Analysis of Edge Detection on Compressed Images with Different Complexities

Vladimir Maksimović[1], Branimir Jakšić[1], Mile Petrović[1], Petar Spalević[1], Mirko Milošević[2]

[1] Faculty of Technical Sciences, University of Pristina in Kosovska Mitrovica, Knjaza Milosa 7, 38220 Kosovska Mitrovica, Serbia,
vladimir.maksimovic@pr.ac.rs, branimir.jaksic@pr.ac.rs, mile.petrovic@pr.ac.rs, petar.spalevic@pr.ac.rs, mirko.milosevic@viser.edu.rs

[2]School of Electrical and Computer Engineering of Applied Studies, Vojvode Stepe 283, 11000 Belgrade, Serbia

Abstract: This paper provides edge detection analysis on images, which consist of different numbers of details (small, medium and high number of details) and which are compressed by different compression algorithms - JPEG, JPEG2000 and SPIHT. Images from the BSD (Berkeley Segmentation Database) database were used and compressed with different number of bits per pixel. The analysis was performed for five edge detectors: Canny, LoG, Sobel, Prewitt, and Roberts. The fidelity of the detected edges was determined using the objective measures Figure of Merit (FOM), F measure and Performance Ratio (PR), where the reference value was taken from the GroundTruth image. Based on the results presented in the tables, it can be concluded that edge detection behaves differently depending on the number of bits per pixel and applied compression algorithm, as well as, the number of details in the image. Roberts operator has been proven to be the best solution, when it is necessary to perform better edge detection over compressed images with small a number of details, but Canny shows better results for images with a high number of details.

Keywords: edge detection; compression; image processing; Figure of Merit (FOM); F measure; Performance Ratio (PR); image complexity; bit per pixel (BPP)

1 Introduction

In today's multimedia systems, it is almost impossible to find a system that does not use image, video or audio compression. However, the development of technology has also brought an increasing use and processing of images, from use in daily life to those more serious professional uses such as image analysis in medicine, sensor networks, smart and security systems, television and so on. An uncompressed image requires more storage space for storage and processing, as well as, transmission via telecommunication channels. Considering this fact, there is a great deal of interest among researchers regarding image processing and compression. The size of the image can be large so that it is very impractical to store or transfer, especially when it comes to real-time image processing systems. For this reason, many image compression methods have been developed, but we can divide them all into lossy and lossless ones [1-3].

Depending on the need, various compression techniques and compression algorithms are applied, and as a result, the most popular are JPEG and JPEG2000. JPEG standard compression is based on Discrete Cosine Transform (DCT), while JPEG2000 compression is based on Discrete Wavelet Transform (DWT) [4-9]. Also, the compression algorithm based on Embedded Zero Tree Wavelet (EZW) is the SPIHT algorithm [10-12]. As mentioned, compression and coding techniques are used in many systems, i.e. where image processing is performed, so there are techniques for medical images [13-17], radar images [18-20], satellite images [21-23] and for many other smart systems combining different compression and coding techniques [24-26]. Image processing is an integral part of machine learning and artificial intelligence, where there are classifiers and neural network models that can be used as in [27], [28]. Also, the mathematical models presented in [29], [30] provide ideas for improving the algorithms for estimating image complexity used in this paper.

We are also witnessing an increase in the use of smart networks, the use of artificial intelligence to analyze, collect and process data. Such systems are mainly based on image processing and data processing, where the main processes are the extraction of a particular object from the scene, where edge detection and segmentation play an important role [31-34]. However, all of this gain particular weight and interest with the emergence and implementation of such systems on devices like Raspberry Pi and Arduino, which very often use real-time image processing, object detection and segmentation [35-39]. Many techniques and enhancements have been proposed to maximize the quality of edge detection and segmentation [39-43]. Given that the resolutions and image quality are increasing, thus occupying a large storage space, it is important to do compression so as not to impair the quality. Compression will affect edge detection, as examined in [44] using a wavelet transform, which underlies some compression algorithms, as well as facial recognition [45-47]. Therefore, the effect of compression on edge detection is presented in [44] where the authors examined only the influence of wavelet-based compression. The authors in [45-47] examined the effect of compression on face recognition using the JPEG and JPEG2000 algorithms, while the effect on edge detection was not examined. In this paper, the idea is to examine the impact of compression on edge detection using the most common compression algorithms.

The rest of the paper is divided as follows: Section 2 explains the system model, that is, the basic setting on which a detailed edge detection analysis was made. The images that were used for analysis are given, followed by tabulated values obtained during compression using different algorithms. In this section can be seen the method used by the authors to perform the analysis. Section 3 presents the obtained results of edge detection for five edge detectors over compressed images using different compression algorithms. The tables show three objective measures, and based on the results the discussion was made. In Section 3, there are sub sections for each operator. Finally, the conclusion of this paper is given, as well as the direction of future research.

2 System Model

This paper analysis the impact of JPEG, JPEG2000, and SPIHT algorithm on edge detection, where images are compressed with different number of bits per pixel (BPP), namely: 0.1, 0.3, 0.5, 1, 1.5, and 3 BPP. Images from the BSD. Used images are from the BSD database with the corresponding GroundTruth [48]. The images were selected to meet the three complexity criteria of small, medium and high complexity [49], that is, each image consists of a different level of detail: small, medium, and high level of details [49]. Table 1 shows the obtained values on the basis of which are the selected images from the database BSD, which meet the defined criteria.

Complexity in an image shows information about how much details exists in that image, and this can be observed for both static images and video formats. The simplest way of determining complexity is on the basis of observer’s visual assessment. However, it is not an objective measure to confirm the credibility of that assessment [50-52]. Since this paper looks at the effect of compression on edge detection, there are also methods that measure image complexity based on compression and thus make a link between compression, quality and complexity. One way of doing this is shown in [50]. JPEG, JPEG200 and SPIHT algorithms are based on the DCT and DWT techniques, so the number of details was calculated by making DCT and DWT on the high-frequency components (details), which are divided into four quadrants, along both directions (x and y). After that, the mean absolute value of the amplitude of the components belonging to the quadrants is calculated according to [49]: DCT in quadrant 1 (DCTD); DCT in quadrants 2 and 3 (DCTM); DWT in quadrant 1 (DWT); DWT in quadrants 2 and 3 (DWTM).

Edge detection and analysis were performed on the selected and compressed images for five edge detectors, namely: Canny, LoG, Sobel, Prewitt, and Roberts. Gradient and Laplace edge detection algorithms were written in Matlab, while image compression was performed using VcDemo. So, first, the images extracted from the BSD database with the corresponding GroundTruth were selected to satisfy the criteria in [49] using the technique from that paper. After that, the images were compressed in VcDemo using JPEG, JPEG2000 and SPIHT algorithm with different BPP. In the end, edge detection over compressed images was performed using five operators and objective measures are calculated in Matlab.

Table 1
Complexity criteria

	Images	DCTD	DCTM	WVTD	WVTM
Criterion L	**#238011**	**<2**	**<3.5**	**<0.8**	**<1.2**
		0.75	1.69	0.17	0.44
Criterion M	**#245051**	**3-4**	**4.5-6.5**	**1.4-1.8**	**2-2.8**
		3.11	7.02	1.12	2.09
Criterion H	**#231015**	**>4.9**	**>9**	**>1.9**	**>3.9**
		5.48	10.97	2.14	7.29

The authors have created a repository [53] containing used images for analysis, obtained images and codes.

Objective measures that were used are:

F measure (F1 score) which ranges from 0 to 100 and can be calculated [54]:

$$F = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}\,call}{\mathrm{Pr}\,ecision \times \mathrm{Re}\,call} \times 100 \quad (1)$$

F is within the limits of 0 ≤ F ≤ 1, ideally, F is equal to 1. The precision, also known as the positive predictive value is calculated [31]:

$$\mathrm{Pr}\,ecision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \quad (2)$$

while Sensitivity (Recall):

$$Sensitivity = \frac{TruePositive(TP)}{TruePositive(TP) + FalseNegative(FN)} \quad (3)$$

Where is TP - True Positive, pixels correctly segmented as foreground; FP - False Positive, pixels falsely segmented as foreground; TN - True Negative, pixels correctly detected as background and FN - False Negative, pixels falsely detected as background. Figure of Merit (FoM) which also ranges from 0 to 100, respectively represents the percentage value and can be calculated [55]:

$$FoM = \frac{1}{\max\{I_d, I_i\}} \sum_{k=1}^{I_d} \frac{1}{1 + \delta e^2(k)} \times 100 \quad (4)$$

where Id is the number of points on the detected edge, and Ii is the number of points on the ideal edge, represents the distance between the detected edge and the ideal edge, and is scaling constant and is usually 1/9.

Performance Ratio (PR) which ranges from 0 to infinite [56]:

$$PR = \frac{TrueEdge(EdgePixelsIdentifiedAsEdges)}{FalseEdges(NonEdgePixelsIdentifiedAsEdges) + (EdgePixelsIdentifiedAsNon - EdgePixels)} \times 100 \quad (5)$$

Table 2 shows the Peak Signal to Noise ratio (PSNR) [49] values which show how the number of bits per pixel (BPP) is affecting image compression. It can be seen from Table 2 that the increase in BPP contributes significantly to image quality, especially with JPEG compression, when the number of image details is small. JPEG2000 and SPIHT obtained similar results, but the number of details noticeably affects the compression.

Table 2
PSNR values for three compression algorithms with different BPP and level of details

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	28.2	40.3	52.7	62.2	61.8	61.8
	JPEG2000	36.7	42.5	46.4	52.4	52.4	52.4
	SPIHT	36.6	42.7	46.6	53.5	58.5	70.7
MD	**JPEG**	20.7	26.5	28.9	31.7	41.5	56.4
	JPEG2000	24.3	28.6	31.6	37.5	42.1	50.8
	SPIHT	24.0	28.4	31.7	37.5	42.1	53.9
HD	**JPEG**	19.1	22.8	24.7	27.6	28.8	50.4
	JPEG2000	21.4	24.2	26.3	30.3	33.7	43.6
	SPIHT	21.3	24.2	26.1	30.1	33.6	43.5

Fig. 1, Fig. 2, and Fig. 3 show a compressed image for a different number of BPP and small number of details (SD) when using JPEG, JPEG2000, and SPIHT compression, respectively. In Fig. 4, Fig. 5 and Fig. 6, images with medium level of details compressed with JPEG, JPEG2000, and SPIHT compression and different BPP are shown, respectively. When it comes to a high number of details in an image (HD), using the JPEG, JPEG2000 and SPIHT algorithm, the resulting compressed images for different BPP are shown in Fig. 7, Fig. 8 and Fig. 9, respectively.

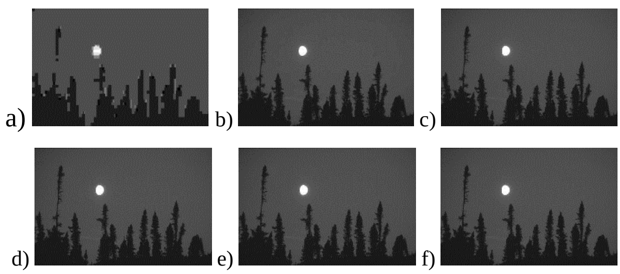


Figure 1
SD image with JPEG compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

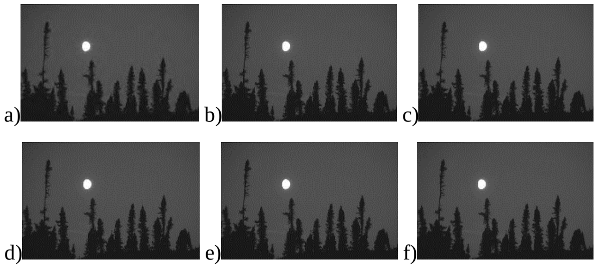


Figure 2

SD image with JPEG2000 compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

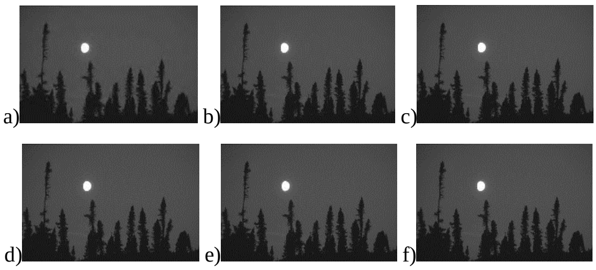


Figure 3

SD image with SPIHT compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

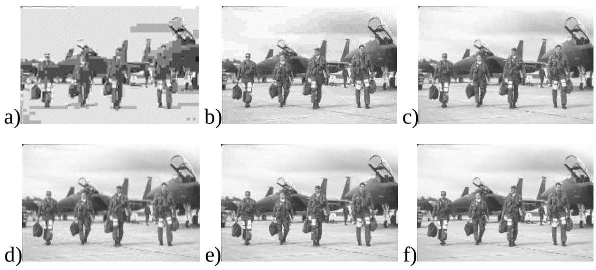


Figure 4

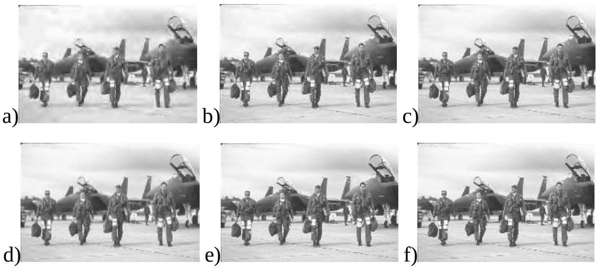
MD image with JPEG compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

From the shown figures can be seen that the quality is usable by applying all kinds of compression for all levels of detail in the image. However, degradation is greatest with a high number of details and at lower BPP, which is confirmed by results in Table 1.

d) e) f)

Figure 5

MD image with JPEG2000 compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

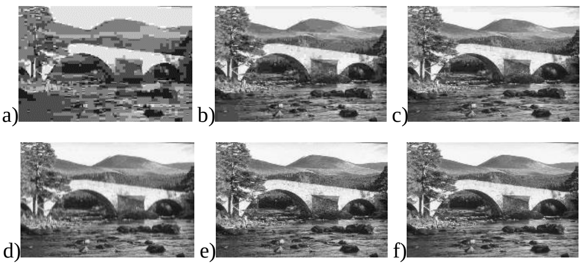


a) b) c)

d) e) f)

Figure 6

MD image with SPIHT compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

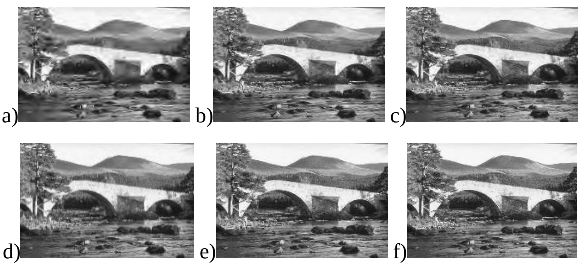


a) b) c)

d) e) f)

Figure 7

HD image with JPEG compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3



a) b) c)

d) e) f)

Figure 8

HD image with JPEG2000 compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

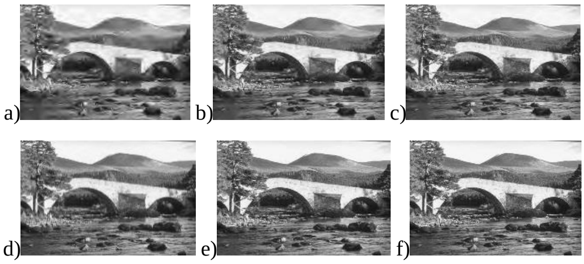


Figure 9

HD image with SPIHT compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

Thus, lower BPP, compression algorithm and number of details significantly affect image quality. However, the main aim of this paper is to examine the impact of edge detection over these images, i.e. how much all of this affects the quality of the detected edge. Table 3 shows the F, FOM and PR values obtained by applying five edge detection operators over images with different level of details. Based on these results, it can be seen that the best values are obtained when the number of details in the image is small. The Roberts operator obtained the best values and the LoG the worst when the number of details in the image is small and medium, while at a high number of details, Canny obtained higher values than the others. In order to present these results visually, Fig. 10, Fig. 11 and Fig. 12 show an image with small, medium and high number of details over which edge detection was performed using five operators.

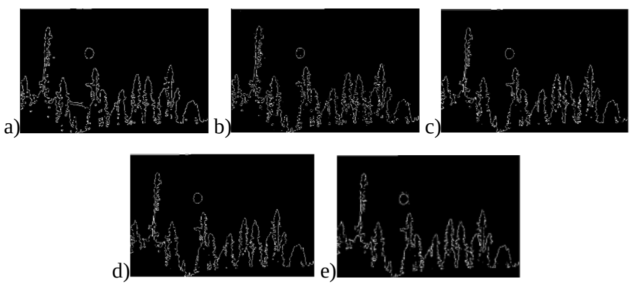
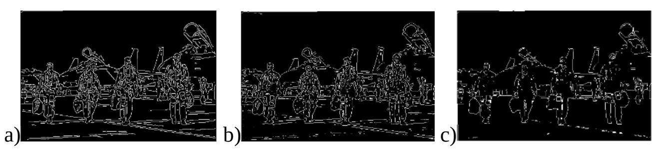


Figure 10

SD image: a) Canny, b) LoG, c) Prewitt, d) Sobel, e) Roberts



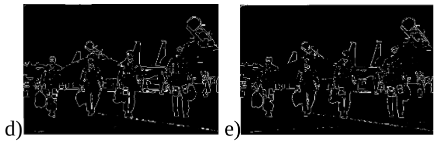


Figure 11

MD image: a) Canny, b) LoG, c) Prewitt, d) Sobel, e) Roberts

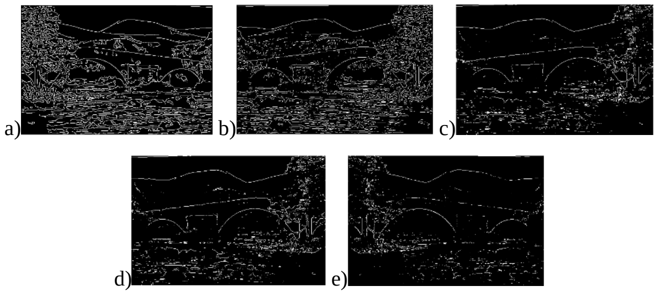


Figure 12

HD image: a) Canny, b) LoG, c) Prewitt, d) Sobel, e) Roberts

Table 3

F, FOM and PR values obtained by applying different edge detectors

	Operator	**F**	**FOM**	**PR**
SD	Canny	35.11	89.39	27.54
	LoG	32.40	90.07	24.38
	Prewitt	35.15	89.40	26.36
	Sobel	34.07	89.49	26.31
	Roberts	46.91	91.74	44.05
MD	Canny	20.99	46.59	13.28
	LoG	18.90	57.70	11.65
	Prewitt	23.73	80.98	15.56
	Sobel	23.80	81.01	15.62
	Roberts	35.24	80.17	27.21
HD	Canny	22.02	68.94	14.12
	LoG	19.05	80.83	11.76
	Prewitt	17.98	63.16	10.96
	Sobel	18.16	63.75	11.09
	Roberts	16.88	56.78	10.15

3 Results

The previous section showed how these results were obtained. In Section 3, the results are divided by edge detector, i.e. a sub-section is made for each detector to make the results more transparent. The results were obtained using the mathematical models defined in Section 2. The calculation is based on the theoretical models presented in [44-47]. The results are presented in tables and for each combination of parameters (compression and edge detector) can be found in the repository [53], as well as, full size images used code.

3.1 Canny Edge Detector

Table 4, Table 5 and Table 6 show the F, FOM and PR values, respectively, obtained by applying a Canny edge detector over images with different number of details compressed by different compression algorithms. Based on the obtained results, it can be seen that by increasing the number of bits per pixel, better values are obtained. The best values are obtained when the number of details in the image is small, while when the number of details in the image is medium and high obtained values are similar.

Table 4
F values obtained by using a Canny edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	25.55	32.91	35.47	35.52	35.49	35.49
	JPEG2000	28.19	34.42	35.32	35.55	35.55	35.55
	SPIHT	28.66	34.05	35.15	35.52	35.63	35.56
MD	**JPEG**	17.97	19.49	20.90	20.51	20.88	21.04
	JPEG2000	20.37	20.94	20.74	20.89	21.13	21.22
	SPIHT	20.63	20.94	21.18	21.19	21.36	21.96
HD	**JPEG**	20.45	20.44	21.94	21.97	21.90	22.00
	JPEG2000	19.24	21.26	21.29	22.08	22.09	22.98
	SPIHT	19.73	21.12	21.29	21.83	22.03	22.11

Table 5
FOM values obtained by using a Canny edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	75.03	89.19	89.39	89.39	89.36	89.36
	JPEG2000	86.75	89.30	89.39	89.36	89.36	89.36
	SPIHT	88.37	89.11	89.28	89.39	89.39	89.40
MD	**JPEG**	47.01	42.50	47.24	44.35	46.31	47.89
	JPEG2000	52.63	47.34	46.30	46.64	46.96	47.09
	SPIHT	52.21	47.51	49.91	49.97	51.11	51.79

HD	**JPEG**	58.16	65.58	67.46	67.08	66.95	69.95
	JPEG2000	64.25	73.27	73.83	78.63	78.55	78.32
	SPIHT	55.65	73.91	75.09	75.13	75.38	75.65

Table 6

PR values obtained by using a Canny edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	17.16	24.53	27.49	27.54	27.51	27.51
	JPEG2000	19.63	26.24	27.31	27.58	27.58	27.58
	SPIHT	20.08	25.82	27.10	27.55	27.68	27.59
MD	**JPEG**	10.95	12.10	13.21	12.90	13.20	13.32
	JPEG2000	12.79	13.24	13.08	13.21	13.39	13.71
	SPIHT	12.99	13.24	13.44	13.47	13.52	13.76
HD	**JPEG**	12.85	12.85	14.06	14.08	14.02	14.11
	JPEG2000	11.91	13.50	13.53	14.17	14.05	14.28
	SPIHT	12.29	13.38	13.52	13.96	14.13	14.21

3.2 LoG Edge Detector

Table 7, Table 8 and Table 9 show the F, FOM and PR values, respectively, obtained by applying LoG edge detectors over images with different number of details compressed by different compression algorithms. The LoG detector gave the best results when the number of details in the image is small. However, it can be seen that the values are lower when the number of details in the image is medium and high at BPP 0.1, but the values did not increase much by further increasing the BPP from 0.3 upwards.

Table 7

F values obtained by using a LoG edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	21.46	31.98	33.07	32.77	32.72	32.72
	JPEG2000	28.82	31.67	32.24	32.73	32.73	32.73
	SPIHT	29.10	31.89	32.50	32.77	32.77	32.84
MD	**JPEG**	16.22	17.99	18.45	18.55	18.83	18.89
	JPEG2000	18.61	18.38	18.54	18.61	18.84	18.98
	SPIHT	18.49	18.45	18.51	18.55	18.98	18.85
HD	**JPEG**	17.78	18.78	18.99	19.06	19.08	19.21
	JPEG2000	15.57	18.56	18.63	19.18	19.16	19.22
	SPIHT	16.48	18.26	18.70	18.86	18.96	19.04

Table 8
FOM values obtained by using a LoG edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	74.29	89.92	90.11	90.07	90.72	90.06
	JPEG2000	89.49	90.02	90.09	90.09	90.09	90.09
	SPIHT	89.25	90.01	90.02	90.09	90.08	90.11
MD	**JPEG**	53.66	55.71	56.54	56.20	57.44	57.72
	JPEG2000	57.94	59.15	59.40	57.18	57.38	59.55
	SPIHT	69.19	59.70	59.13	57.55	59.71	60.69
HD	**JPEG**	78.42	79.42	80.88	81.50	81.54	81.86
	JPEG2000	64.75	77.07	79.16	80.79	80.80	80.88
	SPIHT	64.04	76.20	79.13	80.10	80.64	80.82

Table 9
PR values obtained by using a LoG edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	13.66	25.51	24.71	24.37	24.31	24.31
	JPEG2000	20.25	23.18	23.79	24.33	24.33	24.33
	SPIHT	20.52	23.42	24.08	24.37	24.39	24.42
MD	**JPEG**	9.68	10.97	11.31	11.39	11.60	11.65
	JPEG2000	11.43	11.26	11.38	11.42	11.61	11.76
	SPIHT	11.34	11.31	11.36	11.36	11.71	11.91
HD	**JPEG**	10.81	11.56	11.72	11.77	11.78	11.92
	JPEG2000	9.22	11.40	11.45	11.87	11.77	11.94
	SPIHT	9.86	11.17	11.50	11.62	11.70	11.76

3.3 Sobel Edge Detector

Table 10, Table 11 and Table 12 give the F, FOM and PR values, respectively, obtained by applying a Sobel edge detector over images with different number of details compressed by different compression algorithms. Sobel detector gave good results when it comes to the small and medium number of details in the image. With high number of details, the results are poor at lower BPP.

Table 10
F values obtained by using a Sobel edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	20.18	33.95	34.40	34.29	34.34	34.34
	JPEG2000	31.49	33.84	34.18	34.33	34.33	34.33
	SPIHT	30.99	33.83	34.42	34.34	34.41	34.56
MD	**JPEG**	17.81	22.61	23.31	23.75	23.89	23.77

	JPEG2000	20.33	22.84	23.32	23.47	23.49	23.76
	SPIHT	20.25	22.80	23.36	23.65	23.62	23.89
HD	**JPEG**	11.62	15.85	17.15	17.62	17.79	18.07
	JPEG2000	12.88	17.28	17.65	18.00	17.98	18.18
	SPIHT	13.44	17.44	17.82	18.18	18.11	18.14

Table 11

FOM values obtained by using a Sobel edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	68.69	88.86	89.32	89.47	89.50	89.50
	JPEG2000	85.54	88.94	89.36	89.38	89.37	89.38
	SPIHT	84.90	88.98	89.33	89.40	89.44	89.56
MD	**JPEG**	68.37	77.33	79.56	81.89	80.88	81.01
	JPEG2000	67.29	77.80	79.80	80.68	80.89	80.98
	SPIHT	66.28	77.92	79.28	80.57	80.80	81.08
HD	**JPEG**	48.64	57.85	60.82	62.73	64.66	63.77
	JPEG2000	46.09	59.73	61.93	63.32	63.61	63.97
	SPIHT	46.98	60.32	61.46	63.03	63.37	63.77

Table 12

PR values obtained by using a Sobel edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	12.64	25.70	26.22	26.10	26.15	26.15
	JPEG2000	22.98	25.58	25.96	26.13	26.13	26.13
	SPIHT	22.45	25.56	26.25	26.15	26.27	26.29
MD	**JPEG**	10.83	14.61	15.20	15.58	15.70	15.59
	JPEG2000	12.76	14.80	15.20	15.33	15.35	15.58
	SPIHT	12.69	14.77	15.24	15.49	15.57	15.69
HD	**JPEG**	6.58	9.42	10.35	10.70	10.82	11.03
	JPEG2000	7.39	10.45	10.72	10.98	10.96	11.11
	SPIHT	7.77	10.56	10.84	11.11	11.06	11.09

3.4 Prewitt Edge Detector

Table 13, Table 14 and Table 15 show the F, FOM and PR values, respectively, obtained by applying a Prewitt edge detector over images with different number of details compressed by different compression algorithms. The Prewitt operator obtained well values with JPEG2000 and SPIHT compression in SD images even when the number of bits per pixel is low.

Table 13
F values obtained by using a Prewitt edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	18.52	33.93	34.54	34.49	34.62	34.62
	JPEG2000	31.71	34.10	31.40	34.50	34.50	34.52
	SPIHT	31.01	34.02	34.53	34.57	34.67	34.75
MD	**JPEG**	17.87	22.73	23.39	23.87	24.06	23.79
	JPEG2000	20.40	22.91	23.34	23.66	23.65	24.02
	SPIHT	20.37	22.91	23.34	23.64	23.67	23.74
HD	**JPEG**	11.72	15.87	17.15	17.78	17.98	18.05
	JPEG2000	12.92	17.39	17.97	18.05	18.04	18.15
	SPIHT	13.47	17.62	17.98	18.06	18.15	18.33

Table 14
FOM values obtained by using a Prewitt edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	68.12	88.72	89.30	89.42	89.46	89.47
	JPEG2000	85.62	89.09	89.29	89.36	89.36	89.36
	SPIHT	85.02	88.82	89.22	89.31	89.40	89.44
MD	**JPEG**	68.08	77.44	79361	81.68	80.80	81.03
	JPEG2000	67.51	77.78	79.68	80.69	81.05	81.79
	SPIHT	66.56	77.94	79.18	80.66	80.72	80.86
HD	**JPEG**	48.76	57.13	60.49	62.61	64.00	64.13
	JPEG2000	46.25	59.53	61.77	63.14	62.97	63.23
	SPIHT	47.20	60.28	61.51	62.89	62.71	63.25

Table 15
PR values obtained by using a Prewitt edge detector

	BPP	0.1	0.3	0.5	1	1.5	3
SD	**JPEG**	11.36	25.68	26.38	26.32	26.48	26.50
	JPEG2000	23.21	25.87	26.22	26.33	26.33	26.33
	SPIHT	22.47	25.78	26.31	26.42	26.55	26.59
MD	**JPEG**	10.88	14.71	15.27	15.68	15.84	15.60
	JPEG2000	12.82	14.86	15.22	15.50	15.49	15.88
	SPIHT	12.79	14.86	15.30	15.48	15.51	15.57
HD	**JPEG**	6.63	9.43	10.35	10.81	10.96	11.04
	JPEG2000	7.42	10.53	10.95	11.01	11.00	11.19
	SPIHT	7.79	10.69	10.96	11.02	11.09	11.00

3.5 Roberts Edge Detector

Table 16, Table 17 and Table 18 show the F, FOM and PR values, respectively, obtained by applying Roberts edge detectors over images with different number of details compressed by different compression algorithms. When it comes to the small number of details in an image, also and mostly when the number of details in an image is medium, the Roberts operator obtained the best results using JPEG2000 and SPIHT compression.

Table 16

F values obtained by using a Roberts edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	20.64	43.98	46.85	46.64	46.69	46.69
	JPEG2000	38.21	45.22	46.86	46.55	46.55	46.61
	SPIHT	37.63	45.90	46.70	46.60	46.73	46.69
MD	**JPEG**	18.42	31.08	32.71	33.95	35.03	35.24
	JPEG2000	24.46	31.33	25.83	34.94	35.25	35.78
	SPIHT	24.51	31.36	34.32	35.01	35.25	35.42
HD	**JPEG**	12.78	14.06	14.81	15.11	14.55	17.05
	JPEG2000	11.03	14.16	15.77	17.48	17.74	17.98
	SPIHT	13.47	17.62	17.98	18.06	18.15	18.33

Table 17

FOM values obtained by using a Roberts edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	69.50	90.55	91.62	91.69	91.73	91.73
	JPEG2000	82.01	90.38	91.59	91.66	91.66	91.66
	SPIHT	80.93	90.79	91.50	91.72	91.72	91.66
MD	**JPEG**	66.86	75.36	76.53	80.48	80.44	81.11
	JPEG2000	58.17	72.75	79.08	80.21	80.14	81.01
	SPIHT	56.63	71.30	78.51	79.89	80.17	80.24
HD	**JPEG**	47.90	52.59	52.61	53.27	54.00	56.67
	JPEG2000	32.82	45.35	52.80	57.17	58.30	59.94
	SPIHT	47.20	60.28	61.51	62.89	62.71	63.25

Table 18

PR values obtained by using a Roberts edge detector

	BPP	**0.1**	**0.3**	**0.5**	**1**	**1.5**	**3**
SD	**JPEG**	13.01	39.55	44.07	43.70	43.79	43.79
	JPEG2000	30.92	41.27	44.09	43.54	43.54	43.54
	SPIHT	30.16	42.42	43.81	43.64	43.76	43.61
MD	**JPEG**	11.29	22.55	24.31	25.70	26.96	27.21

	JPEG2000	16.19	22.81	25.83	26.85	27.22	27.46
	SPIHT	16.23	22.84	26.13	26.94	27.22	27.29
HD	**JPEG**	7.33	8.18	8.69	8.90	8.51	10.27
	JPEG2000	6.20	8.25	9.36	10.59	10.79	10.88
	SPIHT	7.79	10.69	10.96	11.02	11.09	11.11

In Section 2 it could be seen visually and objectively how compression and different BPP values effect on quality of images. While in Section 3, edge detection was performed on these images, and based on the results presented in tables by all operators, the effect of compression on edge detection can be seen. BPP effect on edge detection as well as the number of details in an image. The Canny operator has proven to be a good solution but when the number of details in the image is small or medium, the Roberts operator finds its application. For this reason, Figure 13 shows the edge detection using the Roberts operator. Detection is shown for an image that is compressed with JPEG technique at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3.

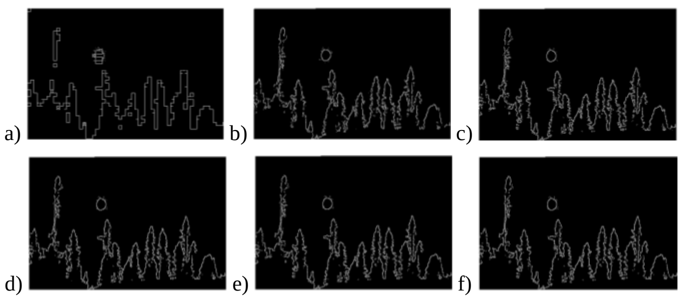


Figure 13
Roberts detection for SD image with JPEG compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

Since the Canny operator proved to be a very good solution when the number of details in the image is high even at lower BPP values, Figure 14 shows the edge detection using this operator. Detection is shown for an image that is compressed with JPEG technique at BPP a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3.

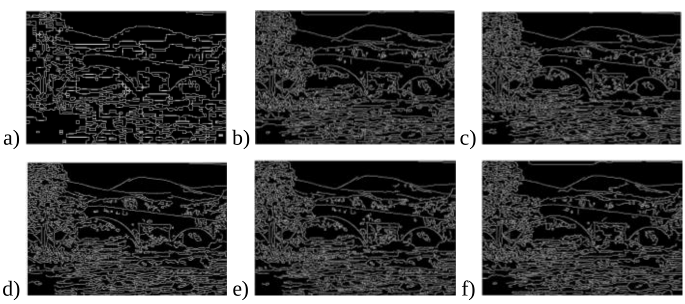


Figure 14
Canny detection for HD image with JPEG compression at BPP: a) 0.1, b) 0.3, c) 0.5, d) 1, e) 1.5, f) 3

Thus, Figure 13 and Figure 14 are only a visual representation of the results obtained in Section 3. All images for the results obtained can be found in the repository [53].

Conclusions

This paper analyzes gradient (Sobel, Prewitt, Robert), Laplacian of Gaussian and Canny operator. Operators were applied to images which consist different number of details (small, medium and high) and compressed by JPEG, JPEG2000 and SPIHT compression algorithms at different bits per pixel. Objective measures were used - F measure, FOM and PR and the results are tabulated.

Based on the obtained results, it can be concluded that when the number of details in the image is small and medium and when using JPEG compression, the best results are obtained using the Roberts operator, only at a BPP of 0.1 Canny achieved better values. Other operators have similar values. With the same amount of detail in image, JPEG2000 and SPIHT compression achieve better results than JPEG, which is reflected in edge detection. Also, the Roberts operator obtained the best results over the other operators, however, the values are similar to JPEG compression except when the BPP is low. Using high detail images and JPEG, JPEG2000 and SPIHT compression, the best edge detection was obtained with the Canny operator. Edge detection is better when JPEG2000 and SPIHT are used.

The results obtained in this paper contribute to the further development of image compression algorithms to be more suitable for use in systems where image processing such as segmentation and edge detection is used. However, it provides an incentive to optimize edge detectors for image compression at lower bits per pixel values, with consideration of the complexity of the image.

References

[1] A. J. Hussain, A. Al-Fayadh and N. Radi: Image compression techniques, A survey in lossless and lossy algorithms, Neurocomputing, 2018, Vol. 300, pp. 44-69

[2] D. Ravichandran, R. Nimmatoori and M. G. Ahamad: A Study on State-of-the-Art Image Compression Algorithms and Standards, International Journal On Advanced Computer Theory And Engineering (IJACTE), 2016, Vol. 5, Iss. 3, pp. 2319-2523

[3] K. V. Gomathi and R. Lotus: Digital image compression techniques, IJRET: International Journal of Research in Engineering and Technology, 2014, Vol. 03 Iss. 10, pp. 285-290

[4] G. K. Wallace: The JPEG Still Picture Compression Standard, Comm. of the ACM, 1991, Vol. 34, No. 4, pp. 30-44

[5] C. Christopoulos, A. Skodras and T. Ebrahimi: The JPEG2000 Still Image Coding System: An Overview, IEEE Trans. on Consumer Electronics, November 2000, Vol. 46, No. 4, pp. 1103-1127

[6] R. C. Gonzalez, R. E. Woods and S. L. Eddins: Digital Image Processing Using MATLAB. Pearson Prentice Hall, 2004

[7] D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos: JPEG 2000 Still Image Coding Versus other Standards, ISO/IEC JTC1/SC29/WG1 (ITU-T SG8), July 2000, http://www.jpeg.org/public/wg1n1816.pdf

[8] J. Kumar and M. Kumar: Comparison of image compression methods on various images, International Conference on Advances in Computer Engineering and Applications, Ghaziabad, 2015, pp. 114-118

[9] R. Singh, and V. K. Srivastava: JPEG2000: A review and its performance comparison with JPEG. Seconnd International Conference on Power, Control and Embedded Systems, Allahabad, India, 2012

[10] P. L. Dragotti, G. Poggi and A. Ragozini: Compression of multispectral images by three-dimensional SPIHT algorithm, IEEE Trans. on Geosciences and Remote Sensing, January 2000, Vol. 38, No. 1, pp. 416-428

[11] B. I. Kochi, and B. B. S. Kumar: EZW and SPIHT Algorithms for Image Compression and Denoising, ITSI Transactions on Electrical and Electronics Engineering (ITSI-TEEE), 2016, Vol. 4, No. 2, pp. 2320-8945

[12] C. Kaur and S. Budhiraja: Improvements of SPIHT in Image Compression-Survey, International Journal of Emerging Technology and Advanced Engineering, January 2013, Vol. 3, Iss. 1, pp. 652-656

[13] Sharon M. Perlmutter, Pamela C. Cosman, Chien-Wen Tseng, Richard A. Olshen, Robert M. Gray, King C. P. Li and Colleen J. Bergin: Medical Image Compression and Vector Quantization, Statistical Science, 1998, Vol. 13, No. 1, pp. 30-53

[14] D. A. Koff and H. Shulman: An Overview of Digital Compression of medical Images: Can We Use Lossy Image Compression in Radiology, Radiol J., Oct. 2006, Vol. 57, No. 4, pp. 211-217

[15] M. R. Ashwin Dhivakar, M. G. Ahamad and D. Ravichandran: Medical image compression using Embedded Zerotree Wavelet (EZW) coder, International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, Indina, 2016, pp. 17-23

[16] S. Suma and V. Sridhar: A Review of the Effective Techniques of Compression in Medical Image Processing, International Journal of Computer Applications, 2016, Vol. 97, pp. 23-30

[17] T. H. Oh and R. Besar: Medical image compression using jpeg-2000 and jpeg: a comparison study, Journal of Mechanics in Medicine and Biology, 2002, Vol. 2, No. 3 & 4, pp. 313-328

[18] Y. Li: Synthetic Aperture Radar (SAR) Image Compression Using the Wavelet Transform, Faculty of Engineering and Applied Sciences Mernorial University of Newfoundland August, 1997

[19] A. T. Chien, K. S. Miettinen, A. Lan amd M. A. Lepley: Applications of Digital Image Processing XXIII, 2000, doi:10.1117/12.411604

[20] U. Pestel-Schiller: Coding of SAR image data for data compression, in 10th European Conference on Synthetic Aperture Radar (EUSAR), Berlin, Germany, Jun. 2014

[21] K. Sahnoun and B. Noureddine: Satellite Image Compression Algorithm Based on the FFT, The International journal of Multimedia & Its Applications, 2016, Vol. 6, No. 1, pp. 77-83

[22] J. G. Kumar and A. Singh: Fractal Compression of Satellite Images, J. Indian Soc. Remote Sens. (December 2008) Vol. 36, pp. 299-311

[23] Z. Liang, T. Xinming, Z. Guo and W. Xiaoliang: Effects of jpeg2000 and spiht compression on image classification, In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Conference, Beijing, China, August 2008

[24] B. Emmanuel, M. Mu'Azu, S. Sani and S. Garba: A Review of Wavelet-Based Image Processing Methods for Fingerprint Compression in Biometric Application, British Journal of Mathematics & Computer Science, 2014, Vol. 4, pp. 2781-2798

[25] A. J. Hussain, D. Al-Jumeily, N. Radi and P. Lisboa: Hybrid neural network predictive-wavelet image compression system, Neurocomputing, 2014, Vol. 151, pp. 975-984

[26] R. Jumar, H. Maaß, V. Hagenmeyer: Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis, Computers & Electrical Engineering, Vol. 71, October 2018, pp. 465-476

[27] M. Trojanová, and A. Hošovský: Comparison of Different Neural Networks Models for Identification of Manipulator Arm Driven by Fluidic Muscles, Acta Polytechnica Hungarica, 2018, Vol. 15, No. 7, pp. 7-28

[28] R. Zall, M. R. Kangavari: On the construction of multi-relational classifier based on canonical correlation analysis, International Journal of Artificial Intelligence, 2019, Vol. 17, No. 2, pp. 23-43

[29] J. C. Spall: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Transactions on Automatic Control, 1992, Vol. 37, No. 3, pp. 332-341

[30] C. Pozna, R. E. Precup, J. Tar, I. Škrjanc, S. Preitl: New results in modelling derived from Bayesian filtering, Knowledge-Based Systems, 2010, Vol. 23, No. 2, pp. 182-194

[31] S. I. Jabbar, C. R. Day, N. Heinz and E. K. Chadwick: Using Convolutional Neural Network for edge detection in musculoskeletal ultrasound images, 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, Canada, 2016, pp. 4619-4626

[32] M. Hagara and P. Kubinec, About Edge Detection in Digital Images, Radioengineering, 2018, Vol. 27, No. 4, pp. 919-929

[33] E. Moya-Albor, H. Ponce and Jorge Brieva: An Edge Detection Method using a Fuzzy Ensem- ble Approach, Acta Polytechnica Hungarica, 2017, Vol. 14, No. 3, pp. 149-168

[34] K. B. Krishnan, S. P. Ranga and N. Guptha: Survey on Different Edge Detection Techniques for Image Segmentation, Indian Journal of Science and Technology, 2017, Vol. 10, No. 4

[35] N. A. Othman, M. U. Salur, M. Karakose and I. Aydin: An Embedded Real-Time Object Detection and Measurement of its Size, International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-4

[36] A. Rupani, P. Whig, G. Sujediya and P. Vyas: A robust technique for image processing based on interfacing of Raspberry-Pi and FPGA using IoT, International Conference on Computer, Communications and Electronics (Comptelix), Jaipur, India, 2017, pp. 350-353

[37] T. Mustafa: Object detection and tracking for real time field survelliance applications, January 2017, Ankara, Turkey

[38] M. S. Munna, B. K. Tarafder, M. G. Robbani and T. C. Mallick: Design and implementation of a drawbot using Matlab and Arduino Mega, International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 769-773

[39] P. Arunkumar, S. Shantharajah, and M. Geetha: Improved canny detection algorithm for processing and segmenting text from the images, Cluster Computing, 2018, Vol. 22, pp. 7015-7021

[40] S. Biswas, D. Ghoshal and R. Hazra: A new algorithm of Image Segmentation using Curve Fitting Based Higher Order Polynomial Smoothing, Optik, 2016, Vol. 127, No. 20, pp. 8916-8925

[41] S. Biswas and R. Hazra: Robust edge detection based on Modified Moore-Neighbor, Optik, 2018, Vol. 168, pp. 931-943

[42] S. Sergyán: A New Approach of Face Detection-based Classification of Image Databases, Acta Polytechnica Hungarica, 2009, Vol. 6, No. 1, pp. 175-184

[43] B. Kuljić, J. Simon and T. Szakáll: Pathfinding Based on Edge Detection and Infrared Distance Measuring Sensor, Acta Polytechnica Hungarica, 2009, Vol. 6, No. 1, pp. 103-116

[44] V. Maksimovic, P. Lekic, M. Petrovic, B. Jaksic and P. Spalevic: Experimental analysis of wavelet decomposition on edge detection, Proceedings of the Estonian Academy of Sciences, 2019, Vol. 68, No. 3, pp. 284-298

[45] K. Wat and S.H. Srinivasan: Effect of Compression on Face Recognition, Proceedings of the fifth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, 21-23 April 2004, Lisboa, Portugal

[46] H. Moon and P. J. Phillips: Computational and Performance Aspects of PCA-based Face recognition Algorithms, Perception, 2001, Vol. 30, pp. 303-321

[47] Delac K., Grgic M. and S. Grgic: Effects of JPEG and JPEG2000 Compression on Face Recognition, Pattern Recognition and Image Analysis, 2005, Vol. 3687, pp. 136-145

[48] The Berkeley Segmentation Dataset and Benchmark, https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/, Acessed: 14.07.2019

[49] S. Ilic, M. Petrovic, B. Jaksic, P. Spalevic, Lj. Lazic and M. Milosevic: Experimental analysis of picture quality after compression by different methods. Przegląd Elektrotechniczny, 2013, Vol. 89, No. 11, pp. 190-194

[50] H. Yu and S. Winkler: Image complexity and spatial information. Proceedings of fifth International Workshop on Quality of Multimedia Experience (QoMEX), Klagenfurt am Worthersee, Austria, 2013, pp. 12-17

[51] H. Wu, C. Mark, and K. Robert: A study of video motion and scene complexity. Tech. Rep. WPI-CS-TR-06-19, 2006, Worcester Polytechnic Institute

[52] V. Chikhman, V. Bondarko, M. Danilova, A. Goluzina and Y. Shelepin: Complexity of images: Experimental and computational estimates compared. Perception, 2012, Vol. 41, No. 6, pp. 631-47

[53] Dataset repository: https://drive.google.com/open?id= 1s6MkarMqT_au0JOsTUODCNOxgWfIYZut, Accesed: 28.01.2020

[54] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik: Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, Vol. 33, No. 5, pp. 898-916

[55] W. K. Pratt: Digital Image Processing. 4th ed., Hoboken, New Jersey (USA): John Wiley & Sons, 2007, ISBN: 9780471767770

[56] P. A. Khaire, N. V. Thakur: A Fuzzy Set Approach for Edge Detection. International Journal of Image Processing, 2012, Vol. 6, No. 6, pp. 403-412