

## Analysis of a robust edge detection system in different color spaces using color and depth images

*S.M.H. Mousavi<sup>1</sup>, V. Lyashenko<sup>2</sup>, V.B.S. Prasath<sup>3</sup>*

<sup>1</sup>*Department of Computer Engineering, Faculty of Engineering, Bu Ali Sina University, Hamadan, Iran;*

<sup>2</sup>*Department of Informatics (INF), Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;*

<sup>3</sup>*Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati OH 45229 USA*

### *Abstract*

Edge detection is very important technique to reveal significant areas in the digital image, which could aids the feature extraction techniques. In fact it is possible to remove un-necessary parts from image, using edge detection. A lot of edge detection techniques has been made already, but we propose a robust evolutionary based system to extract the vital parts of the image. System is based on a lot of pre and post-processing techniques such as filters and morphological operations, and applying modified Ant Colony Optimization edge detection method to the image. The main goal is to test the system on different color spaces, and calculate the system's performance. Another novel aspect of the research is using depth images along with color ones, which depth data is acquired by Kinect V.2 in validation part, to understand edge detection concept better in depth data. System is going to be tested with 10 benchmark test images for color and 5 images for depth format, and validate using 7 Image Quality Assessment factors such as Peak Signal-to-Noise Ratio, Mean Squared Error, Structural Similarity and more (mostly related to edges) for prove, in different color spaces and compared with other famous edge detection methods in same condition. Also for evaluating the robustness of the system, some types of noises such as Gaussian, Salt and pepper, Poisson and Speckle are added to images, to shows proposed system power in any condition. The goal is reaching to best edges possible and to do this, more computation is needed, which increases run time computation just a bit more. But with today's systems this time is decreased to minimum, which is worth it to make such a system. Acquired results are so promising and satisfactory in compare with other methods available in validation section of the paper..

**Keywords:** Edge detection, ant colony optimization (ACO), color spaces, depth image, kinect V.2, image quality assessment (IQA), image noises.

**Citation:** Mousavi SMH, Lyashenko V, Prasath VBS. Analysis of a robust edge detection system in different color spaces using color and depth images. Computer Optics 2019; 43(4): 632-646. DOI: 10.18287/2412-6179-2019-43-4-632-646.

**Acknowledgments:** The authors received no financial support for the research and/or authorship of this article.

### **Introduction**

Determining vital parts of the digital image is a challenging area these days (especially in depth images). A lot of feature extraction methods are made and a lot of modification is applied on them so far. But extracting features in spatial and frequency domain from image, applies on whole image, even less desirable part with low amount of information. Using edge detection techniques [1], it is possible to extract just vital and highly desirable part from an image. So, after determining the vital part of the image, applying feature extraction methods is so logical. For example in extracting facial muscles features' in facial expression recognition problem, we need to just deal with muscles part and not with other parts of the face. Using edge detection technique, it is possible to first determine and extract those muscles and then apply feature extraction algorithms on them. There are some famous traditional edge detection algorithms. Also some modified version of them is made too. Recently some other intelligent edge detection algorithms have appeared, which they use Evolutionary Algorithms (EA) [2] to determine the vital parts of the image. We are going to use the third version, which is evolutionary one, but with some modification and also some pre and post processing technique to make a bit slow but robust edge detection method. Edge detection tech-

nique has application in segmentation [3] feature detection [4] object detection and tracking [5] medicine [6], biometrics [7], industry [8] and more. The paper is divided into 5 sections, which section "introduction", defines the fundamentals needs of the problem and second section pays to some of the related research in the subject area. Section three demonstrate proposed evolutionary edge detection algorithm on different color spaces for color and depth images in details. Evaluations using benchmark test images for color and depth data, and comparing with other edge detection algorithms using Image Quality Assessment (IQA) factors in different color spaces with different noise environment is take placed in section "Validation and Results", and final section, include the conclusion, suggestions and discussion for future works.

### ***Color spaces***

A device color space simply describes the range of colors, or gamut, that a camera can see, a printer can print, or a monitor can display. Editing color spaces, on the other hand, such as Adobe RGB or sRGB, are device-independent. They also determine a color range you can work in. Their design allows you to edit images in a controlled, consistent manner. A color space is a specific organization of colors. In combination with physical device profiling, it allows for reproducible representations of

color, in both analog and digital representations. In order to evaluate the system we are going to use different colour space systems. Used colour spaces are RGB, CIE (Lab), YIQ and YCbCr [9–11]. Each colour space has its characteristics. One of the most famous colour space is RGB space which is widely used by researchers and that is due to its ease of use and understanding. In Figure 1 (a), “monarch” [12–13] benchmark test image is represented in different color space channels. Also a manually recorded depth test image (Teapot) using Kinect sensor is displayed in four color spaces in this figure too. It is mentionable that, in all tests for CIE color space, L component or channel for color data and all components or channels for depth data are used. Color spaces are represented in their actual condition, so they might not be completely visible in printed version and must be analyzed in electronic version. For example, depth data in Figure 1 (a) in YIQ and YCbCr are visible so hard for human eyes, but as this is the first step of pre-processing, could be ignored.

Figure 1 (b), represents monarch test image in different channels of CIE-Lab color space, which as it is clear, all component mode is not subtle for proposed method, which leads us to us just L channel for color data. But all channels are used for depth data. Final edge detected result is nicely visible for human eyes, as Figure 6 represents.

## 1. RGB

RGB (Red, Green and Blue) describes what kind of light needs to be emitted to produce a given color. Light is added together to create form the darkness. RGB stores individual values for red, green and blue. RGB is not a color space, it is a color model. There are many different RGB color spaces derived from this color model. The sRGB color space, or

standard RGB (Red Green Blue), is an RGB color space created cooperatively by Hewlett-Packard and Microsoft Corporation for use on the Internet [9–11].

## 2. CIE 1976 or CIE (Lab)

The CIELAB color space (also known as CIE  $L \times a \times b \times$  or sometimes abbreviated as simply "Lab" color space) is a color space defined by the International Commission on Illumination (CIE) in 1976. It expresses color as three numerical values,  $L \times$  for the lightness and  $a \times$  and  $b \times$  for the green-red and blue-yellow color components. CIELAB was designed to be perceptually uniform with respect to human color vision. [9–11]. Also this color space is device-independent and includes both the gamuts of the RGB and CMYK color models.

## 3. YIQ

YIQ was used in NTSC (North America, Japan and elsewhere) television broadcasts. This system stores a luma value with two chroma or chrominance values, corresponding approximately to the amounts of blue and red in the color. The Y component represents the luma information, and is the only component used by black-and-white television receivers. I and Q represent the chrominance information [9–11].

## 4. YCbCr

YCbCr, Y'CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y' is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y' (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries [9–11].

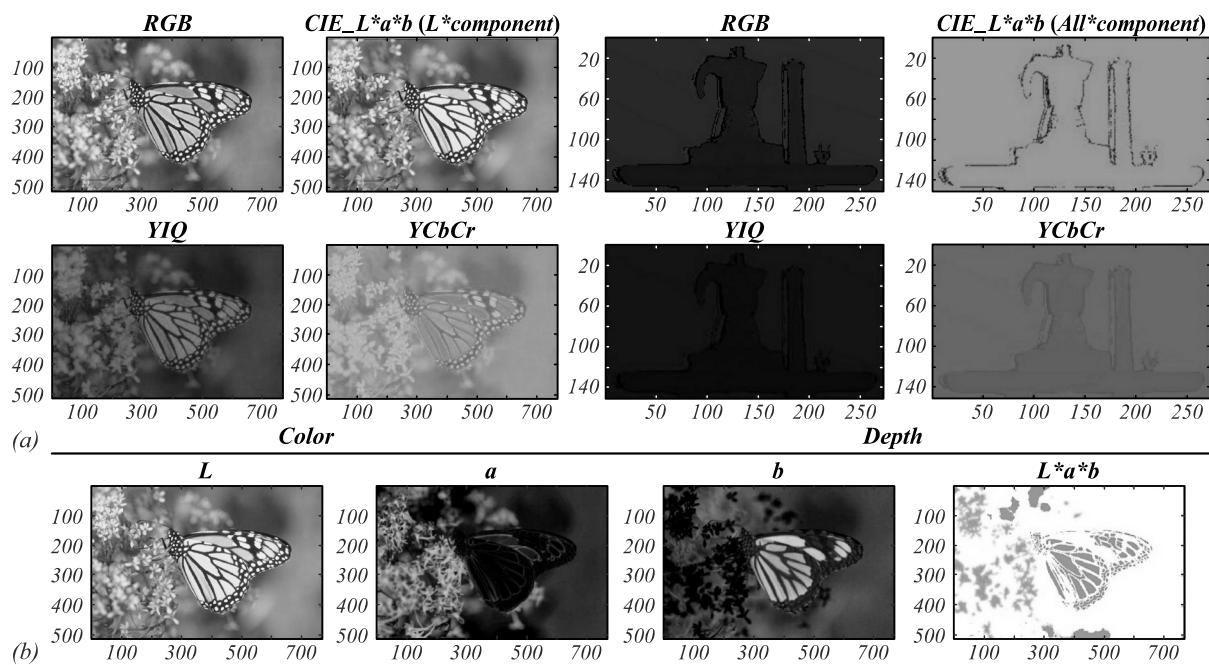


Fig. 1. (a) Monarch color test image in RGB, CIE – Lab ( $L$  component for color data and all components for depth data), YIQ and YCbCr color spaces, and manually recorded depth test image using Kinect V.2 sensor in mentioned color spaces (top).

(b) Monarch test image in different  $L$ ,  $a$ ,  $b$  and all Lab components

### Image noises

Also we are going to add some diverse noises to the test images for evaluating the robustness of the system. We are going to use four famous noises which are Gaussian, Salt & pepper, Poisson and Speckle. Figure 2 represents “cat” benchmark test image, which is polluted with these four noises and different parameters in different color spaces.

#### 1. Gaussian

Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the normal distribution, which is also known as the Gaussian distribution. The probability density function  $P$  of a Gaussian random variable  $Z$  is given by (1):

$$P_g(Z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(Z-\mu)^2}{2\sigma^2}}, \quad (1)$$

where  $Z$  represents the grey level,  $\mu$  the mean value and  $\sigma$  the standard deviation.

Principal sources of Gaussian noise in digital images arise during acquisition sensor noise caused by poor illumination and/or high temperature, and/or transmission e.g. electronic circuit noise. In digital image processing Gaussian noise can be reduced using a spatial filter, and smoothing the image with a low pass filter [14–15].

#### 2. Salt & pepper

Salt-and-pepper noise (2) is a form of noise sometimes seen on images. It is also known as impulse noise. This noise can be caused by sharp and sudden disturbances in the image signal. It presents itself as sparsely occurring white and black pixels. An effective noise reduction method for this type of noise is a median filter or a morphological filter [14–15].

$$P(Z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b, \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

If  $b > a$ , then gray level  $b$  will appear as a light dot in the image. If either  $P_a$  or  $P_b$  is zero, the impulse noise is called unipolar noise. If neither  $P_a$  nor  $P_b$  is zero and if they are approximately equal the impulse noise is called salt & pepper.

#### 3. Poisson

Shot noise or Poisson noise (3) is a type of electronic noise which can be modeled by a Poisson process. In electronics shot noise originates from the discrete nature of electric charge. Shot noise also occurs in photon counting in optical devices, where shot noise is associated with the particle nature of light [16].

$N$  measured by a given sensor element over a time interval  $t$  is described by the discrete probability distribution

$$\Pr(N = k) = \frac{e^{-\lambda t} (\lambda t)^k}{k!}, \quad (3)$$

where  $\lambda$  is the expected number of photons per unit time interval, which is proportional to the incident scene irradiance.

#### 4. Speckle

Speckle noise (4) in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area. The origin of this noise is seen if we model our reflectivity function as an array of scatterers. Because of the finite resolution, at any time we are receiving from a distribution of scatterers within the resolution cell.

$$g(m, n) = f(m, n) \times u(m, n) \times \eta(m, n), \quad (4)$$

where  $g(m, n)$  is corrupted image,  $u(m, n)$  is multiplicative component and  $\eta(m, n)$  is additive component [17].

#### Depth or range image and sensors

Depth or range images are type of digital images which represents the distance between object and the sensor and returns the calculated distance in different metric units depend on sensor's type. Also this happens using different technology but mostly these sensors use infrared spectrum to capture the range. Main factor of these sensors is their vision in pure darkness like night time which is an advantage versus color sensors. Their outputs are mostly in black and white or gray scale and each pixel shows the distance till object in for example millimeter (Kinect here). Depth sensor technology is almost new but not so much and provides more information from object along with color sensor to us. Depth images are called 2.5 dimension images, because it is possible to extract 3D model of the object out of it. There are different depth sensors like Microsoft Kinect [18] Asus Xtion Pro [19] Intel® Real Sense™ Depth Camera [20] Primesense carmine [21] and more. Kinect [18] is one of the Microsoft productions for capturing color and depth images, which we used Kinect version 2 in this research (just for capturing depth data).

#### **Prior related researches**

Most of the classic edge detection operators such: Canny [22], Zerocross [23], Log [24], Roberts [25] Prewitt [26], Sobel [27] are the examples of Gradient-based edge detection methods. These operators, due to being so sensitive to noises, are not suitable for the main stage of image processing. Recently, a variety of different methods on edge detection of noisy images (without the effect of noise on the edge), has been innovated. Some of them are: wavelet transform [28], mathematical morphology method [29], neural networks [30], Fuzzy method [31] and etc. Also from evolutionary type, it is possible to refer ACO edge detection algorithm [32] or modified version of it [33]. Also a new system is made by S.M.H. Mousavi and Marwa Alkharaiz in 2017 [34] which was so robust against different type of noises VS traditional edge detection algorithms.

#### **Proposed method**

The paper presents a new robust system for edge detection, which could overcome different type of noises, in different color spaces.

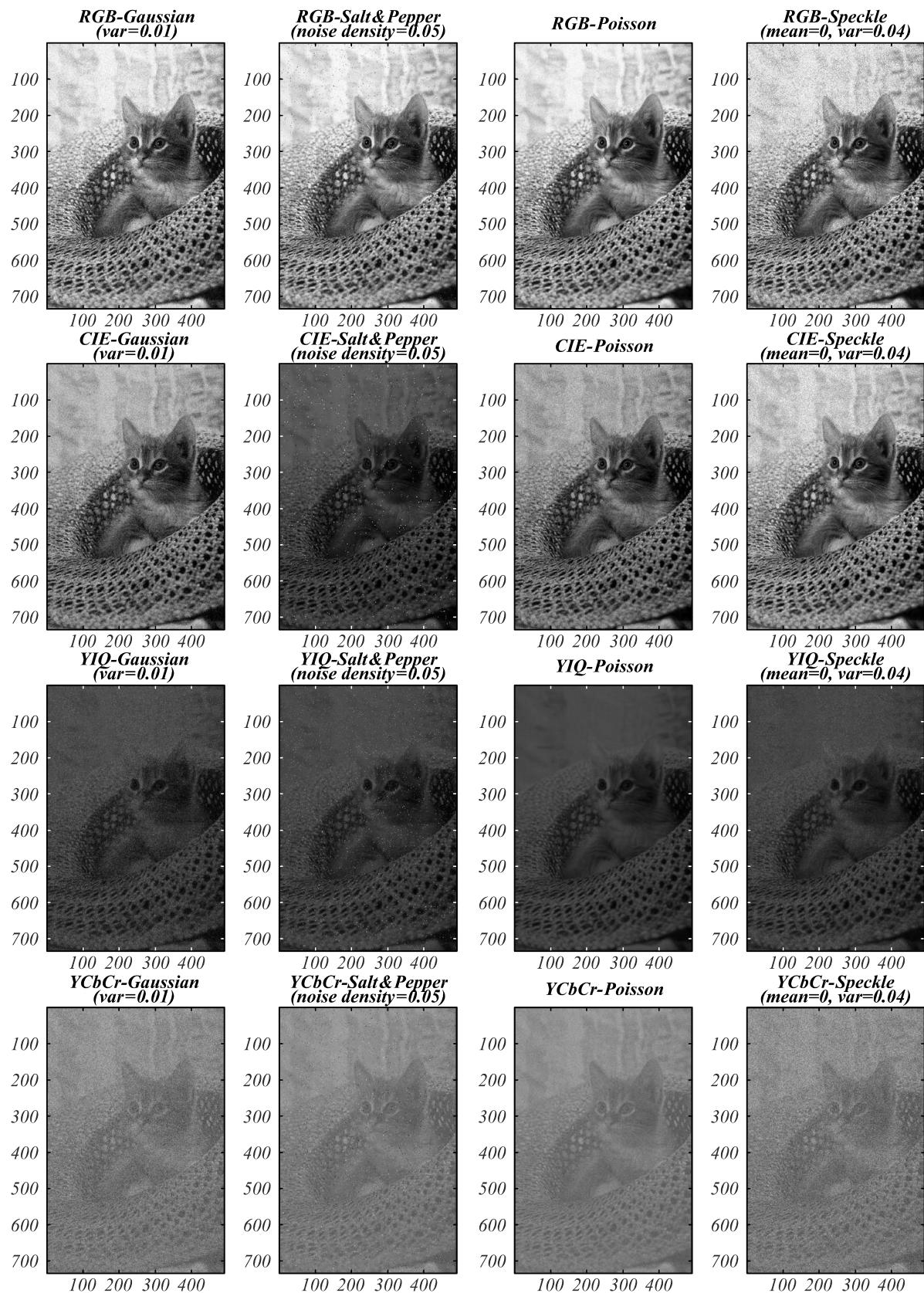


Fig. 2. Cat test image, polluted with Gaussian, Impulse, Poisson and Speckle noises in four color spaces

System starts with receiving gray and color benchmark test images and also manually recorded depth images for input. Four type of noises, which includes Gaussian, Impulse (Salt & pepper), Poisson and Speckle will be added to the test image with specified parameters. Next process is to converting input image into RGB, CIE (Lab), YIQ and YCbCr color spaces [9–11] and also splitting the channels to three for further processing. Applying low pass (smoothing) median filter and high pass (sharpening) un-sharp masking filters for each channels are as shown in Figure 3 on “shoulderCR”. The main purpose of using median filter is to fix the noises. And also in low intensity noisy images, smoothing the image for next step of sharpening is require to use median filter (it helps to sharpening process by emphasizing on edges). After filtering, it is time to combine or join splitted channels to one channel as next process needed. In this step, color images turn to gray image for edge detection process by modified Ant Colony Optimization (ACO) edge detection algorithm [33] (using default parameters based on the main paper). After edge detection, some holes, open areas or disconnected lines appears in some color spaces which should be fixed in post processing stage. Post processing stage is consists of erosion and closing morphological operation which fix the issues appeared in last step. Finally edge detected image should be compared and evaluated with other edge detection algorithms such as Canny [22], Zerocross [23], Log [24], Roberts [25] Prewitt [26], Sobel [27], Fuzzy [31], ACO [32] and [34] in different color spaces and with different types of noises. Also this step includes statistical validation for proposed method results and other edge detection algorithms in having noise and not having noise condition with different Image Quality Assessment (IQA) metrics such as Edge Based Structural Similarity (ESSIM) [35], Peak Signal to Noise Ratio (PSNR) [36], Mean Square Error (MSE) [37], EME (Enhancement Measure) [38], Edge Based Image Quality Assessment (EBIQA) [39], Non-Shift Edge Based Ratio (NSER) [40] and Gradient Conduction Mean Square Error (GCMSE) [41]. Modified ACO edge detection algorithm and IQA metrics are described further. Figure 4 represents proposed edge detection system’s flow chart (made by Microsoft Visio software) and Figure 5 shows proposed method result on a manually recorded depth image using Kinect V.2 in RGB color space (Poisson noise). Also Figure 6 shows proposed method steps on “boy” benchmark test image in YIQ color space (Impulse noise). Also it is mentionable that, Q channel in steps: “Q-Noisy” and “Q- Median Filter” might not be visible to human eyes, but valuable data is hidden behind the scene which will be visible at the end of the process.

#### *Filters*

Our system is based on very important pre and post-processing actions such as smoothing low pass, sharping high pass filters, morphological operations like erosion and closing, using a custom filter for evolutionary edge detection and some similar post-processing (but not all).

#### 1) Median filter

The median filter (5) is a nonlinear digital filtering technique, often used to remove noise from an image or signal. It is widely used for removing noises (especially salt & pepper or impulse noise). The main idea of the median filter is to run through the signal or image entry by entry, replacing each entry with the median of neighboring entries [42].

$$y[m, n] = \text{median}\{x[i, j], (i, j) \in w\}, \quad (5)$$

where w represents a neighborhood defined by the user, centered on location  $[m, n]$  in the image.

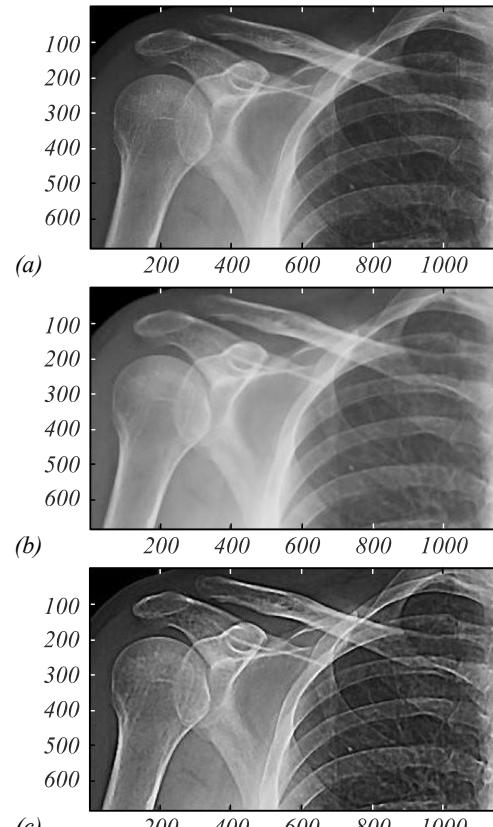


Fig. 3. (a) Original shoulderCR.jpg; (b) median filter on (a) with  $9 \times 9$  neighboring pixels; (c) unsharp masking with radius 5 and amount 3 parameters

#### 2) Un-sharp masking

This technique is using to sharpening the image and effects like using high pass filter on image. Un-sharp means smooth or not sharp, and masking the un-sharp (6) is the opposite of un-sharp or not un-sharp. So it means sharp, which is applied to the edges of the image and sharp it. This technique is controlled by three main parameters which are, amount, radius and threshold. Amount is listed as a percentage and controls the magnitude of each overshoot (how much darker and how much lighter the edge borders become). Radius affects the size of the edges to be enhanced or how wide the edge rims become, so a smaller radius enhances smaller-scale detail. Threshold controls the minimal brightness change that will be sharpened or how far apart adjacent tonal values have to be before the filter does anything [43].

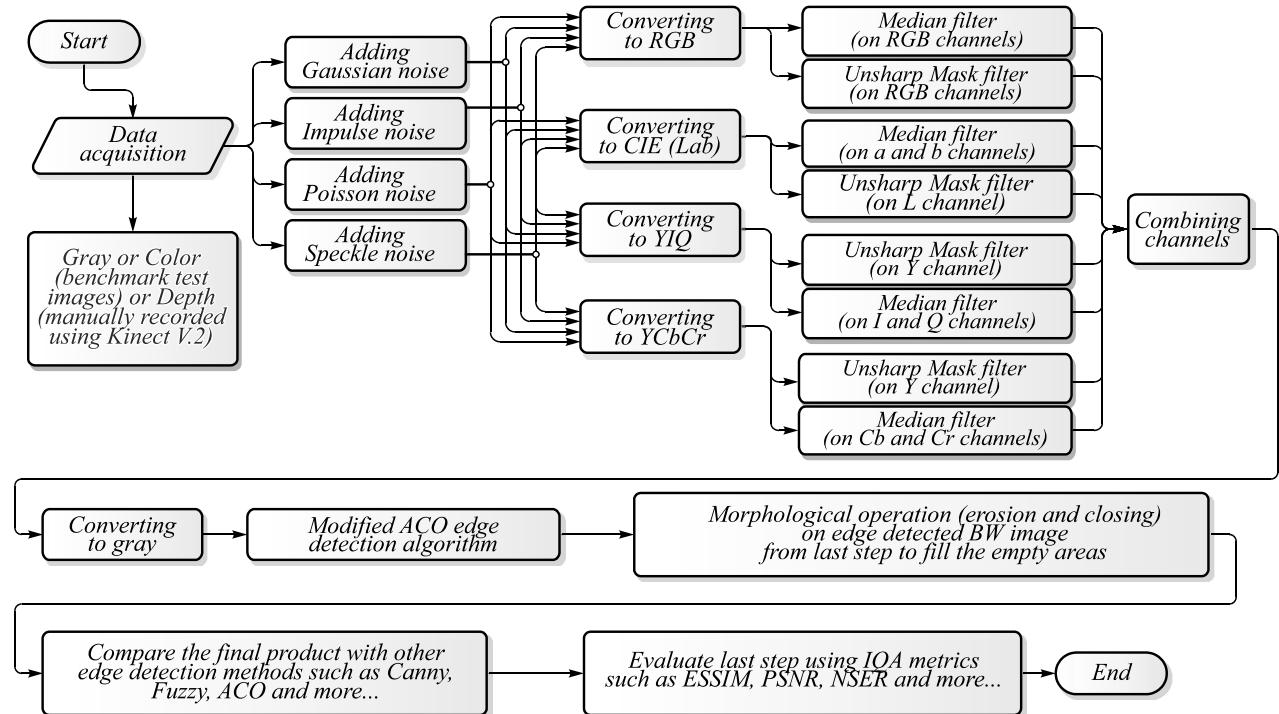


Fig. 4. Proposed method's flowchart

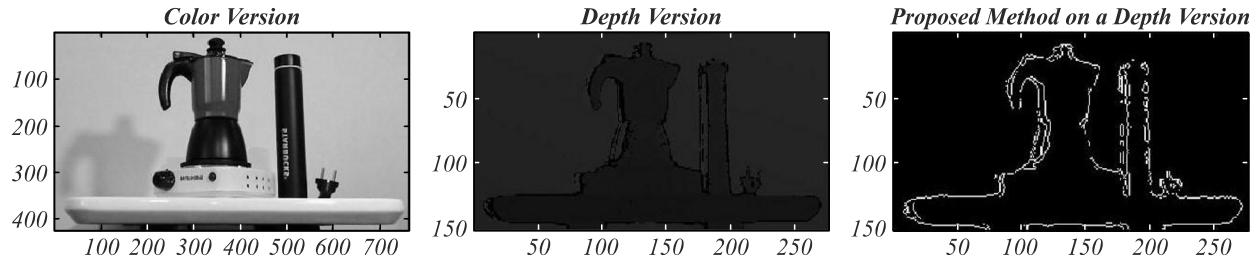


Fig. 5. Proposed method result on a manually recorded depth image using Kinect V.2 in RGB color space

Un-sharp masking produce and image  $g(x,y)$  from an image  $f(x,y)$  via:

$$g(x,y) = f(x,y) - f_{smooth}(x,y), \quad (6)$$

where  $f_{smooth}(x,y)$  is smoother version of  $f(x,y)$ .

#### Morphology operations

Morphological operators often take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement). They process objects in the input image based on characteristics of its shape, which are encoded in the structuring element. Usually, the structuring element is sized  $3 \times 3$  and has its origin at the center pixel. It is shifted over the image and at each pixel of the image its elements are compared with the set of the underlying pixels. If the two sets of elements match the condition defined by the set operator, the pixel underneath the origin of the structuring element is set to a pre-defined value (0 or 1 for binary images).

Four important morphological operation which we are going to use some of them are dilation (grow image regions), erosion (shrink image regions), opening (structured removal of image region boundary pixels) and closing (structured filling in of image region boundary pixels) [14]. For more information about morphological opera-

tions, it can be referred to [14]. Figure 7 represents using these operations on “girlface” edge detected benchmark test image. We are going to use these operations for post-processing purposes.

Let  $E$  be a Euclidean space or an integer grid, and  $A$ , a binary image in  $E$ . The erosion of the binary image  $A$  by the structuring element  $B$  is defined by (7): $\pm$

$$A - B = \{z \in E | B_z \subseteq B\}, \forall z \in E, \quad (7)$$

where  $B_z$  is the translation of  $B$  by the vector  $z$ . The erosion of  $A$  by  $B$  is also given by the (8) expression:

$$A - B = \bigcap_{b \in B} A - b. \quad (8)$$

The dilation of  $A$  by the structuring element  $B$  is defined by (9):

$$A \oplus B = \bigcup_{a \in A} B_a. \quad (9)$$

The closing of  $A$  by  $B$  is obtained by the dilation of  $A$  by  $B$ , followed by erosion of the resulting structure by  $B$  (10):

$$A \bullet B = (A \oplus B) - B. \quad (10)$$

**Modified ant colony optimization (ACO)  
edge detection**

ACO edge detection algorithm, generate a pheromone matrix which represents edge information at each pixel position on the routes shaped by ants dispatched on the image. Ants try to find possible edges by using a heuristic information based on the degree of edginess of each pixel. The modified ACO-based edge detection [33] also uses the fuzzy clustering (FCM) to determine whether a pixel is edge or not. The algorithm consists of three main steps of:

- Initialization

Artificial ants are distributed over the image, they move from one pixel to another using the transition probability which depends on local intensity.

- Iteratively update process
- Please refer to [33].
- Decision process.

Pheromone matrix is used for decision making that which pixel is to be considered as edge pixel by using Ostu method of thresholding.

Figure 8 represents output of modified ACO edge detection algorithm on “MR” benchmark color test image with defined parameters in the main paper [33].

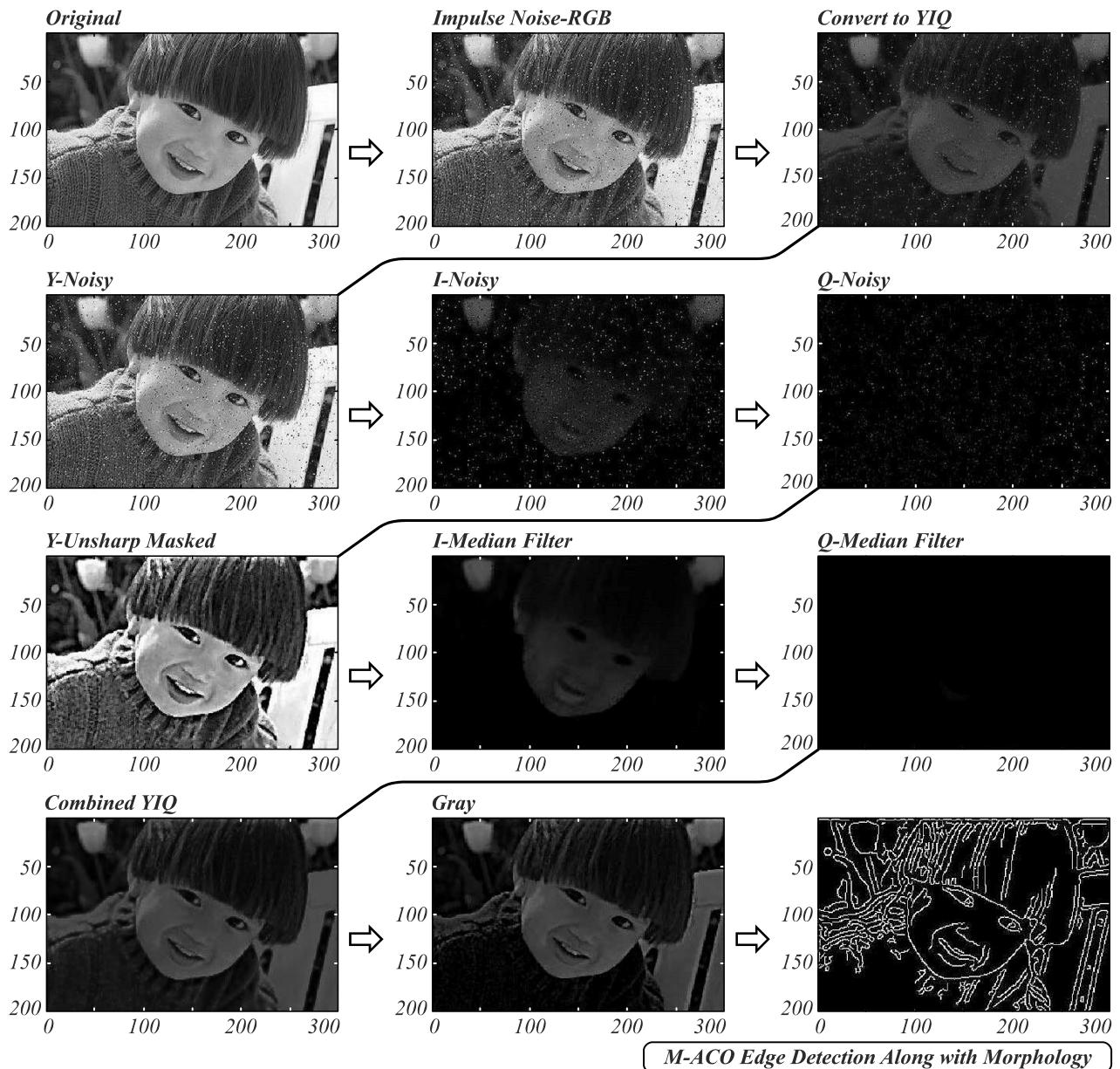


Fig. 6. Proposed method steps on “boy” benchmark test image in YIQ color space

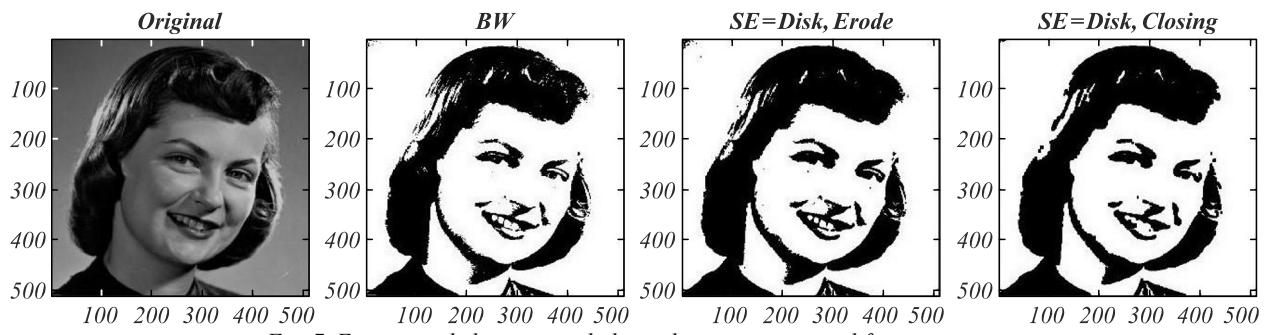


Fig. 7. Erosion and closing morphological operations on girl face test image



Fig. 8. Modified ACO edge detection algorithm result on MR color test image (from left to right)

### Validation and results

In this section, proposed edge detection system validates suing 7 edge base and non-edge based Image Quality Assessment (IQA) metrics, and compare with the other famous edge detection algorithms so far in same condition. This experiments is done, under effect of four different noises (Gaussian, Salt & pepper, Poisson and Speckle) which mentioned before. Parameters on noises are: Gaussian noise had variance of 0.01, Salt and pepper noise had noise density of 0.05 and Speckle noise had mean of 0 and variance of 0.04. All the experiments preforms for four color spaces (RGB, CIE (Lab), YIQ and YCbCr) to see in which color space, system works better

on 10 different color benchmark test image and 5 depth images. All calculations and validations are done using Matlab software. Figure 9 represents 10 different color benchmark test images for validating the system. Also Figure 10 shows 5 depth images which is recorded manually using Kinect V.2 sensor for validation purpose. As it is clear in Figure 10, manually recorded depth data are presented on their actual depth structure at the left side of their color version. So it might not be analyzable in printed version and should be used electronic version of the image for full details. Depth data are presented just for the note of the reader, and their true structure is recognizable by the computer.



Fig. 9. 10 different color benchmark test images used for validating the proposed method

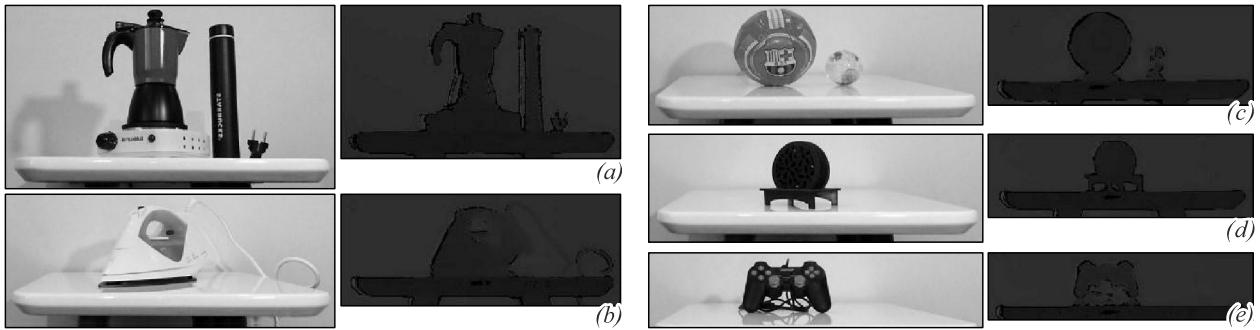


Fig. 10. 5 depth images (along with color format) which is recorded manually using Kinect V.2 sensor for validation purpose  
 ((a) Teapot, (b) Iron, (c) Balls, (d) Coasters and (e) Joystick)

#### Image quality assessment factors

Image Quality Assessment (IQA) factors could compare the difference between two almost similar images based on images characteristics in numerical way. This similarity could be in edges, colors, brightness, shape, parts movement and more. Some of these factors which are based on edges (based on the paper subject) and some other factors will be discussed for using in validation section.

#### 1) Edge Based Structural Similarity (ESSIM)

$$\text{ESSIM} = \text{function}(l, (I_1, I_2), c(I_1, I_2), e(I_1, I_2)). \quad (11)$$

It is the edge oriented version of SSIM metric [44]. The following steps do the computation of this metric:

- Vertical and Horizontal maps are created by the use of Sobel operator.
- Image is sub-sampled to  $16 \times 16$  blocks.
- Histogram of edge direction is created according to the sum of amplitudes with similar (1/8) directions.
- Using standard deviations of obtained histograms the edge factor is calculated in (12).

$$e(I_1, I_2) = \frac{\sigma_{II_2} + C_3}{\sigma_{I_1} \sigma_{I_2} + C_3}. \quad (12)$$

Application of Sobel filter is considered to be time efficient, effective and simple. Provides better performance than PSNR and SSIM in evaluation and had a good response to image degradation.

#### 2) Peak Signal to Noise Ratio (PSNR)

It indicates the level of losses or signals integrity. The Peak Signal to Noise Ratio (PSNR) (13) block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image [36].

$$\text{PSNR} = 10 \log_{10} \frac{L^2}{\text{MSE}} \quad (13)$$

in which L determines the range of value, which a pixel could have. Its unit is DB, and has a limit of 50. The proper value is between 20 and 50.

#### 3) Mean Square Error (MSE)

The Mean Square Error (MSE) (14) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error [37].

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i, j) - Y(i, j)]^2. \quad (14)$$

In which, X and Y are two arrays with the size of  $M \times N$ . To any extent Y resembles X, the value of MSE will reduce.

#### 4) EME (Enhancement Measure)

Measure of enhancement (15) is another image quality metric or assessment. This measure is related with concepts of the Webers Law of the human visual system. It helps to choose (automatically) the best parameters and transform [38].

$$\text{EME} = \max_{\Phi \in \{\Phi\}} \chi(\text{EME}(\Phi)),$$

which  $\max_{\Phi \in \{\Phi\}} = \chi \left( \frac{1}{k_1 k_2} \sum_{l=1}^{k_2} \sum_{k=1}^{k_1} 20 \log \frac{I_{\max;k;l}^o}{I_{\min;k;l}^o} \right)$ . (15)

Let an image  $x(n, m)$  be splitted into  $k_1 k_2$  blocks  $w_{k,l}$  ( $i, j$ ) of size  $l_1 \times l_2$  and let  $\{\Phi\}$  to be a given class of orthogonal transform used for image enhancement. Also  $I_{\max;k;l}^o$  and  $I_{\min;k;l}^o$  are respectively minimum and maximum of the image  $x(m, m)$  inside the block  $w_{k,l}$ . The function  $\chi$  is the sign function.

#### 5) Edge Based Image Quality Assessment (EBIQA)

Edge preservation one of the most important aspect during the human visual assessment. Attar, Shahbahrami, and Rad proposed Edge Based Image Quality Assessment (EBIQA) technique that aims to operate on the human perception of the features [39]. Steps are:

- Edges locations are identified utilizing Sobel edge detector technique in both images.
- The  $16 \times 16$  pixel window size vectors are formed at each image based on (16) and (17), -where  $I_1$  is reference image and  $I_2$  is the image under test.

$$I_1 = (O, AL, PL, N, VHO), \quad (16)$$

$$I_2 = (O, AL, PL, N, VH), \quad (17)$$

which ' $O$ ' stands for edge orientation in the image, which is a total number of edges. ' $AL$ ' or value of the average length of all edges. ' $PL$ ' estimates the number of pixels with a similar level of intensity values. ' $N$ ' responds for sum of pixels, which form edges. ' $VH$ ' corresponds for sum of pixels, which form edges in either vertically or horizontally located edges. Finally, we estimate  $EBIQA$  by (18), where average Euclidean distance of proposed vectors is estimated.

$$EBIQA = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \sqrt{(I_1 - I_2)^2}. \quad (18)$$

#### 6) Non-Shift Edge Based Ratio (NSER)

This method is based on zero-crossings and was proposed by Zhang, Mou and Zhang [40]. It makes decisions upon edge maps. Steps are as follow:

- Gaussian kernel is applied to the interesting images on different standard deviation scales to identify edges.
- Then operation is performed between two images. Ratio of the common edge number located by initial edge number is found using (19):

$$p_i = \|I_1 \cap I_2\| / \|I_1\|. \quad (19)$$

The result of this operation is normalized by log function to improve correlation factor (20):

$$NSER(I_1, I_2) = -\sum_{i=1}^N \log 10(1 - p_i). \quad (20)$$

#### 7) Gradient Conduction Mean Square Error (GCMSE)

Lopez-Randulfe et al. in their paper introduced an edge aware metric based on  $MSE$  [37]. In this algorithm weighted sum of gradients (distance pixels) is taken into account [41].  $GCMSE$  Provides better performance than  $MSE$  and  $SSIM$ . The main steps are as follow:

- Directional gradients are estimated in four directions using (21) and then average value  $G_p$  is found. Note that the results are optimized by the coefficient  $k$ :

$$G = \frac{(I_2 - I_1)^2}{(I_2 - I_1)^2 + k^2}. \quad (21)$$

The  $GCMSE$  is estimated based on (22):

$$GCMSE = \frac{\sum_{x=1}^m \sum_{y=1}^n [(I_2(x, y) - I_1(x, y)) G_p]^2}{C1 + \sum_{x=1}^m \sum_{y=1}^n G_p}. \quad (22)$$

#### Validation's results

As we have 7 IQA metrics, 15 color and depth test images, 9 comparing methods, 4 noise types and 4 color spaces, representing this amount of combinations in table format requires 112 ( $15 \times 11$ ) tables which is not possible to fit in the paper. Due to that, we decided to select 7 tables which covers all areas of changes and results from these 112 tables. But as we analyzed the results, proposed method was so robust versus other methods, as some of

these results are represented in Tables 1–7. Also we selected 7 table which covers all 7 IQA metrics with (diverse noise and color spaces). Comparisons are based on un-noised edge detected versions on the test images versus noisy versions using IQA metrics. For example and for baboon test image in CIE color space with Gaussian noise, calculated with ESSIM IQA metric, edge detection using Sobel operators performs and the acquired image is going to compare with un-noised version of the baboon test image. This happens for all edge detection methods in same condition on same test image. Then it is possible to compare acquired results in bigger view on tables.

Note that all the comparing results are normalized between 0-1 ranges for better understanding (even PSNR).

As the result are closer to 1, it shows better results or highest similarity (depend on the metric's definition) and vice versa. Tables 1 to 7 represent evaluations results from 7 IQA metrics for comparing different edge detection methods (including proposed method) on diverse test images (color and depth) using ESSIM, PSNR, MSE, EME, EBIQA, NSER and GCMSE metrics, respectively with different noises and in different color spaces. Figure 11 shows the visual comparison between proposed method and other methods in YCbCr color space which Brain1 test image is polluted with Speckle noise. It is mentionable that lower and upper threshold value for canny edge detector was [0.01, 0.12]. As it is clear in Figure 11, a lot of details are presented on each method, due to brain's structure and lines. So it might not be analyzable in printed version and should be used electronic version of the image for full details. After making different types of edge detection techniques in decades by other researchers, now it is time to pay to the details after all to make a promising new method.

#### **Conclusion discussion and suggestions**

With using evolutionary algorithm, it is possible to achieve proper result in edge detection subject, just like other subjects in artificial intelligence. It is concluded that, different color spaces have different effects on different edge detection methods and different results could be acquired. But with a precise look at the results, we can see that RGB and YIQ are the best color spaces for edge detection. Also YCbCr color space is not so bad, but CIE color space is better for depth data than color. Proposed evolutionary edge detection had perfect robustness against Impulse and Poisson noises in different color spaces. Speckle and Gaussian noises are in next ranks in relation to robustness. Validation results represents that proposed method returned more results close to 1 versus other methods for IQA metrics, which this shows the power of proposed method in different conditions. In the other hand, proposed method returned promising and satisfactory results on manually recorded depth images with Kinect V.2 that could open ways in this era too. Also system is almost fast and can work in real time purposes too. It is suggested to use other evolutionary algorithms for edge detection purposes such as ABC [45], Bat algorithm [46], PSO [47], GGO [48] and ICA [49], which may perform better in that way.

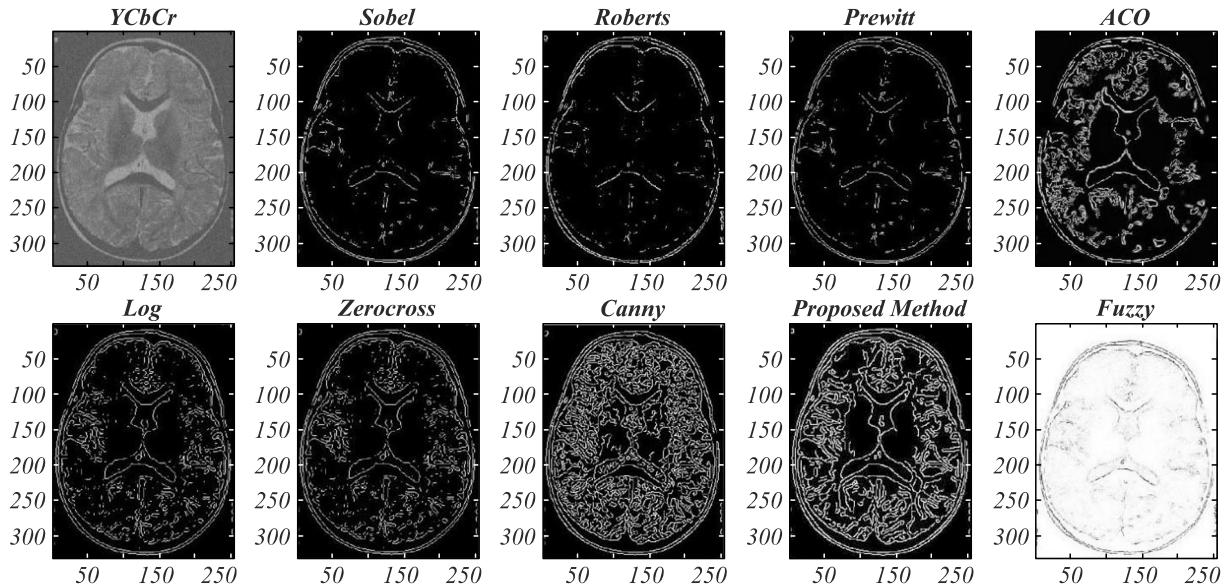


Fig. 11. The visual comparison between proposed method and other methods in YCbCr color space which Brain1 test image is polluted with Speckle noise

Table 1. Comparing various methods with ESSIM (Impulse noise in YIQ color space)

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.623	0.614	0.523	0.762	0.715	0.786	0.803	0.802	0.811	0.762
2	<b>boat</b>	0.515	0.504	0.413	0.652	0.605	0.676	0.693	0.696	0.701	0.652
3	<b>boy</b>	0.748	0.736	0.645	0.884	0.837	<b>0.908</b>	<b>0.925</b>	<b>0.924</b>	<b>0.933</b>	0.884
4	<b>Brain1</b>	0.701	0.694	0.603	0.842	0.795	0.866	0.883	0.882	<b>0.897</b>	0.842
5	<b>cat</b>	0.592	0.588	0.497	0.736	0.689	0.760	0.777	0.776	0.785	0.736
6	<b>girl face</b>	0.681	0.679	0.588	0.827	0.780	0.851	<b>0.898</b>	0.865	<b>0.896</b>	0.827
7	<b>Lenna</b>	0.622	0.615	0.524	0.763	0.716	0.787	0.804	0.803	0.812	0.763
8	<b>monarch</b>	0.724	0.714	0.623	0.862	0.815	0.886	<b>0.903</b>	<b>0.904</b>	<b>0.911</b>	0.862
9	<b>MR</b>	0.773	0.762	0.671	0.910	0.863	<b>0.934</b>	<b>0.951</b>	<b>0.950</b>	<b>0.959</b>	<b>0.910</b>
10	<b>shoulderCR</b>	0.574	0.561	0.470	0.709	0.662	0.733	0.750	0.749	0.758	0.709
Depth											
11	<b>Balls</b>	0.668	0.658	0.567	0.806	0.759	0.830	0.847	0.843	0.855	0.806
12	<b>Coasters</b>	0.627	0.619	0.528	0.767	0.720	0.791	0.808	0.807	0.816	0.767
13	<b>Iron</b>	0.697	0.681	0.590	0.829	0.782	0.853	0.870	0.862	0.878	0.829
14	<b>Joystick</b>	0.605	0.596	0.505	0.744	0.697	0.768	0.785	0.784	0.793	0.744
15	<b>Teapot</b>	0.718	0.704	0.613	0.852	0.805	0.876	<b>0.893</b>	<b>0.891</b>	<b>0.901</b>	0.852

Table 2. Comparing various methods with PSNR (Poisson noise in RGB color space)

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.574	0.702	0.730	0.768	0.761	0.902	0.920	0.919	<b>0.921</b>	0.876
2	<b>boat</b>	0.454	0.582	0.610	0.648	0.642	0.782	0.800	0.799	0.801	0.756
3	<b>boy</b>	0.633	0.761	0.789	0.827	0.822	<b>0.961</b>	<b>0.979</b>	<b>0.978</b>	<b>0.980</b>	<b>0.935</b>
4	<b>Brain1</b>	0.548	0.676	0.704	0.742	0.734	0.876	<b>0.904</b>	<b>0.893</b>	<b>0.895</b>	0.850
5	<b>cat</b>	0.538	0.666	0.694	0.732	0.726	0.866	0.884	0.883	0.885	0.840
6	<b>girl face</b>	0.543	0.671	0.699	0.737	0.735	0.871	0.889	0.888	<b>0.890</b>	0.845
7	<b>Lenna</b>	0.565	0.693	0.721	0.759	0.753	<b>0.893</b>	<b>0.911</b>	<b>0.910</b>	<b>0.912</b>	0.867
8	<b>monarch</b>	0.625	0.753	0.781	0.819	0.816	<b>0.953</b>	<b>0.971</b>	<b>0.970</b>	<b>0.972</b>	<b>0.927</b>
9	<b>MR</b>	0.640	0.768	0.796	0.834	0.828	<b>0.968</b>	<b>0.986</b>	<b>0.985</b>	<b>0.987</b>	<b>0.942</b>
10	<b>shoulderCR</b>	0.511	0.639	0.667	0.705	0.697	0.839	0.857	0.856	0.858	0.813
Depth											
11	<b>Balls</b>	0.526	0.654	0.682	0.720	0.718	0.854	0.872	0.871	0.873	0.828
12	<b>Coasters</b>	0.550	0.678	0.706	0.744	0.738	0.878	<b>0.896</b>	<b>0.895</b>	<b>0.897</b>	0.852
13	<b>Iron</b>	0.571	0.699	0.727	0.765	0.758	<b>0.899</b>	<b>0.917</b>	<b>0.916</b>	<b>0.918</b>	0.873
14	<b>Joystick</b>	0.506	0.634	0.662	0.700	0.699	0.834	0.852	0.851	0.853	0.808
15	<b>Teapot</b>	0.589	0.717	0.745	0.783	0.770	<b>0.917</b>	<b>0.935</b>	<b>0.934</b>	<b>0.936</b>	<b>0.893</b>

Table 3. Comparing various methods with MSE (Gaussian noise in CIE color space)

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.568	0.594	0.604	0.722	0.653	0.730	0.731	0.700	0.746	0.726
2	<b>boat</b>	0.548	0.574	0.580	0.702	0.633	0.710	0.711	0.680	0.723	0.706
3	<b>boy</b>	0.712	0.638	0.641	0.866	0.797	0.874	0.875	0.844	<b>0.890</b>	0.870
4	<b>Brain1</b>	0.649	0.575	0.583	0.803	0.734	0.811	0.812	0.781	0.827	0.807
5	<b>cat</b>	0.560	0.586	0.592	0.714	0.645	0.722	0.723	0.692	0.738	0.718
6	<b>girl face</b>	0.659	0.585	0.593	0.813	0.744	0.821	0.822	0.791	0.844	0.825
7	<b>Lenna</b>	0.560	0.586	0.594	0.714	0.645	0.722	0.723	0.692	0.738	0.718
8	<b>monarch</b>	0.696	0.622	0.635	0.750	0.781	<b>0.898</b>	0.859	0.828	0.874	<b>0.891</b>
9	<b>MR</b>	0.708	0.634	0.642	0.762	0.793	0.870	0.871	0.840	<b>0.926</b>	0.876
10	<b>shoulderCR</b>	0.574	0.593	0.606	0.721	0.652	0.729	0.730	0.699	0.743	0.725
Depth											
11	<b>Balls</b>	0.566	0.592	0.601	0.720	0.651	0.728	0.729	0.698	0.743	0.724
12	<b>Coasters</b>	0.565	0.591	0.600	0.719	0.650	0.727	0.728	0.697	0.739	0.723
13	<b>Iron</b>	0.708	0.553	0.561	0.781	0.712	<b>0.889</b>	<b>0.890</b>	0.759	<b>0.895</b>	0.785
14	<b>Joystick</b>	0.574	0.600	0.610	0.728	0.659	0.736	0.737	0.706	0.750	0.733
15	<b>Teapot</b>	0.559	0.583	0.591	0.711	0.642	0.719	0.720	0.689	0.730	0.715

Table 4. Comparing various methods with EME (Speckle noise in YCbCr color space)

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.603	0.517	0.455	0.803	0.715	0.828	0.817	0.816	0.829	0.814
2	<b>boat</b>	0.505	0.419	0.357	0.705	0.617	0.730	0.705	0.718	0.731	0.716
3	<b>boy</b>	0.704	0.618	0.556	0.904	0.816	<b>0.929</b>	<b>0.928</b>	<b>0.917</b>	<b>0.930</b>	<b>0.915</b>
4	<b>Brain1</b>	0.625	0.539	0.477	0.825	0.737	0.85	<b>0.894</b>	0.838	<b>0.901</b>	0.836
5	<b>cat</b>	0.509	0.423	0.361	0.709	0.621	0.734	0.713	0.722	0.735	0.720
6	<b>girl face</b>	0.590	0.504	0.442	0.790	0.702	0.815	0.807	0.803	0.816	0.801
7	<b>Lenna</b>	0.584	0.498	0.436	0.784	0.696	0.809	0.782	0.797	0.810	0.795
8	<b>monarch</b>	0.655	0.569	0.507	0.855	0.767	0.880	0.858	0.868	0.881	0.866
9	<b>MR</b>	0.673	0.587	0.525	0.873	0.785	<b>0.898</b>	0.872	0.886	<b>0.890</b>	0.884
10	<b>shoulderCR</b>	0.512	0.426	0.364	0.712	0.624	0.737	0.718	0.725	0.738	0.723
Depth											
11	<b>Balls</b>	0.589	0.503	0.441	0.789	0.701	0.814	0.793	0.802	0.815	0.800
12	<b>Coasters</b>	0.550	0.464	0.402	0.750	0.662	0.775	0.753	0.762	0.776	0.761
13	<b>Iron</b>	0.612	0.526	0.464	0.812	0.724	0.837	0.813	0.825	0.838	0.823
14	<b>Joystick</b>	0.537	0.451	0.389	0.737	0.649	0.762	0.760	0.750	0.763	0.748
15	<b>Teapot</b>	0.582	0.496	0.434	0.782	0.694	0.807	0.782	0.741	0.808	0.793

Table 5. Comparing various methods with EBIQA (Impulse noise in CIE color space)

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.567	0.602	0.616	0.577	0.569	0.642	0.725	0.724	0.732	0.664
2	<b>boat</b>	0.617	0.552	0.565	0.627	0.619	0.593	0.675	0.643	0.682	0.644
3	<b>boy</b>	0.629	0.564	0.674	<b>0.910</b>	0.794	0.807	<b>0.921</b>	<b>0.893</b>	<b>0.924</b>	0.886
4	<b>Brain1</b>	0.537	0.572	0.585	0.647	0.639	0.715	0.795	0.762	0.882	0.744
5	<b>cat</b>	0.583	0.616	0.574	0.591	0.583	0.655	0.739	0.706	0.746	0.738
6	<b>girl face</b>	0.538	0.573	0.585	0.648	0.640	0.716	0.796	0.764	0.803	0.745
7	<b>Lenna</b>	0.598	0.533	0.548	0.608	0.600	0.676	0.756	0.730	0.763	0.655
8	<b>monarch</b>	0.583	0.618	0.633	0.693	0.685	<b>0.898</b>	0.879	0.806	<b>0.898</b>	0.888
9	<b>MR</b>	0.595	0.630	0.648	0.705	0.697	0.773	0.853	0.816	0.880	<b>0.892</b>
10	<b>shoulderCR</b>	0.597	0.532	0.549	0.607	0.599	0.678	0.755	0.722	0.762	0.654
Depth											
11	<b>Balls</b>	0.624	0.559	0.574	0.634	0.626	0.709	0.782	0.749	0.789	0.790
12	<b>Coasters</b>	0.602	0.537	0.557	0.612	0.604	0.680	0.760	0.726	0.767	0.759
13	<b>Iron</b>	0.580	0.615	0.630	0.590	0.582	0.660	0.738	0.705	0.745	0.637
14	<b>Joystick</b>	0.599	0.534	0.546	0.609	0.608	0.680	0.757	0.726	0.764	0.656
15	<b>Teapot</b>	0.627	0.562	0.677	0.907	<b>0.899</b>	0.801	<b>0.895</b>	<b>0.897</b>	<b>0.924</b>	<b>0.894</b>

*Table 6. Comparing various methods with NSER (Poisson noise in YIQ color space)*

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.513	0.496	0.678	0.773	0.806	0.837	0.859	0.843	0.874	0.796
2	<b>boat</b>	0.450	0.433	0.615	0.718	0.743	0.774	0.796	0.78	0.811	0.733
3	<b>boy</b>	0.558	0.541	0.723	0.827	0.851	0.882	<b>0.904</b>	0.888	<b>0.919</b>	0.841
4	<b>Brain1</b>	0.535	0.518	0.700	0.798	0.828	0.859	0.881	0.865	<b>0.896</b>	0.818
5	<b>cat</b>	0.423	0.406	0.588	0.687	0.716	0.747	0.769	0.753	0.784	0.706
6	<b>girl face</b>	0.506	0.489	0.671	0.769	0.799	0.83	0.852	0.836	0.867	0.789
7	<b>Lenna</b>	0.474	0.457	0.639	0.738	0.767	0.798	0.82	0.804	0.835	0.757
8	<b>monarch</b>	0.567	0.550	0.732	0.880	0.860	<b>0.891</b>	<b>0.913</b>	<b>0.897</b>	<b>0.928</b>	0.850
9	<b>MR</b>	0.539	0.522	0.704	0.802	0.832	0.863	0.885	0.869	<b>0.900</b>	0.822
10	<b>shoulderCR</b>	0.340	0.323	0.505	0.609	0.633	0.664	0.686	0.670	0.701	0.623
Depth											
11	<b>Balls</b>	0.455	0.438	0.62	0.715	0.748	0.779	0.801	0.785	0.816	0.738
12	<b>Coasters</b>	0.464	0.447	0.629	0.722	0.757	0.788	0.81	0.794	0.825	0.747
13	<b>Iron</b>	0.478	0.461	0.643	0.741	0.771	0.802	0.824	0.808	0.839	0.761
14	<b>Joystick</b>	0.424	0.407	0.589	0.683	0.717	0.748	0.77	0.754	0.785	0.707
15	<b>Teapot</b>	0.561	0.544	0.726	0.823	0.854	<b>0.895</b>	<b>0.907</b>	<b>0.891</b>	<b>0.922</b>	0.844

*Table 7. Comparing various methods with GCMSE (Speckle noise in YCbCr color space)*

	Test Images	Sobel	Roberts	Prewitt	Log	Cross	Canny	[34]	ACO	Proposed	Fuzzy
Color											
1	<b>baboon</b>	0.556	0.521	0.488	0.659	0.691	0.726	0.719	0.764	0.786	0.751
2	<b>boat</b>	0.443	0.411	0.378	0.545	0.581	0.616	0.609	0.654	0.673	0.641
3	<b>boy</b>	0.623	0.593	0.56	0.723	0.763	0.798	0.791	0.836	0.853	0.823
4	<b>Brain1</b>	0.613	0.581	0.548	0.716	0.751	0.786	<b>0.879</b>	0.824	<b>0.896</b>	0.811
5	<b>cat</b>	0.395	0.365	0.332	0.496	0.535	0.570	0.563	0.608	0.724	0.595
6	<b>girl face</b>	0.633	0.606	0.573	0.736	0.776	0.811	0.804	0.849	0.866	0.836
7	<b>Lenna</b>	0.576	0.542	0.509	0.677	0.712	0.747	0.740	0.785	0.805	0.772
8	<b>monarch</b>	0.579	0.549	0.516	0.679	0.719	0.754	0.747	0.792	0.803	0.779
9	<b>MR</b>	0.709	0.673	0.640	0.804	0.843	0.878	<b>0.897</b>	<b>0.916</b>	<b>0.933</b>	<b>0.903</b>
10	<b>shoulderCR</b>	0.508	0.478	0.445	0.604	0.648	0.683	0.676	0.721	0.736	0.708
Depth											
11	<b>Balls</b>	0.550	0.525	0.492	0.654	0.695	0.730	0.723	0.768	0.785	0.755
12	<b>Coasters</b>	0.530	0.506	0.473	0.634	0.676	0.711	0.704	0.749	0.766	0.736
13	<b>Iron</b>	0.617	0.588	0.555	0.718	0.758	0.793	0.786	0.831	0.847	0.818
14	<b>Joystick</b>	0.618	0.583	0.55	0.719	0.753	0.788	0.781	0.826	<b>0.898</b>	0.813
15	<b>Teapot</b>	0.658	0.628	0.595	0.759	0.798	0.833	<b>0.896</b>	0.871	<b>0.899</b>	0.858

### References

- [1] Davis LS. A survey of edge detection techniques. Computer Graphics and Image Processing 1975; 4(3): 248-270.
- [2] Fogel DB. Evolutionary computation: the fossil record. Wiley-IEEE Press; 1998.
- [3] Dasarathy BV, Dasarathy H. Edge preserving filters – Aid to reliable image segmentation. SOUTHEASTCON'81 Proceedings of the Region 3 Conference and Exhibit 1981: 650-654.
- [4] Ren Ch-X, et al. Enhanced local gradient order features and discriminant analysis for face recognition. IEEE Transactions on Cybernetics 2016; 46(11): 2656-2669.
- [5] Liu Y, Ai H, Xu G-Y. Moving object detection and tracking based on background subtraction. Proc SPIE 2001; 4554: 62-66.
- [6] Leondes CT. Mean curvature flows, edge detection, and medical image segmentation. In Book: Leondes CT. Computational methods in biophysics, biomaterials, biotechnology and medical systems. Boston, MA: Springer-Verlag US; 2003: 856-870.
- [7] Pflug A, Christoph B. Ear biometrics: a survey of detection, feature extraction and recognition methods. IET Biometrics 2012; 1(2): 114-129.
- [8] Rosenberger M. Multispectral edge detection algorithms for industrial inspection tasks. 2014 IEEE International Conference on Imaging Systems and Techniques (IST) 2014: 232-236.
- [9] Tkalcic M, Tasic JF. Colour spaces: perceptual, historical and applicational background. The IEEE Region 8 EURO-CON 2003. Computer as a Tool 2003; 1: 304-308.
- [10] Chaves-González JM, et al. Detecting skin in face recognition systems: A colour spaces study. Digital Signal Processing 2010; 20(3): 806-823.
- [11] Gonzalez RC, Woods RE. Digital image processing. 3<sup>rd</sup> ed. Upper Saddle River, NJ: Prentice-Hall Inc; 2016.
- [12] Public-domain test images for homeworks and projects. Source: <<https://homepages.cae.wisc.edu/~ece533/images/>>.
- [13] C/Python/Shell programming and image/video processing/compression. Source: <<http://www.hlevkin.com/06testimages.htm>>.
- [14] Gonzales RC, Woods RE. Digital image processing. Boston, MA: Addison and Wesley Publishing Company; 1992.
- [15] Jain AK. Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice Hall; 1989.
- [16] Hasinoff SW. Photon, poisson noise. In Book: Ikeuchi K, ed. Computer vision. Boston, MA: Springer US; 2014: 608-610.

- [17] Jaybhay J, Shastri R. A study of speckle noise reduction filters. *Signal & Image Processing: An International Journal (SIPIJ)* 2015; 6.
- [18] Zhang Zh. Microsoft kinect sensor and its effect. *IEEE Multimedia* 2012; 19(2): 4-10.
- [19] Xtion PRO. Source: ([https://www.asus.com/3D-Sensor/Xtion\\_PRO/](https://www.asus.com/3D-Sensor/Xtion_PRO/)).
- [20] Keselman L, et al. Intel RealSense stereoscopic depth-cameras. Source: (<https://arxiv.org/abs/1705.05548>).
- [21] Primesense Carmine 1.09. Source: (<http://xtionprolive.com/primesense-carmine-1.09>).
- [22] Canny J. A computational approach to edge detection. In Book: Fischler MA, Firschein O, eds. *Readings in computer vision: issues, problems, principles, and paradigms*. San Francisco, CA: Morgan Kaufmann Publishers Inc; 1987: 184-203.
- [23] Haralick RM. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1984; 1: 58-68.
- [24] Lindeberg T. Scale selection properties of generalized scale-space interest point detectors. *Journal of Mathematical Imaging and Vision* 2013; 46(2): 177-210.
- [25] Roberts LG. Machine perception of three-dimensional solids. Diss PhD Thesis. Cambridge, MA; 1963.
- [26] Prewitt JMS. Object enhancement and extraction. *Picture Processing and Psychopictorics* 1970; 10(1): 15-19.
- [27] Sobel I, Feldman G. A 3x3 isotropic gradient operator for image processing, presented at a talk at the Stanford Artificial Project. In Book: Duda R, Hart P, eds. *Pattern classification and scene analysis*. John Wiley & Sons; 1968: 271-272.
- [28] Shih M-Y, Tseng D-Ch. A wavelet-based multiresolution edge detection and tracking. *Image and Vision Computing* 2005; 23(4): 441-451.
- [29] Lee J, Haralick R, Shapiro L. Morphologic edge detection. *IEEE Journal on Robotics and Automation* 1987; 3(2): 142-156.
- [30] Rajab, MI, Woolfson MS, Morgan SP. Application of region-based segmentation and neural network edge detection to skin lesions. *Computerized Medical Imaging and Graphics* 2004; 28(1): 61-68.
- [31] Akbari AS, Soraghan JJ. Fuzzy-based multiscale edge detection. *Electronics Letters* 2003; 39(1): 30-32.
- [32] Tian J, Yu W, Xie S. An ant colony optimization algorithm for image edge detection. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) 2008: 751-756.
- [33] Rajeswari R, Rajesh R. A modified ant colony optimization based approach for image edge detection. 2011 International Conference on Image Information Processing, 2011.
- [34] Mousavi SMH, Kharazi M. An edge detection system for polluted images by gaussian, salt and pepper, poisson and speckle noises. 4<sup>th</sup> National Conference on Information Technology, Computer & TeleCommunication 2017.
- [35] Chen G-H, et al. Edge-based structural similarity for image quality assessment. 2006 IEEE International Conference on Acoustics Speech and Signal Processing 2006; 2: II-II.
- [36] Wang Z, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 2004; 13(4): 600-612.
- [37] Lehmann EL, Casella G. *Theory of point estimation*. Springer Science & Business Media; 2006.
- [38] Agaian SS, Lentz KP, Grigoryan AM. A new measure of image enhancement. IASTED International Conference on Signal Processing & Communication 2000.
- [39] Attar A, Shahbahrami A, Rad RM. Image quality assessment using edge based features. *Multimedia Tools and Applications* 2016; 75(12): 7407-7422.
- [40] Zhang M, Mou X, Zhang L. Non-shift edge based ratio (NSER): An image quality assessment metric based on early vision features. *IEEE Signal Processing Letters* 2011; 18(5): 315-318.
- [41] López-Randulfe J, et al. A quantitative method for selecting denoising filters, based on a new edge-sensitive metric. 2017 IEEE International Conference on Industrial Technology (ICIT) 2017: 974-979.
- [42] Huang T, Yang G, Tang G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1979; 27(1): 13-18.
- [43] Polesel A, Ramponi G, Mathews VJ. Image enhancement via adaptive unsharp masking. *IEEE Transactions on Image Processing* 2000; 9(3): 505-510.
- [44] Wang Z, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 2004; 13(4): 600-612.
- [45] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06. Erciyes University, Turkey; 2005.
- [46] Yang X-S. A new metaheuristic bat-inspired algorithm. In Book: González JR, Pelta DA, Cruz C, Terrazas G, Krassnogor N, eds. *Nature inspired cooperative strategies for optimization (NISCO 2010)*. Berlin, Heidelberg: Springer; 2010: 65-74.
- [47] Kennedy J. Particle swarm optimization. In Book: Sammut C, Webb GI, eds. *Encyclopedia of Machine Learning*. Boston, MA: Springer US; 2011: 760-766.
- [48] Hossein Mousavi SM, Mirinezhad SY, Dezfoolian MH. Galaxy gravity optimization (GGO) an algorithm for optimization, inspired by comets life cycle. 2017 Artificial Intelligence and Signal Processing Conference (AISP) 2017: 306-315.
- [49] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. 2007 IEEE Congress on Evolutionary Computation 2007: 4661-4667.

#### *Author's information*

**Seyed Muhammad Hossein Mousavi** was born at Tehran, Iran in July 30, 1990. He received his M.Sc. degrees from Bu-Ali Sina University, in 2017 in the branch of Artificial Intelligence. He is expert in machine vision, depth image processing, fuzzy logic, evolutionary computation, evolutionary art, OCR, NLP, data mining, expert systems and micro facial expressions recognition using RGBD images. Personal website: <http://s-m-h-mousavi.ir/>.

E-mail: [mosavi.a.i.buali@gmail.com](mailto:mosavi.a.i.buali@gmail.com) .

**Vyacheslav Lyashenko** was born at Ukraine. In 1985 – graduated from the Kharkov Institute of Radio Electronics with a degree in "Applied Mathematics". He is a senior researcher of Kharkov National University of Radio Electronics and Head of the Laboratory "Transmission of Information Technology Systems for Disaster Risk Reduction". He has over 100 publications, including 4 Monographs so far. E-mail: [lyashenko.vyacheslav@gmail.com](mailto:lyashenko.vyacheslav@gmail.com) .

**Surya Prasath** is an assistant professor in the Division of Biomedical Informatics at the Cincinnati Children's Hospital Medical Center, and at the Biomedical Informatics, Electrical Engineering and Computer Science departments, University of Cincinnati, USA. He received his PhD in Mathematics from the Indian Institute of Technology Madras, India in 2009 (defended in March 2010). He has been a postdoctoral fellow at the Mathematics Department, University of Coimbra, Portugal, for two years. From 2012 to 2017 he was with the Computational Imaging and VisAnalysis (CIVA) Lab at the University of Missouri, USA and worked on various mathematical image processing and computer vision problems. He had summer fellowships/visits at Kitware Inc. NY, USA, The Fields Institute, Canada, and Institute for Pure and Applied Mathematics (IPAM), University of California Los Angeles, USA. His main research interests include nonlinear PDEs, regularization methods, inverse & ill-posed problems, variational, PDE based image processing, computer vision with applications in remote sensing, biometrics, and biomedical imaging domains.

---

*Code of State Categories Scientific and Technical Information (in Russian – GRNTI)): 29.31.15, 29.33.43, 20.53.23.*

*Received September 2, 2018. The final version – April 19, 2019.*

---