 2012. He is currently a Ph.D student at Aydin University. His research interests are Image ProcessinG, Artificial Intelligence and Human Computer Interaction. Mailing Address: Baglarbasi Mahallesi, Gümüshane Üniversitesi Lojmanlari A-Blok No:11 Gümüshane, Turkey

Nazife Cevik She obtained the PhD. degree from Istanbul University in 2012. She joined the Computer Engineering Departmento of Istanbul Arel University in 2015 and continues to work as an Assistant Professor. Her research interests are Image Processing and Machine Learning and Bioinformatics. Mailing Address: Istanbul Arel University Tepekent Department of Computer Engineering Büyükçekmece Istanbul Turkey

Serdar Yilmaz He received the B.Sc. degree in electronics engineering from Istanbul University, Istanbul in 1999, and obtained his Master of Science degree in electronics engineering from Fatih University, Istanbul in 2003. Following, completed Ph.D. electronics and communication at Yildiz Technical University in 2012. He joined the İstanbul University Vocational School of Technical Sciences. His research interests are Image Processing, Speech Processing and Infrared Imaging. Mailing Address: Senlikkoy Mahallesi, Hurriyet Caddesi Ay Sokak No: 12/7 Bakırköy Istanbul Turkey