

# About Edge Detection in Digital Images

Miroslav HAGARA, Peter KUBINEC

<sup>1</sup>Inst. of Electronics and Photonics, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia

miroslav.hagara@stuba.sk, peter.kubinec@stuba.sk

Submitted October 14, 2018 / Accepted October 15, 2018

**Abstract.** *Edge detection is one of the most commonly used procedures in digital image processing. In the last 30–40 years, many methods and algorithms for edge detection have been proposed. This article presents an overview of edge detection methods, the methods are divided according to the applied basic principles. Next, the measures and image database used for edge detectors performance quantification are described. Ordinary users as well as authors proposing new edge detectors often use Matlab function without understanding it in details. Therefore, one section is devoted to some of Matlab function parameters that affect the final result. Finally, the latest trends in edge detection are listed. Picture Lena and two images from Berkeley segmentation data set (BSDS500) are used for edge detection methods comparison.*

## Keywords

Image processing, edge detection, gradient operator, morphological operator, fractional differentiation, Berkeley segmentation data set (BSDS), NYU depth dataset, Matlab

## 1. Introduction

Since there is a direct link between the edges in the image and the physical properties of the scene (the edges in the picture mostly represent the outline of the scene objects), a lot of scene information can be obtained from the edges detected in the image. Therefore, edge detection is one of the key operations in image processing. Mostly it is used for general goals such as image segmentation, boundary detection or object recognition. But edge detectors are also often used for special tasks like fire detection [1], car-license plate detection [2], fingerprint identification [3], synthetic aperture radar (SAR) image processing [4–6], lunar surface crater topology [7], polyp detection in colonoscopy [8], etc.

Edge detectors convert a gray-scale or color image to a binary image - edge map that preserves much of the information content of the original image. Most methods can be divided into groups according to the basic principle they use. A brief description of the frequently applied edge

detection methods is presented in Sec. 2. In the following two sections, the measures and image database used for edge detectors performance quantification are described.

Matlab function for edge detection is often applied as part of a larger process or it is used by authors to calculate edge maps by older algorithms for comparison with their new proposed methods. But both, users as well as authors of new edge detectors often use Matlab function without understanding it in details. They usually use this function with default settings but there are some function parameters which can be changed. The effect of these parameters on the final result is described in a separate section. In the final section the latest trends in edge detection are listed with focusing on edge detectors based on fractional differentiation. Picture Lena and some images from Berkeley segmentation data set (BSDS500) are used for computations.

## 2. Edge Detection Methods

The methods given in the next two subsections do not directly create an edge map, but their output is a gradient image (edge strength image) and the edge map of it is obtained using one of the methods listed in Sec. 2.3. The whole process is illustrated in Fig. 1.

### 2.1 Gradient Edge Detectors

These detectors are based on the fact that the value of the image function  $f(x,y)$  changes significantly at the edge point. The change of function is expressed by its gradient  $\nabla f$ , a two-dimensional vector formed by the partial derivations

$$\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right). \quad (1)$$

Gradient magnitude  $|\nabla f|$  and orientation  $\psi$  can be computed from

$$|\nabla f| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}, \quad (2)$$

$$\psi = \arctg \left( \frac{\partial f}{\partial x} / \frac{\partial f}{\partial y} \right). \quad (3)$$

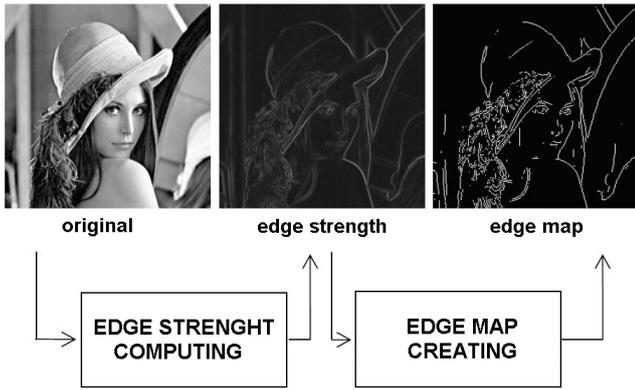


Fig. 1. Two-step process of edge detection.

Pixels with high gradient magnitude are considered as edge points. The edge direction at a given point is perpendicular to the direction of the pixel gradient. Gradient operators are implemented using convolution masks. The most used operators are [9]:

Prewitt

$$h_{Px} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_{Py} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad (4)$$

Sobel

$$h_{Sx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_{Sy} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad (5)$$

Robinson

$$h_{Rx} = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad h_{Ry} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \quad (6)$$

Kirsch

$$h_{Kx} = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}, \quad h_{Ky} = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}. \quad (7)$$

Then edge strength  $ES(x,y)$  can be computed [5] from

$$ES_s(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (8)$$

where

$$G_x(x,y) = h_x * f(x,y), \quad G_y(x,y) = h_y * f(x,y) \quad (9)$$

are the gradient components (gradient estimation) in the  $x$  and  $y$  directions, which are calculated by the convolution masks of the above mentioned operators. Edge strength images obtained by these operators are in Fig. 2. If it is not necessary to know the gradient direction but only the gradient magnitude, it can be estimated by Laplace's operator represented by masks  $h_{L4}$  for 4-connected and  $h_{L8}$  for 8-connected neighborhood [9]

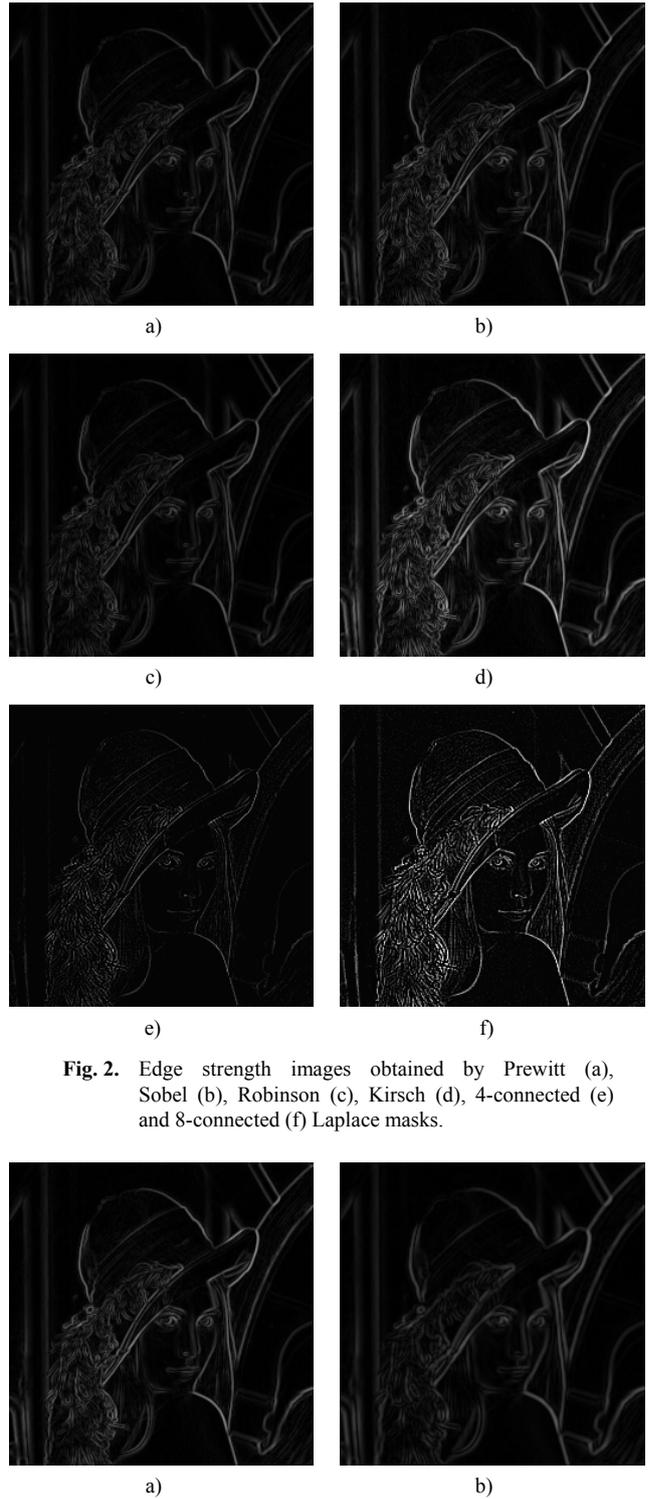


Fig. 2. Edge strength images obtained by Prewitt (a), Sobel (b), Robinson (c), Kirsch (d), 4-connected (e) and 8-connected (f) Laplace masks.

Fig. 3. Gradient magnitude obtained by  $5 \times 5$  Canny detector with sigma = 1 (a) and sigma = 2 (b).

$$h_{L4} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h_{L8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (10)$$

Edge strength images obtained by these operators are in Fig. 2. Canny [10] has chosen an analytic approach to the design of gradient operator for edge detection. He has

defined three criteria that an optimal operator for edge detection should meet. Description of a detector can be simplified as follows. First, the Gaussian filter is applied. Then the derivative operators (corresponding masks) are used to compute gradient magnitude and direction. Then non-maximum suppression is used to get edges with a one-pixel width. Finally, the threshold with hysteresis is applied to eliminate weak edges. Gradient magnitudes obtained by  $5 \times 5$  Canny detectors are in Fig. 3.

## 2.2 Morphologic Edge Detectors

Morphological edge detectors can be considered as non-linear gradient operators, because they produce edge strength, similar to linear gradient operators. However, edge strength is obtained by non-linear morphological operations. The erosion  $e(k,l)$  of the gray scale image  $f(k,l)$  by a gray-scale structuring element  $g(k,l)$  is defined [11]

$$e(k,l) = (f \ominus g)(k,l) = \min_{(i,j)} (f(k+i,l+j) - g(i,j)). \quad (11)$$

A simple edge detector producing edge strength  $G_e(k,l)$  can be defined [12]

$$G_e(k,l) = f(k,l) - e(k,l),$$

$$G_e(k,l) = f(k,l) - \min_{(i,j) \in N_4} (f(k+i,l+j)),$$

$$G_e(k,l) = \max_{(i,j) \in N_4} (f(k,l) - f(k+i,l+j)) \quad (12)$$

where  $N_4$  is 4-connected neighborhood. Similar detector can be defined also for 8-connected neighborhood. Edge strength images obtained by both detectors are in Fig. 4.

An edge detector can be defined also by dilation. Then edge strength  $G_d(k,l)$  is defined [12]

$$G_d(k,l) = d(k,l) - f(k,l),$$

$$G_d(k,l) = \max_{(i,j) \in N_4} (f(k+i,l+j) - f(k,l)) \quad (13)$$

where  $d(k,l)$  is a dilation of the gray scale image  $f(k,l)$  by a gray-scale structuring element  $g(k,l)$  [11]

$$d(k,l) = (f \oplus g)(k,l) = \max_{(i,j)} (f(k-i,l-j) + g(i,j)). \quad (14)$$

$G_e(k,l)$  and  $G_d(k,l)$  shift the edge, each in the opposite direction. Their combination  $G_{de}(k,l)$  eliminates this lack [12]

$$G_{de}(k,l) = \max(G_d(k,l), G_e(k,l)),$$

$$G_{de}(k,l) = \max_{(i,j) \in N_4} (|f(k+i,l+j) - f(k,l)|) \quad (15)$$

The above-mentioned edge detectors are very sensitive to noise in the processed image. To improve robustness



Fig. 4. Edge strength images obtained by 4-connected (a) and 8-connected (b) morphological edge detectors.

of morphological edge detectors a new approach was proposed based on combination of the biwindow configuration with weighted median filters and anisotropic morphological directional derivatives [13].

## 2.3 Edge Map Creating

In order to obtain an edge map, after gradient operator application, it is necessary to use post-processing, such as some of the thresholding methods, the non-maximum suppression or a three-module strategy.

### 2.3.1 Thresholding

The most common way to get edge map is the simple thresholding or thresholding with hysteresis. The simple thresholding is defined [5] as the rule  $g(x,y)$  assigning the value  $V_1$  or  $V_2$  depending on the threshold value  $t$  to each point  $(x,y)$  of the input image  $f(x,y)$

$$g(x,y) = \begin{cases} V_1 & \text{for } f(x,y) > t \\ V_2 & \text{for } f(x,y) \leq t \end{cases} \quad (16)$$

In practice, values  $V_1, V_2$  are often selected as the maximum or minimum of the image function, in the case of binary output image these values are 1 and 0. Edge strength values from gradient operators represent an input data for thresholding procedure.

Hysteresis thresholding for edge detection uses two thresholds  $t_u > t_d$ . Any pixel which has a value above the higher threshold  $t_u$  is marked as edgel (=edge pixel). Any pixel whose value lies between the higher and lower thresholds and is connected to an edgel is also marked as edgel. Other points are rejected.

Thresholding produces wide edges. To get edges with a one-pixel width it is necessary to apply a suitable thinning procedure, usually morphological algorithm [14]. Figure 5 shows the result of simple thresholding and morphological thinning.

Thresholding itself is not complicated, determining the appropriate threshold value, however, can pose quite a big problem. In the case of hysteresis thresholding, this is not critical, but for simple thresholding several methods for determining the optimal threshold have been developed.



**Fig. 5.** Edge strength from Sobel operator (Fig. 2b) after simple thresholding with threshold = 0.1 (a) and after morphological thinning (b).

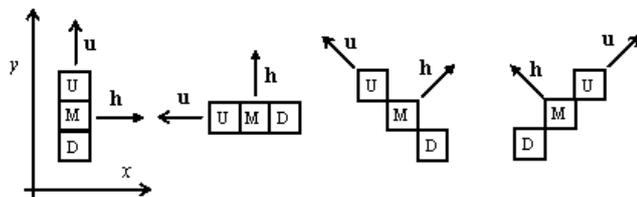
The triangle method [15] is based on the geometric shape of the edge strength histogram, but most methods are based on the statistical properties of the histogram. Some statistical methods for threshold determining are based on minimizing of the certain edge-histogram parameter, e.g. the overall entropy [16]. In another method optimal threshold maximize the separability of the resultant classes in gray levels. [17]. Sometimes, the thresholding is based on the specific features of the image to be processed (e.g. the radar image) [18]. The above-mentioned procedures produce a threshold value which is valid for the whole image. Sometimes it is more appropriate to use adaptive methods in which the image is divided into several smaller parts [19] or a small neighborhood is chosen for each point to calculate the threshold [20].

**2.3.2 Non-maximum Suppression**

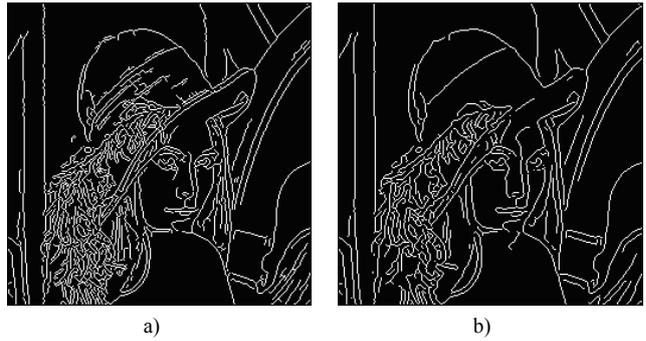
Then non-maximum suppression (NMS) is used to get edges with a one-pixel width. It was proposed by Canny [21] as part of his edge detector operator but in principle it may be used with gradient operators that produce two edge parameters - edge strength and direction.

The pixel  $(x,y)$  is denoted M and two adjacent points perpendicular to the direction  $\mathbf{h}$  of the edge (in the gradient direction  $\mathbf{u}$ ) are designated U and D (Fig. 6). U has coordinates  $(x,y+1)$ ,  $(x-1,y)$ ,  $(x-1,y+1)$  or  $(x+1,y+1)$ , D has coordinates  $(x,y-1)$ ,  $(x+1,y)$ ,  $(x+1,y-1)$  or  $(x-1,y-1)$ . If the edge strength at point U or D is greater than the edge strength at point M, then the point M is not considered as an edge point and the value of the output image function (edge map) at the point is set to the background value. Figure 7 shows two examples of using NMS.

Three-module strategy [22] is based on a similar principle as the non-maximum suppression. The first module



**Fig. 6.** Determination of points M, U, D for NMS.



**Fig. 7.** Edge map obtained by NMS within Canny operator. Hysteresis thresholding values are  $t_i = 0.14$  a  $t_d = 0.06$ , Filter parameters are sigma = 0.6 (a) and sigma = 1 (b).

computes local edge strength and direction, so any gradient operator that produces these two edge parameters can be used. The last module is a sequential process tracking edges. The originality of the method resides in the second module, which can be seen as a generalization of the non-maximum suppression algorithm.

**2.4 Zero-crossing Edge Detection**

In the zero-crossing edge detectors, edges are detected as the location where the second derivative of the (usually smoothed) image crosses zero. The approximations of the second derivative are more sensitive to noise than the first derivative approximations. A sufficiently robust solution to this problem is the convolution of the image with a suitable filter [23], for example, whose convolutional mask corresponds to the Gaussian function  $G(x,y)$ . Then Laplacian  $\nabla^2$  is applied to the filtered image. Due to the linearity of derivation and convolution, the sequence of operations can be changed

$$\nabla^2(G(x,y) * f(x,y)) = (\nabla^2 G(x,y)) * f(x,y). \quad (17)$$

Operator that includes the second derivation as well as Gaussian filter is denoted Laplacian of Gaussian (LoG)

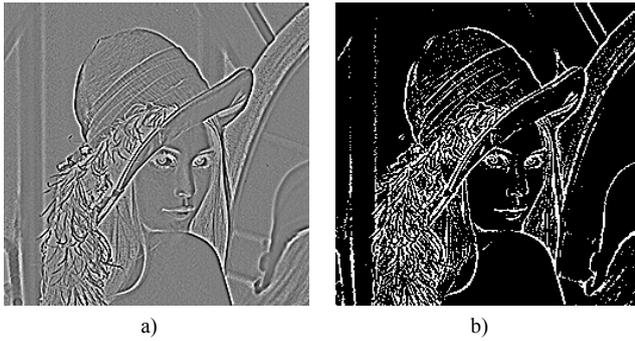
$$\text{LoG} = \nabla^2 G(x,y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (18)$$

Elements  $h(x,y)$  of the convolutional mask can be calculated

$$h(x,y) = c \left( \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (19)$$

where  $c$  is a normalization constant. In practice, the approximation of the LoG by this convolutional mask is often used

$$h = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (20)$$



**Fig. 8.** Response of  $5 \times 5$  LoG convolutional mask (a) and corresponding edge map (b).

Response of LoG is not processed by thresholding but it is inspected whether the sign has changed within the mask of  $2 \times 2$  pixels around the examined point. Figure 8a shows the response of the above-mentioned  $5 \times 5$  LoG convolutional mask applying for image Lena. The mid brightness value (gray points) represents a zero response, high brightness values correspond to positive response values and a low brightness values to negative response values. The resulting map of the edges (detected zero crossings) is in Fig. 8b.

### 2.5 Other Edge Detectors

So far, many methods for edge detection have been published, and not all can be described here in details. However, some methods should be described briefly, at least.

The image often contains complex information in the sense that it contains small objects visible only in close-up view and, on the other hand, large objects that are evident when viewed from a greater distance and whose contours do not have such sharp edges (compared to small details). Therefore, some authors apply multiscale approach to edge detection. A few multiscale methods are based on morphological edge detectors. One method [11] applies the same morphological edge detector to compute edge strength  $G_d^n(k,l)$  at various scales. The resulting edge strength  $G_d(k,l)$  is calculated by

$$G_d(k,l) = \sum_{n=N_1}^{N_2} w_n G_d^n(k,l) \quad (21)$$

where  $N_1, N_2$  is the range of used scales and  $w_n$  are the weighing coefficients chosen by the user. This edge detector was slightly improved by removing the short edges in individual scales before the final summation [24]. Other multiscale detection method [25] combines Canny's detector and scale multiplication.

A truly comprehensive tool for multiscale edge detection is wavelet transformation [26], [27]. Mallat [26] has shown that a multiscale Canny edge detection is equivalent to finding the local maxima of a wavelet transform with smoothing function equal to Gaussian function. Multiscale approach combines local maxima of a wavelet transform

found at various scales to get the resulting edge map [28] or multiplies the wavelet coefficients at two adjacent scales [29].

Edge detectors for color images form a specific category and can be divided into three groups: methods with outputs fusion, methods based on color vector and multi-dimensional gradient detector. Methods based on output fusion apply a suitable edge detector to each color component separately, creating three component edge maps. These maps are then combined into the resulting edge map, most often using logical sum [30], [31]. Multidimensional gradient methods calculate gradients for individual color components, combining them into the resulting function representing edge strength. Koschan [32] uses Sobel operator to compute gradients for each color component. Then for each pixel mean value of color gradients is computed and thresholded to get final edge map. The main problem of the above-mentioned two groups of edge detectors for color images is how to combine edge maps or gradients of color components. Vector methods do not have this problem and appear to be optimal for detecting edges in color images. The most commonly used vector methods [33], [34] are based on vector order statistics (VOS).

### 3. Edge Detection Performance

To compare the various edge detection techniques, it is necessary to design some quantitative criteria for the edge detection performance. Peli and Malah [35] have proposed a number of quantitative measures to study the performance of the edge detectors: percentage of edge points detected on the ideal (desired) edge, number of edge points that do not coincide with the ideal edge (normalized by the number of points on the edge), noise-to-signal ratio and mean width of a detected edge. If all the points of the detected edges create a set  $W_d$  with a number of points  $I_d$ , and the actual edge points create a set  $W_i$  with a number of points  $I_i$ , then the correctly detected edge points form a set  $(W_d \cap W_i)$  with a number of points  $I_c$  and incorrectly detected edge points form a set  $(W_d - W_i)$  with a number of points  $I_e = I_d - I_c$ . Percentage  $C_A$  of edge points detected on the ideal (desired) edge is defined

$$C_A = \frac{I_c}{I_i} 100\% . \quad (22)$$

Percentage  $C_E$  of edge points that do not coincide with the ideal edge (normalized by the number of points on the edge) is computed by

$$C_E = \frac{I_e}{I_i} 100\% . \quad (23)$$

Noise-to-signal ratio  $C_R$  is

$$C_R = 10 \log \left( \frac{I_c}{I_e} \right) [\text{dB}] . \quad (24)$$

Mean width of a detected edge  $C_w$  is

$$C_w = \frac{I_d}{I_i} \tag{25}$$

The above parameters do not include the shift of the detected edge with respect the actual one. So Pratt has introduced a figure of merit  $FoM$  [9] that balances above mentioned three measures (excluding noise-to-signal ratio)

$$FoM = \frac{1}{\max \{I_d, I_i\}} \sum_{k=1}^{I_d} \frac{1}{1 + \delta e^2(k)} \tag{26}$$

where  $\delta$  is a scale constant (usually 1/9) and  $e(k)$  is the pixel-miss distance of the  $k$ -th edge point. It is clear that  $0 \leq FoM \leq 1$ . The more the detected edges match the real edges, the closer the value of  $FoM$  is to 1. If  $FoM = 1$ , then two compared binary images (edge maps) are the same. Binary image with real edges is often referred to as a ground truth.

### 4. Berkeley Segmentation Data Set

In order to correctly evaluate the edge detector performance, it is necessary to have real edges for comparison. It is not a problem in the case of artificially created images. However, edge detection is mainly used to process natural images. In 2001, Martin et al. have introduced The Berkeley Segmentation Database (BSD) [36], a large dataset of natural images that have been manually segmented. The human annotations serve as a ground truth for comparing different segmentation and boundary detection algorithms. A few years later, Martin et al. presented BSDS300, the collection of 300 pictures with their manual segmentations [37]. The images are divided into a training set of 200 images, and a test set of 100 images. For the researchers proposing new methods, the authors of BSDS300 have prepared a guideline. All the learning, parameter tuning, etc. should be done exclusively on the train set. After training, the proposed algorithm should be run only once with fixed parameters on the set of testing images. The pictures and ground-truths of the test set cannot be used for tuning parameters of the proposed algorithm.

The human segmented images provide ground truth boundaries. Any boundary marked by a human subject can be considered to be valid. Since each image has multiple segmentations by different subjects, it is the collection of these human-marked boundaries that constitutes the ground truth, a soft boundary map with one-pixel wide boundaries, valued from zero to one where higher values indicate greater match with a boundary. Figure 9 and Figure 10 show two images from BSDS300, their manual segmentations and ground truths. Normally, a ground truth is a binary image with real edges. In the case of BSDS, a ground truth is a grey scale image where high values of luminance signify greater confidence in the existence of a boundary.

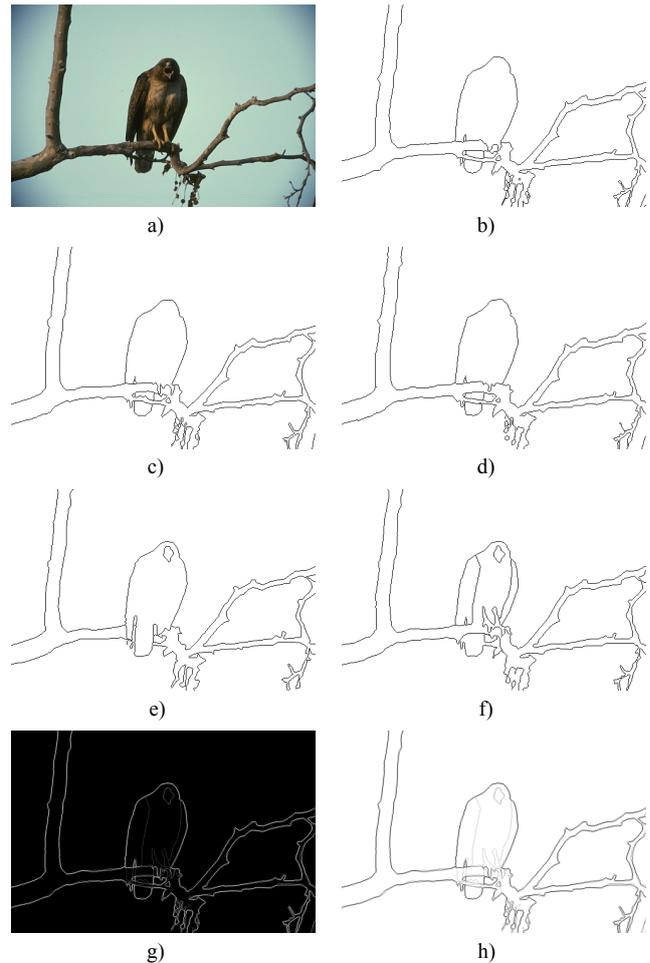
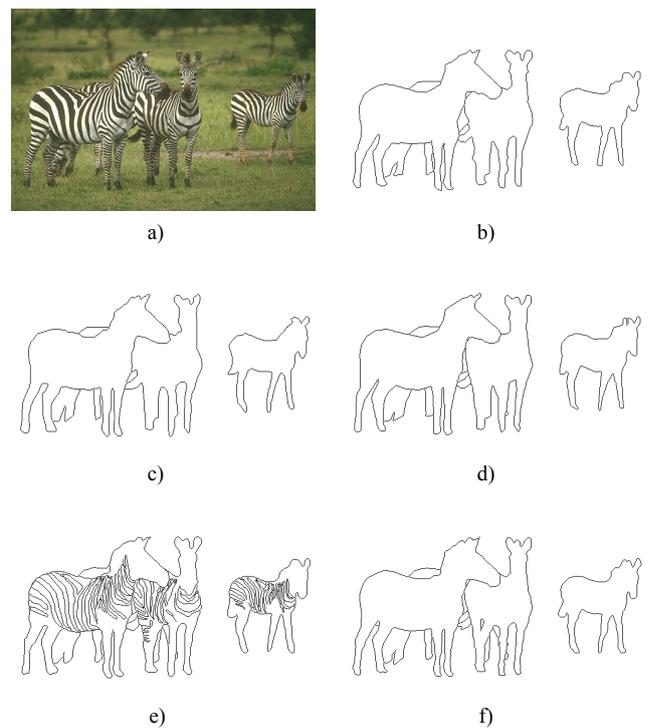


Fig. 9. Original image 42049.jpg (a) from BSDS300, its manual segmentations (b-f), ground truth (g) and negative of ground truth (h).



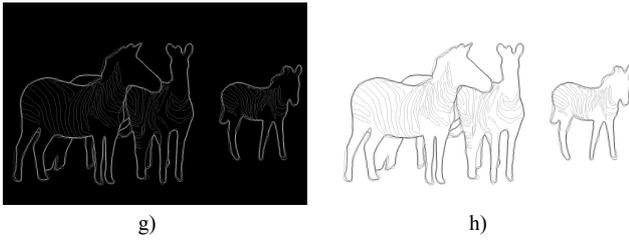


Fig. 10. Original image 253027.jpg (a) from BSDS300, its manual segmentations (b-f), ground truth (g) and negative of ground truth (h).

A new database BSDS500 is currently available. It is an extension of the BSDS300, where the original 300 pictures are used for training or validation and 200 new pictures, with manual segmentations, are added for testing.

Image segmentation is the process of partitioning a digital image into multiple segments and it is typically used to identify objects and boundaries in pictures. Edge detectors look for edge points (edgels), the pixels of a digital image at which the intensity changes rapidly. An edge is the boundary between two regions of different constant intensity. It means that image segmentation and edge detection are not the identical processes. This should be taken into account when evaluating the edge detector performance based on segmentations from BSDS500, especially, when the results of segmentation from different people vary considerably (Fig. 10e).

### 5. Edge Detection in Matlab

Matlab software is a very useful tool for image processing. There is a special Matlab function *edge* which can be directly used for edge detection. The syntaxes for this function are [14]:

```

BW = edge(I) ,
BW = edge(I,method) ,
BW = edge(I,method,) ,
BW = edge(I,method,thr,dir) ,
BW = edge(___,'nothinning') ,
BW = edge(I,method,thr,sigma) ,
BW = edge(I,method,thr,h) ,
[BW,thrOut] = edge(___) ,
[BW,thrOut,Gv,Gh] = edge(___)
    
```

where *BW* = *edge*(*I*) returns a binary image *BW* (edge map) of the input image *I*. By default, the function uses the Sobel edge detection method. *BW* = *edge*(*I,method*) detects edges in image *I* using the specified edge detection algorithm: 'Sobel', 'Prewitt', 'Roberts', 'Canny', 'approx\_canny', 'zero-cross' and 'log'.

For gradient edge detectors Matlab function *edge* uses thresholding to get the final edge map. Users can set threshold values manually with function syntax *BW* = *edge*(*I,method,thr*). If threshold parameter *thr* is missing, function calculates its value automatically. To get this

calculated threshold value, users can apply syntax [*BW,thrOut*] = *edge*(\_\_\_). For Sobel, Prewitt and Roberts methods it is possible to get the directional gradient magnitudes by syntax [*BW,thrOut,Gv,Gh*] = *edge*(\_\_\_). For the Sobel and Prewitt operators, *Gv* and *Gh* correspond to the vertical and horizontal gradients, for the Roberts operator, *Gv* and *Gh* are the gradients at angles of 45° and 135° from horizontal direction.

The Sobel and Prewitt detector can identify edges in the horizontal direction or vertical direction (or both). The Roberts detector can find edges at angles of 45° from horizontal, 135° from horizontal, or both. Users can apply syntax *BW* = *edge*(*I,method,thr,dir*) to specify the orientation *dir* of detected edges. For Sobel, Prewitt and Roberts detector it is possible to skip the step of edge thinning by syntax *BW* = *edge*(\_\_\_,'nothinning'). The syntax *BW* = *edge*(*I,method,thr,sigma*) specifies the standard deviation *sigma* of the filter used in LoG or Canny edge detectors.

Many authors of new methods for edge detection compare edges produced by their detectors with edge maps of older methods (Sobel, Prewitt, etc.) computed by Matlab function *edge* [38–43]. It seems that they have set just detection method and other parameters have been default. But there are two problems with Matlab function *edge*. It is not clear what a method is used for the threshold calculation and what an algorithm is used for thinning. And there is no guarantee that the automatically calculated threshold value results in optimal edge map for the selected method (maximum values of *FoM*). Table 1 summarizes *FoM* values computed from edge maps obtained by Matlab function *edge*(*I,'Prewitt',thr*) applied on picture 42049.jpg. The used threshold parameters were 1.1, 1.05, 1, 0.95 and 0.9 times the threshold value calculated automatically. *FoM* was computed for each threshold and for all five segmentations of picture 42049.jpg (Fig. 9b – Fig. 9f). The results obtained for the picture 253027.jpg are in Tab. 2. From Tab. 1 and Tab. 2 it is clear that automatically calculated threshold does not guarantee optimal results.

Threshold	Segmentation number				
	1	2	3	4	5
0.1194	0.7870	0.8183	0.7676	0.7633	0.8272
0.1140	0.7701	0.8000	0.7516	0.7466	0.8111
0.1085*	0.7546	0.7837	0.7362	0.7321	0.7966
0.1031	0.7389	0.7673	0.7206	0.7155	0.7781
0.0977	0.7147	0.7422	0.6966	0.6926	0.7536

Tab. 1. *FoM* computed for 42049.jpg for different threshold values. Automatically calculated threshold is marked with \*.

Threshold	Segmentation number				
	1	2	3	4	5
0.1953	0.1925	0.1892	0.1905	0.6817	0.2109
0.1864	0.1958	0.1928	0.1937	0.6978	0.2145
0.1775*	0.1965	0.1925	0.1944	0.7138	0.2151
0.1686	0.2029	0.1980	0.2003	0.7231	0.2205
0.1598	0.2076	0.2015	0.2034	0.7074	0.2237

Tab. 2. *FoM* computed for 253027.jpg for different threshold values. Automatically calculated threshold is marked with \*.

## 6. New Trends in Edge Detection

Over the last ten years, new approaches to edge detection have been presented, for example dictionary learning [44], [45] or fuzzy logic [46], [47]. The most promising methods are based on fractional derivatives [3, 39, 43], [48–50]. Generalized form of the Grunwald-Letnikov fractional derivative  $d^\alpha f(x)$  of order  $\alpha$  is [51]

$$d^\alpha f(x) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{m=0}^{u-l} (-1)^m \frac{\Gamma(\alpha+1)}{m! \Gamma(\alpha-m+1)} f(x-mh) \quad (27)$$

where  $m$  is an integer number and  $u$  and  $l$  are the integer upper and lower limits of differentiation,  $l < \alpha \leq u$ .  $\Gamma(z)$  is the gamma function given by [51]

$$\Gamma(z) = \int_0^\infty e^{-u} u^{z-1} du, \text{ for all } z \in \mathbb{R}. \quad (28)$$

To get fractional derivative mask for edge detection the fractional derivatives  $\nabla_x^\alpha f(x,y)$  and  $\nabla_y^\alpha f(x,y)$  along  $x$  and  $y$  axes can be simplified as follows [39]

$$\nabla_x^\alpha f(x,y) = \sum_{k=0}^{K-1} (-1)^k \binom{\alpha}{k} f(x-k,y), \quad (29)$$

$$\nabla_y^\alpha f(x,y) = \sum_{k=0}^{K-1} (-1)^k \binom{\alpha}{k} f(x,y-k) \quad (30)$$

where

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \quad (31)$$

and  $K \geq 3$  is an integer representing the size of fractional derivative mask. The magnitude of the fractional gradients  $\nabla^\alpha f(x,y)$  is given by

$$\|\nabla^\alpha f(x,y)\| = \sqrt{(\nabla_x^\alpha f(x,y))^2 + (\nabla_y^\alpha f(x,y))^2}. \quad (32)$$

The  $y$ -directional fractional mask of three rows can be computed by [39]

$$\begin{bmatrix} 0.5(-1)^{K-1} \binom{\alpha}{K-1} & \dots & \frac{\alpha^2 - \alpha}{4} & -\frac{\alpha}{2} & 0.5 \\ (-1)^{K-1} \binom{\alpha}{K-1} & \dots & \frac{\alpha^2 - \alpha}{2} & -\alpha & 1 \\ 0.5(-1)^{K-1} \binom{\alpha}{K-1} & \dots & \frac{\alpha^2 - \alpha}{4} & -\frac{\alpha}{2} & 0.5 \end{bmatrix}, \quad (33)$$

and the  $x$ -directional fractional mask of three columns can be computed by [39]

$$\begin{bmatrix} (-1)^{K-1} \binom{\alpha}{K-1} & (-1)^{K-1} \binom{\alpha}{K-1} & (-1)^{K-1} \binom{\alpha}{K-1} \\ \dots & \dots & \dots \\ \frac{\alpha^2 - \alpha}{4} & \frac{\alpha^2 - \alpha}{2} & \frac{\alpha^2 - \alpha}{4} \\ -\frac{\alpha}{2} & -\alpha & -\frac{\alpha}{2} \\ 0.5 & 1 & 0.5 \end{bmatrix}. \quad (34)$$

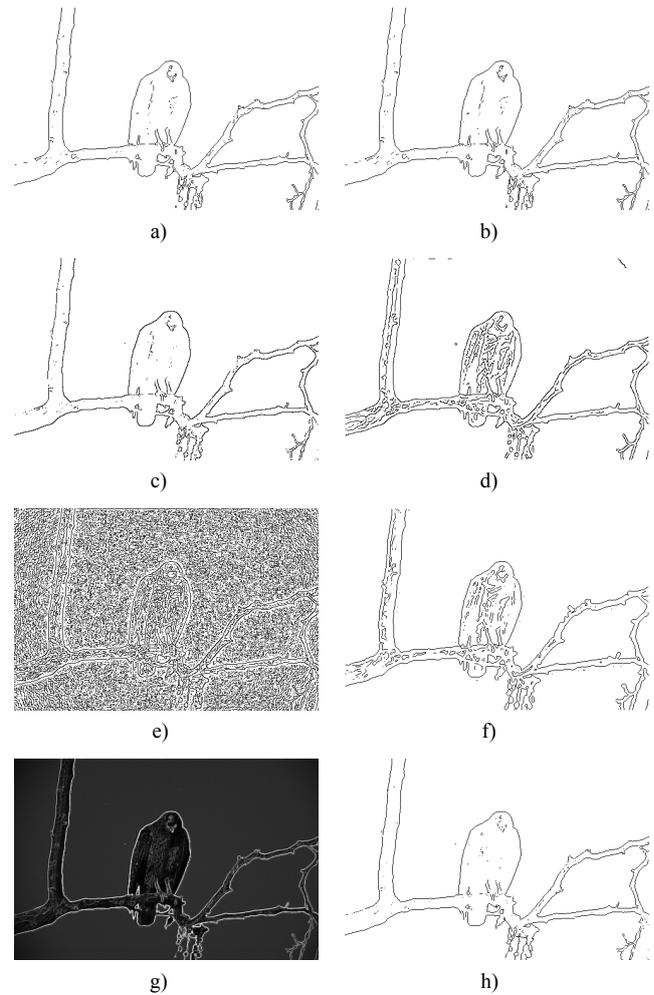
Experiments have shown that the appropriate values of parameters  $\alpha$  and  $K$  for edge detection are  $\alpha = 1.2$  and  $K = 3$  [39]. Then the particular directional fractional masks for edge detection are

$$\begin{bmatrix} 0.06 & -0.6 & 0.5 \\ 0.12 & -1.2 & 1 \\ 0.06 & -0.6 & 0.5 \end{bmatrix} \quad (35)$$

and

$$\begin{bmatrix} 0.06 & 0.12 & 0.06 \\ -0.6 & -1.2 & -0.6 \\ 0.5 & 1 & 0.5 \end{bmatrix}. \quad (36)$$

Similarly to classical gradient edge detectors, even for edge detection based on fractional derivation, it is necessary to use thresholding and thinning to get edges with a one-pixel width. Figure 11 shows the magnitude of fractional derivation (Fig. 11g) and corresponding edges after thresholding and thinning (Fig. 11h). Magnitude image was thresholded by the value of 0.4157, which was calculated by



**Fig. 11.** Edge detected in image 42049.jpg by Prewitt (a), Sobel (b), Roberts (c), Canny (d), approxcanny (e) and LoG (f) edge detectors. Magnitude of fractional derivation (g) and corresponding edges after thresholding (h).

Method	Segmentation number				
	1	2	3	4	5
Prewitt	0.7546	0.7837	0.7362	0.7321	0.7966
Sobel	0.7552	0.7842	0.7362	0.7334	0.7967
Roberts	0.7096	0.7227	0.6877	0.6831	0.7321
Canny	0.4080	0.4221	0.3972	0.3940	0.4388
Ap.can.	0.0600	0.0610	0.0580	0.0579	0.0639
LoG	0.5188	0.5375	0.5059	0.5020	0.5550
Fract.	0.8968	0.9109	0.8673	0.8654	0.8739

Tab. 3. *FoM* computed for 42049.jpg by different edge detectors.

Otsu's method [17]. The Matlab function *bwmorph(BW, 'thin', Inf)* was used for thinning. For comparison, in Fig. 11 there are also edges obtained by other gradient edge detectors. Matlab function *edge* with default setting was used for calculations (Fig. 11a-f). Table 3 summarizes *FoM* values computed from edge maps in Fig. 11 with respect to segmentations of picture 42049.jpg.

The same approach was used to process 253027.jpg. Obtained edge maps are in Fig. 12 and corresponding *FoM* values with respect to segmentations of picture 253027.jpg are in Table 4. Magnitude image was thresholded by the

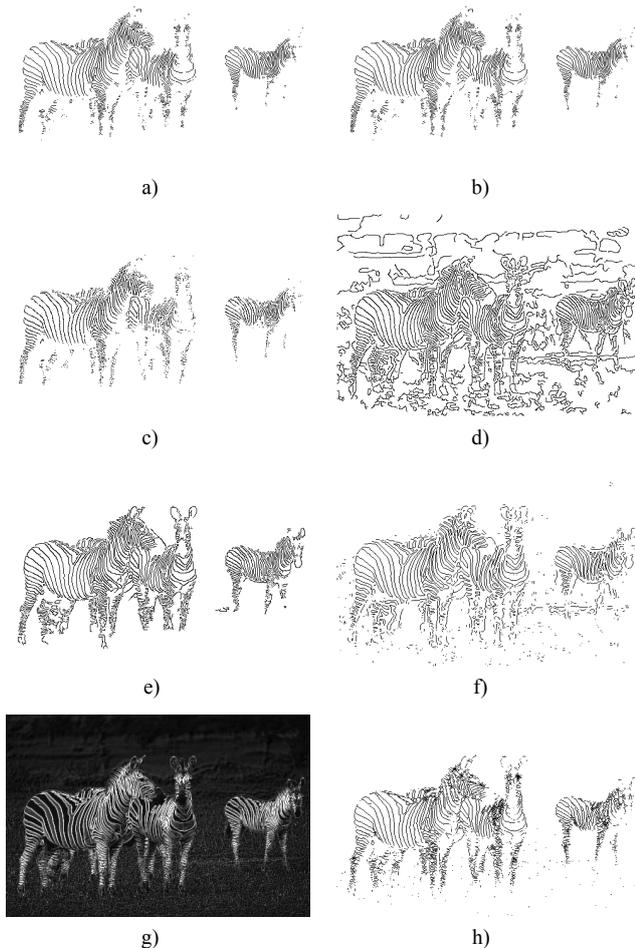


Fig. 12. Edge detected in image 253027.jpg by Prewitt (a), Sobel (b), Roberts (c), Canny (d), approxcanny (e) and LoG (f) edge detectors. Magnitude of fractional derivation (g) and edges after thresholding (h).

Method	Segmentation number				
	1	2	3	4	5
Prewitt	0.1965	0.1925	0.1944	0.7138	0.2151
Sobel	0.1981	0.1938	0.1956	0.7220	0.2164
Roberts	0.2133	0.2085	0.2107	0.7000	0.2330
Canny	0.1222	0.1182	0.1174	0.3154	0.1285
Ap.can.	0.1795	0.1714	0.1719	0.5055	0.1896
LoG	0.2355	0.2264	0.2267	0.6359	0.2461
Fract.	0.2237	0.2165	0.2176	0.6467	0.2382

Tab. 4. *FoM* computed for 253027.jpg by different edge detectors.

value of 0.4, which was calculated by Otsu's method [17]. From the result presented in Tab. 3 and Tab. 4 it is clear that Prewitt and Sobel edge detectors produce almost identical edges. It can be also concluded that edge detection based on fractional derivatives produces comparable or better results than older gradient methods, such as Prewitt, Sobel, etc.

## 7. Conclusion

The paper has four main thematic parts. In the first part, a brief description of the frequently applied edge detection methods is presented. The second part of the article relates to the image databases BSDS300 and BSDS500. They represent very useful tool for edge detectors performance quantification. BSDS500 is a large dataset of natural images that have been manually segmented by different subjects. Two pictures with different characteristics have been chosen to illustrate results of manual segmentation. Picture 42049.jpg contains one simple object (eagle) with clear background. All segmentations of this picture are similar, as indicated by almost the same values of *FoM* (Tab. 1, Tab. 3). The picture 253027.jpg is not so easy task for manual segmentation. There are three subjects with texture, three zebras with white and black strips. One segmentation (Fig. 10e) differs significantly from others, because one subject understands segmentation differently from the others. Interestingly, this distinct segmentation gives the highest values of *FoM* (Tab. 2, Tab. 4). This is a proof that image segmentation and edge detection are not the identical processes. This should be taken into account when evaluating the edge detector performance based on segmentations from BSDS500.

Matlab function for edge detection is often applied as part of a larger process or it is used by researchers to calculate edge maps by older algorithms for comparison with their new proposed methods. Almost always this function is used with default settings, though there are some function parameters which can affect the results significantly. Table 1 and Table 2 show that the threshold value calculated automatically does not guarantee optimal edge maps.

Over the last ten years, new approaches to edge detection have been presented. The most promising methods are based on fractional derivatives. In the paper edge detector based on simple directional fractional derivative masks is presented. Magnitude image obtained by these

masks was thresholded and thinned to get edges with a one-pixel width. The results presented in Tab. 3 and Tab. 4 show that edge detection based on fractional derivatives produces comparable or better results than older gradient methods, such as Prewitt, Sobel, etc.

## Acknowledgments

This work was supported by the Ministry of Education of the Slovak Republic under the grant VEGA No. 1/0668/17.

## References

- [1] QIU, T., YAN, Y., LU, G. An autoadaptive edge-detection algorithm for flame and fire image processing. *IEEE Transactions on Instrumentation and Measurement*, 2012, vol. 61, no. 5, p. 1486–1493. DOI: 10.1109/TIM.2011.2175833
- [2] AL-GHAILI, A. M., MASHOHOR, S., RAMLI, A. R., ISMAIL, A. Vertical-edge-based car-license-plate detection method. *IEEE Transactions on Vehicular Technology*, 2013, vol. 62, no. 1, p. 26–38. DOI: 10.1109/TVT.2012.2222454
- [3] CHEN, J., HUANG, C., DU, Y., LIN, C. Combining fractional-order edge detection and chaos synchronisation classifier for fingerprint identification. *IET Image Processing*, 2014, vol. 8, no. 6, p. 354–362. DOI: 10.1049/IET-IPR.2012.0660
- [4] BASELICE, F., FERRAIOLI, G., REALE, D. Edge detection using real and imaginary decomposition of SAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 2014, vol. 52, no. 7, p. 3833–3842. DOI: 10.1109/TGRS.2013.2276917
- [5] JIN, R., YIN, J., ZHOU, W., YANG, J. Improved multiscale edge detection method for polarimetric SAR images. *IEEE Geoscience and Remote Sensing Letters*, 2016, vol. 13, no. 8, p. 1104–1108. DOI: 10.1109/LGRS.2016.2569534
- [6] LIU, C., XIAO, Y., YANG, J. A coastline detection method in polarimetric SAR images mixing the region-based and edge-based active contour models. *IEEE Transactions on Geoscience and Remote Sensing*, 2017, vol. 55, no. 7, p. 3735–3747. DOI: 10.1109/TGRS.2017.2679112
- [7] SAHEBA, S. M., UPADHYAYA, T. K., SHARMA, R. K. Lunar surface crater topology generation using adaptive edge detection algorithm. *IET Image Processing*, 2016, vol. 10, no. 9, p. 657–661. DOI: 10.1049/iet-ipr.2015.0232
- [8] WANG, Y., TAVANAPONG, W., WONG, J., et al. Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy. *IEEE Journal of Biomedical and Health Informatics*, 2014, vol. 18, no. 4, p. 1379–1389. DOI: 10.1109/JBHI.2013.2285230
- [9] PRATT, W. K. *Digital Image Processing*. 4<sup>th</sup> ed., Hoboken, New Jersey (USA): John Wiley & Sons, 2007. ISBN: 9780471767770
- [10] CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, vol. 8, no. 6, p. 679–698. DOI: 10.1109/TPAMI.1986.4767851
- [11] CHANDA, B., KUNDU, M. K., PADMAJA, Y. V. A multi-scale morphologic edge detector. *Pattern Recognition*, 1998, vol. 31, no. 10, p. 1469–1478. DOI: 10.1016/S0031-3203(98)00014-4
- [12] LEE, J. S., HARALICK, R. M., SHAPIRO, L. G. Morphologic edge detection. *IEEE Journal of Robotics and Automation*, 1987, vol. 3, no. 2, p. 142–156. DOI: 10.1109/JRA.1987.1087088
- [13] SHUI, P., WANG, F. Anti-impulse-noise edge detection via anisotropic morphological directional derivatives. *IEEE Transactions on Image Processing*, 2017, vol. 26, no. 10, p. 4962–4977. DOI: 10.1109/TIP.2017.2726190
- [14] GONZALEZ, R. C., WOODS, R. E., EDDINS, S. *Digital Image Processing Using MATLAB*. 2<sup>nd</sup> ed. Gatesmark Publishing (USA), 2009. ISBN: 978-0982085400
- [15] ROSIN, P. L. Unimodal thresholding. *Pattern Recognition*, 2001, vol. 34, no. 11, p. 2083–2096. DOI: 10.1016/S0031-3203(00)00136-9
- [16] BRINK, A. D., PENDOCK, N. E. Minimum cross-entropy threshold selection. *Pattern Recognition*, 1966, vol. 29, no. 1, p. 179–188. DOI: 10.1016/0031-3203(95)00066-6
- [17] OTSU, N. A threshold selection method from gray-level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979, vol. 9, no. 1, p. 62–66. DOI: 10.1109/TSMC.1979.4310076
- [18] SEN, D., PAL, S. K. Thresholding for edge detection in SAR images. In *Proceedings of IEEE-International Conference on Signal Processing, Communications and Networking*. Chennai (India), 2008, p. 311–316. DOI: 10.1109/ICSCN.2008.4447210
- [19] CHOW, C. K., KANEKO, T. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, 1972, vol. 5, no. 4, p. 388–410. DOI: 10.1016/0010-4809(72)90070-5
- [20] NAIN, N., JINDAL, G., GARG, A., JAIN, A. Dynamic thresholding based edge detection. In *Proceedings of the World Congress on Engineering WCE 2008*. London (U.K.), 2008, p. 694–699. ISBN: 978-988-98671-9-5
- [21] CANNY, J. Finding edges and lines in images. *Report AI-TR-720*, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1983.
- [22] LACROIX, V. A three-module strategy for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988, vol. 10, no. 6, p. 803–810. DOI: 10.1109/34.9103
- [23] MARR, D., HILDRETH, E. Theory of edge detection. *Proceedings of the Royal Society of London, Series B, Biological Sciences*, 1980, vol. 207, no. 1167, p. 187–217. DOI: 10.1098/rspb.1980.0020
- [24] LI, Z., YANG, Y., JIANG, W. Multi-scale morphologic tracking approach for edge detection. In *Proceedings of the Fourth International Conference on Image and Graphics*. Sichuan (China), 2007, p. 358–362. DOI: 10.1109/ICIG.2007.86
- [25] BAO, P., ZHANG, L., WU, X. Canny edge detection enhancement by scale multiplication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27, no. 9, p. 1485–1490. DOI: 10.1109/TPAMI.2005.173
- [26] MALLAT, S., ZHONG, S. Characterization of signals from multi-scale edges. *Transactions on Pattern Analysis and Machine Intelligence*, 1992, vol. 14, no. 7, p. 710–732. DOI: 10.1109/34.142909
- [27] MALLAT, S., HWANG, W. L. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 1992, vol. 32, no. 2, p. 617–643. DOI: 10.1109/18.119727
- [28] HASSANZADEH, M. R., ARDESHIR, G. R., KARAMI MOLLAEI, M. R. Two new methods for finding endocardial and epicardial boundaries in echocardiographic images using wavelet analysis. *European Journal of Scientific Research*, 2009, vol. 27, no. 2, p. 264–274. ISSN: 1450-216X
- [29] ZHANG, L., BAO, P. Edge detection by scale multiplication in wavelet domain. *Pattern Recognition Letters*, 2002, vol. 23, no. 14, p. 1771–1784. DOI: 10.1016/S0167-8655(02)00151-4
- [30] HEDDLEY, M., YAN, H. Segmentation of color images using spatial and color space information. *Journal of Electronic Imaging*, 1992, vol. 1, no. 4, p. 374–380. DOI: 10.1117/12.61158

- [31] FAN, J., YAU, D. K. Y., ELMAGARMID, A. K., AREF, W. G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 2001, vol. 10, no. 10, p. 1454–1466. DOI: 10.1109/83.951532
- [32] KOSCHAN, A. A comparative study on color edge detection. In *Proceedings of the 2nd Asian Conference on Computer Vision ACCV'95*, Singapore, 1995, p. 574–578.
- [33] TRAHANIAS, P. E., VENETSANOPOULOS, A. N. Color edge detection using vector order statistics. *IEEE Transactions on Image Processing*, 1993, vol. 2, no. 2, p. 259–264. DOI: 10.1109/83.217230
- [34] TRAHANIAS, P. E., VENETSANOPOULOS, A. N. Vector order statistics operators as color edge detectors. *IEEE Transactions on Systems, Man, and Cybernetics*, 1996, vol. 26, no. 1, p. 135–143. DOI: 10.1109/3477.484445
- [35] PELI, T., MALAH, D. A study of edge detection algorithms. *Computer Graphics Image Processing*, 1982, vol. 20, no. 1, p. 1–21. DOI: 10.1016/0146-664X(82)90070-3
- [36] MARTIN, D., FOWLKES, C., TAL, D., MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE international conference on computer vision (ICCV 2001)*. Vancouver (Canada), 2001, p. 416–425. DOI: 10.1109/ICCV.2001.937655
- [37] ARBELAEZ, P., MAIRE, M., FOWLKES, C., MARTIN, D. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, vol. 33, no. 5, p. 898–916. DOI: 10.1109/TPAMI.2010.161
- [38] ARORA, S., HANMANDLU, M., GUPTA, G. Edge detection of digital color images using information sets. *Journal of Electronic Imaging*, 2016, vol. 25, no. 6, id. no. 061607. DOI: 10.1117/1.JEI.25.6.061607
- [39] VIGNESH, K., ARORA, S., HANMANDLU, M. Edge detection using fractional derivatives and information sets. *Journal of Electronic Imaging*, 2018, vol. 27, no. 5, id. 051226. DOI: 10.1117/1.JEI.27.5.051226
- [40] YI, S., LABATE, D., EASLEY, G. R., KRIM, H. A shearlet approach to edge analysis and detection. *IEEE Transactions on Image Processing*, 2009, vol. 18, no. 5, p. 929–941. DOI: 10.1109/TIP.2009.2013082
- [41] JIANG, W., LAM, K., SHEN, T. Efficient edge detection using simplified Gabor wavelets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009, vol. 39, no. 4, p. 1036–1047. DOI: 10.1109/TSMCB.2008.2011646
- [42] AGAIAN, S. S., PANETTA, K. A., NERCESSIAN, S. C., DANAHY, E. E. Boolean derivatives with application to edge detection for imaging systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2010, vol. 40, no. 2, p. 371–382. DOI: 10.1109/TSMCB.2009.2024771
- [43] GAO, C. B., ZHOU, J. L., HU, J. R., LANG, F. N. Edge detection of colour image based on quaternion fractional differential. *IET Image Processing*, 2011, vol. 5, no. 3, p. 261–272. DOI: 10.1049/iet-ipr.2009.0409
- [44] MAIRAL, J., LEORDEANU, M., BACH, F., et al. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proceedings of European Conference on Computer Vision (ECCV)*. Marseille (France), 2008, p. 43–56. DOI: 10.1007/978-3-540-88690-7\_4
- [45] WANG, Q., ZHANG, G. Ore image edge detection using HOG-index dictionary learning approach. *The Journal of Engineering*, 2017, vol. 2017, no. 9, p. 542–543. DOI: 10.1049/joe.2017.0343
- [46] MELIN, P., GONZALES, C. I., CASTRO, J. R., et al. Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 2014, vol. 22, no. 6, p. 1515–1525. DOI: 10.1109/TFUZZ.2013.2297159
- [47] VERMA, O. P., PARIHAR, A. S. An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm. *IEEE Transactions on Fuzzy Systems*, 2017, vol. 25, no. 1, p. 114–127. DOI: 10.1109/TFUZZ.2016.2551289
- [48] MATHIEU, B., MELCHIOR, P., OUSTALOUPE, A., et al. Fractional differentiation for edge detection. *Signal Processing*, 2003, vol. 83, no. 11, p. 2421–2432. DOI: 10.1016/S0165-1684(03)00194-4
- [49] AMOAKO-YIRENKYI, P., APPATI, J. K., DONTWI, I. K. A new construction of a fractional derivative mask for image edge analysis based on Riemann-Liouville fractional derivative. *Advances in Difference Equations*, 2016, no. 1, p. 1–23. DOI: 10.1186/s13662-016-0946-8
- [50] NANDAL, A., GAMBOA-ROSALES, H., DHAKA, A., et al. Image edge detection using fractional calculus with feature and contrast enhancement. *Circuits, Systems, and Signal Processing*, 2018, vol. 37, no. 9, p. 3946–3972. DOI: 10.1007/s00034-018-0751-6
- [51] LOVERRO, A. Fractional calculus: history, definitions and applications for the engineer. *Technical report*, University of Notre Dame, Notre Dame (USA), 2004, p. 1–28.

## About the Authors ...

**Miroslav HAGARA** (corresponding author) was born in 1961. He received his M.Sc. in 1986 and Ph.D. degree in 2012 from the Slovak Technical University. His research interests cover 3D television, digital image processing, edge detection with sub-pixel accuracy. He is a member of the Editorial Board of the Radioengineering Journal. He was a member of Technical Program Committee of several international conferences.

**Peter KUBINEC** was born in 1966. He graduated from the Slovak University of Technology in Bratislava with Ph.D. degree in 2010. His Ph.D. work was focused on reverberator algorithms in digital audio signal processing. His research interests cover audio-video technique and audio signal analysis. He gives a lecture on microcontroller and digital signal processor programming techniques.