JAISCR, 2020, Vol. 10, No. 3, pp. 223 – 237
10.2478/jaiscr-2020-0015

DETECTING VISUAL OBJECTS BY EDGE CRAWLING

Rafał Grycuk[1,*], Adam Wojciechowski[2], Wei Wei[3], Agnieszka Siwocha[4]

[1] *Czestochowa University of Technology,*
al. Armii Krajowej 36, 42-200 Częstochowa, Poland

[2] *Institute of Information Technology,*
Lodz University of Technology, Lodz, Poland

[3] *School of Computer Science and Engineering,*
Xi'an University of Technology, China

[4] *Information Technology Institute,*
University of Social Sciences, 90-113 Łodz
Clark University Worcester, MA 01610, USA

[*] *E-mail: rafal.grycuk@pcz.pl*

Submitted: 17th July 2019; Accepted: 11th February 2020

Abstract

Content-based image retrieval methods develop rapidly with a growing scale of image repositories. They are usually based on comparing and indexing some image features. We developed a new algorithm for finding objects in images by traversing their edges. Moreover, we describe the objects by histograms of local features and angles. We use such a description to retrieve similar images fast. We performed extensive experiments on three established image datasets proving the effectiveness of the proposed method.

Keywords: content-based image retrieval, crawler, edge detection, image descriptor, object extraction

1 Introduction

The primary cause of human vision superiority over computers is many years of evolution and adaptation. The interpretation of the human vision process is hard to characterize; thus, even harder to recreate it using computers. Let us take, for instance, a task to find a “beautiful landscape”. A mathematical description of “beautiful” is very challenging, although the definition of beauty for every person is easily definable, yet subjective. In daily life, people use vision at work, leisure time and many other activities. Though this process is simple for us, its reproduction by computers is not trivial.

Computer vision methods are used in many various systems. In most cases, the machines (computers) replace humans in pattern recognition tasks such as: controlling, safety, forensics, medicine, and many others. The cause of such replacement is dictated by the greater efficiency of computers. The machine does not require sleep and is not dependent on other factors, such as anger, fear, laziness, boredom and many others. Nevertheless, currently, the complete replacement is not possible due to the difference in perception.

Recent years brought a vast spread of imaging devices and the abundance of images on the Internet. Thus, there is a need for tools that can rapidly describe images and search for them based on their content in different areas of life, science, and engineering. Smeulders et al. [37] conclude that the significant problems in image description are

- Sensory gap - there is a wide gap between a real-world object and computer ability to sense and describe this object. An infinite number of different information can be produced by the same object (e.g. shape, orientation, texture, colour, and many others). Moreover, two completely different objects can produce a similar description. The human perceptual senses allow to quickly compare various types of information and distinguish (classify) the objects even if there is only an object fragment available. Unfortunately, the current computer sensors cannot overcome this issue.
- Semantic gap - there is a semantic difference between high-level image analysis performed by humans and low-level (mathematical descriptors) performed by computers. The human brain recognizes the objects by understanding it, but computer analysis operates on descriptors and features. It is hard to create a visual object representation in the form which will allow understanding what the object "means".

To better understand these issues, let us take for considerations the image with several objects presented in Figure 1.

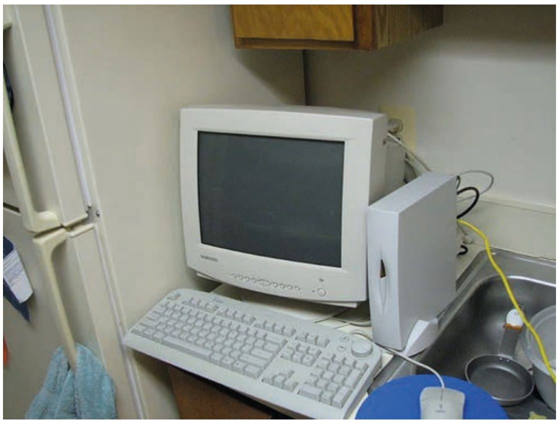

Figure 1. Example image taken from PASCAL VOC dataset.

In the foreground, we have a desktop computer set, which consists of at least four main objects (a monitor, a computer, a mouse and a keyboard). In the background, we see several objects, such as a refrigerator (left side of the image) on which towels hang, a cupboard (image top), a frying pan and a bowl fragment (right side). For humans, this image is completely understandable, and objects are easily recognizable. Even when the human sees only the object fragment, he can positively identify the object. Our sight provides millions of signals to the brain (most of them are aggregated and generalized), and they are extensively analyzed based on previously obtained knowledge. Thanks to this process, humans can recognize objects even if only a fragment is visible (e.g. refrigerator or bowl in Figure 1). Unfortunately, currently available sensors are not so accurate, and they do not provide as many aspects as the human sight (sensory gap), even if we use stereo vision (allows to create 3D images and analyze depth, [11]). Moreover, if we were able to overcome the sensory gap, the semantic gap will be tough to overcome. This issue is challenging because the computer at first must perform image preprocessing. In the given image (Figure 1), it seems reasonable to perform segmentation, in order to obtain the separated objects (monitor, computer, keyboard). However, at this stage, we need to answer the following question: How deep we want to segment? For example: Should the keyboard be segmented into separate buttons? Another problem at the preprocessing stage is: Which attributes (colour, shape, texture, etc.) need to be taken into account to create a versatile object descriptor? Moreover, which of them are more important, and why? These issues are challenging to answer, and it is extremely challenging to create a method that will match the human image recognition process (semantic gap). Therefore, the preprocessing stage is crucial. Many methods use segmentation as pre-processing [3, 45]. The accurate preprocessing is crucial because it eliminates any background noises and not important objects in the image. There are other preprocessing techniques such as image blurring, sharpening, edge detection, and many others. There are also combinations of them (e.g. blurring and edge detection). Although such methods are widely used in computer vision, it is complicated to choose one universal method for all image types.

Currently, accurate, so-called hand-crafted description of image objects is troublesome as our perception is different from a computer, pixel level-like one. Image features are usually based on various attributes e.g. color [32], texture [22], shape [6], local features, multiresolution wavelet analysis [16]. Novel methods for object description are constantly developed as they are needed in image comparison, retrieval, and classification.

One of the essential areas of computer vision is content-based image retrieval, allowing to retrieve similar images from extensive repositories. In the literature, there exist many exciting and sometimes groundbreaking papers [36]. It is not common that such systems can cope with general and various types of images. Thus, those systems need to be adjusted to the specific application (e.g. medical images). Although this process is necessary, most approaches have problems with retrieving images from different classes. For example, if a system has satisfying results for some visual class (e.g. dinosaurs), it can provide unsatisfactory results for another class (e.g. images of drinks). Moreover, many methods are dedicated to retrieving from databases of images with a single object. The researchers still try to develop CBIR methods which would allow retrieving images as effectively as text retrieval methods do.

In this paper, we present a novel fast solution to retrieve images. It relies on automatically detected objects and finding salient local image keypoints. Then, a histogram is built from quantized object outline angles and the keypoints. The retrieval is very fast and much faster than deep learning approaches.

The main innovations of the paper are as follows:

- We created an algorithm for automatic creating image area descriptors.
- We developed a method for determining important features of visual objects.
- We proposed an efficient content-based image retrieval system.
- We evaluated the proposed image descriptor and retrieval system on two large image datasets.

The paper is organized as follows. Section 2 describes some works from the literature and methods used in the approach presented in this paper. The novel method of describing images is presented in Section 3. It is used in Section 4 to create a content-based image retrieval system. Section 5 provides experimental results on the the PASCAL Visual Object Classes (VOC) 2012 dataset [10] and the COREL Database for Content-based Image Retrieval [40] to evaluate *Precision* and *Recall* [7] of the proposed method. Section 6 concludes the paper.

2 Related Work

Large-scale databases of images are growing rapidly with the Internet and the proliferation of recording devices. Taking into account the scale of such repositories, fast and accurate content-based image retrieval techniques play a crucial role in making use of them. They are based on detecting relevant features and then comparing images by these features, which can describe textures, color, edges or shapes. Textures can be detected by e.g. wavelet transform [44], and in [30] textures are described by a bank of Gabor filters. A very efficient method for texture description are local and semi-global Edge Histogram Descriptors (EHD) [34]. More sophisticated techniques are based on keypoints detected by approximations of Laplacian of Gaussian (LoG) with the most popular SIFT, SURF, ORB and FAST. In [15], SURF is modified to incorporate the information about the keypoint vicinity and then to retrieve images. SURF is combined with bag-of-words [13] in [2] to retrieve and classify images. Some retrieval approaches are base on fuzzy sets [14, 24, 25], fuzzy rules are used to classify and fast retrieve images in database environments.

2.1 Canny edge detection

Canny edge detection is a very popular algorithm proposed by John F. Canny in 1986 [4]. It produces from a grayscale image an image showing changes in brightness and is composed of five separate steps. The input image is denoised by Gaussian blur, computed by [29]

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{1}$$

where x and y are distances, and σ is a standard deviation of the Gaussian distribution. This is followed by computing the brightness gradient. Pixels with the highest gradient values are edges. The resultant bitmap with edges depends on the Gaussian filter which has an impact on the size of the detected edges, and the threshold used to determine edges. To remove breaks in edges, edge linking is used which connects broken edges. An example of the entire process is presented in Figure 2.

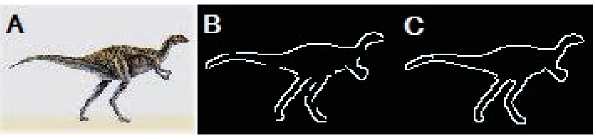

Figure 2. An example of the edge linking process. A - input image, B - edge detection, C - edge linking.

2.2 Speeded-Up Robust Features

SURF (Speeded-Up Robust Features) is local feature detector and descriptor [5] and a newer version of SIFT (Scale-Invariant Feature Transform) [28] enhanced in terms of speed. Nowadays it finds use in many applications such as object description [19], image retrieval [18], object tracking [17], and many others. As we mentioned, SURF is a similar algorithm to SIFT. Speed enhancement comes from the application of integral images as an alternative to the difference of Gaussian. Moreover, it generates a shorter descriptor, 64 elements long, so the images can be compared faster. The example of keypoints detection and matching are presented in Figure 3. The algorithm was also parallelized [39].

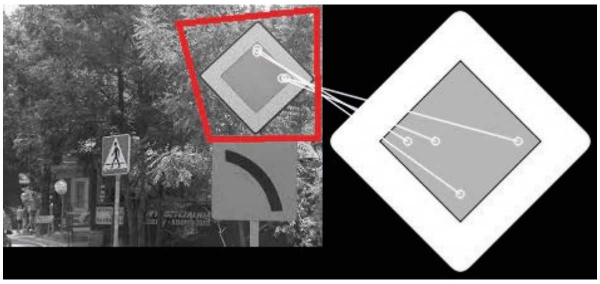

Figure 3. The SURF algorithm example with keypoint detection and matching.

3 Proposed Method for Image Description

In this Section, we present a method for feature extraction which is applied to compare images in Sections 4 and 5. The presented algorithm is based on edge detection (in our case Canny, see Section 2.1) and local keypoints (in our case SURF, see Section 2.2). Object detection consists in automatic finding regions of interest. The resulting quality depends heavily on the homogeneity of the objects. Usually, images do not contain easily extractable objects. The method proposed in the paper extracts important local features as a single set describing the object.

In the beginning, local interest keypoints are extracted. We use SURF ones, but a whole family of the algorithms can be applied in this step. Then, the feature vectors are grouped to compute centroids for histograms, and any broken edges are joined (see Fig 2C). After joining edges, we run a crawler (see Figure 2B).

We use the crawler to select important local points constituting a single, concise object. Algorithm 1 shows the details of the method, and Figure 4 presents an example run ([20]). Aside from surrounding the object, this step computes angles of the object borders.

The top left pixel is the starting point of the crawler, and the next pixel is determined from the current pixel neighbourhood. The direction is toward the neighbouring salient pixel and in the case of more than one salient pixels, we choose the first pixel. It is illustrated in Figure 6. We chose a stack (last in, first out) structure to store the object borders and the information if the pixel was visited already. Then, we determine the angle of the object border at the current pixel and assign it to one of the quantize values: $\phi \in \{45, 90, 135, 180, 225, 270, 315\}$. Such a set is chosen as a trade-off between the adequate description of the object shape and the speed.

All the keypoints inside the object border are left, and the keypoints outside the object borders are removed, what is presented in Figure 8. The salient keypoints are used to generate histograms based on clustered keypoints for the whole set of images. The number of buckets depends, of course, on the initial data set clustering, and is predefined or computed by some optimization method. The two computed angle and keypoint histograms constitute the image descriptor. We present the detailed method in Algorithm 2, and the algorithm flow in Figure 5.

	1													
81	82	2												
80		3												
79		4												
78			5											
77			6											
76				7										
75				8					17	18	19	20		
74	84			9				16				90	21	
	73			10			15			91				22
	72			11			14		94		92			23
	71				12	13				93				24
	70												25	
	69												26	
	68												27	
	67										30	29	28	
	66									31				
	65									32				
	64	85								33				
		63								34				
		62								35				
		61								36				
		60						89	37					
		59						38						
		58						39						
		57						40						
		56						41						
		55						88	42					
		54							43					
		53							44					
		52	86					87	45					
			51	50	49	48	47	46						

Figure 4. Example of determining regions of interest by the crawler.

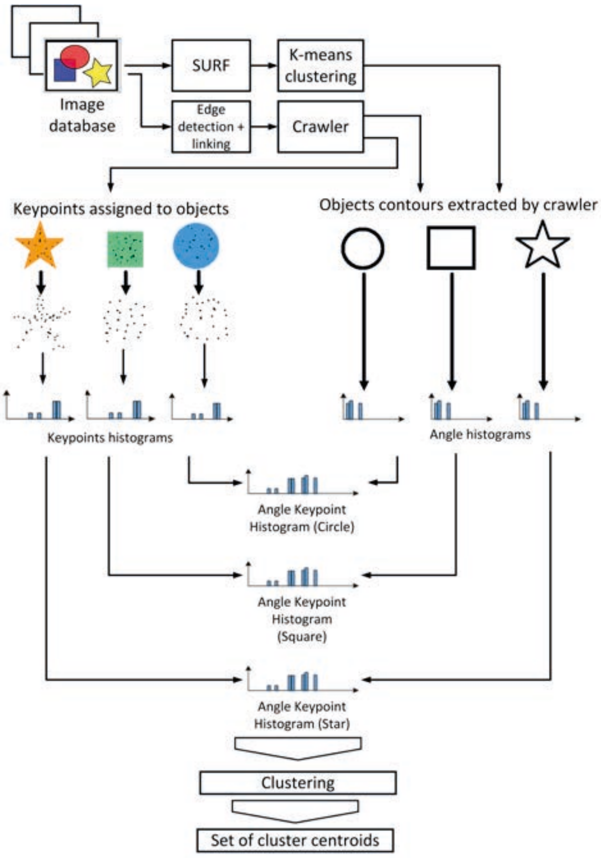
Image database
SURF
K-means clustering
Edge detection + linking
Crawler
Keypoints assigned to objects
Objects contours extracted by crawler
Keypoints histograms
Angle histograms
Angle Keypoint Histogram (Circle)
Angle Keypoint Histogram (Square)
Angle Keypoint Histogram (Star)
Clustering
Set of cluster centroids

Figure 5. To initialize the system all the images have to be indexed.

N		
	C	
P		

Figure 6. Neighbourhood of a pixel C, with the previous pixel P and the next pixel N.

45	90	135
P	C	180
315	270	225

Figure 7. The object boundary angle at the next pixel.

Algorithm 1 Crawler algorithm.

```
INPUT: $ObjectContours$
OUTPUT: $AngleHistogram$
$EdgeDetectIm :=$
  $CannyEdgeDetection(InputImage);$
$EdgeDetectIm :=$
  $EdgeLinking(EdgeDetectIm);$
$CurrentPixel :=$
  $FindPositionOfNearestPixel(EdgeDetectIm);$
$IsCrawlingCompleted := false;$
while $IsCrawlingCompleted = true$ do
    $NPixel = FindNextPixel();$
    while $NPixel! = NULL$ do
        $VisitedPixels.Add(CurrentPixel);$
        $PrevPixel := CurrentPixel;$
        $CurrentPixel := NPixel;$
        $NPixel := FindNextPixel();$
        $CurrentPixel.Angle :=$
          $Angle(PrevPixel, CurrentPixel, NPixel);$
        if
          $VisitedPixels.Contains(NPixel) = true$
          then
            $NPixel := NULL;$
            if $Branches.Count = 0$ then
                $IsCrawlingCompleted :=$
                  $true;$
            end
        end
    end
    if $Branches.Count > 0$ then
        $PrevPixel := NULL;$
        $CurrentPixel := Branches.Last();$
    end
    else if $Branches.Count = 0$ then
        $IsCrawlingCompleted := true;$
    end
end
```

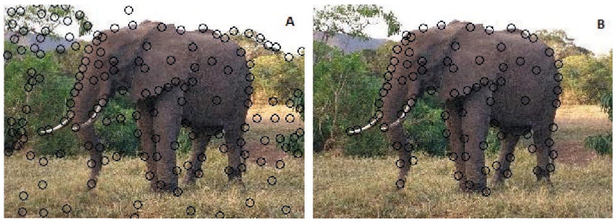

Figure 8. After the object detection, only important local interest points are kept.

Algorithm 2 The image indexing algorithm.

```
INPUT: $InputImages$
OUTPUT: $AngleKeypointHistograms$
$DetectedKeypoints := SurfDetector(InputImages)$;
$ClusterKeypoints(DetectedKeypoints)$;
foreach $InputImage \in InputImages$ do
    $EdgeDetectImg := Canny(InputImage)$;
    $EdgeDetectImg := EdgeLinking(EdgeDetectImg)$;
    $RunCrawler(EdgeDetectedImage)$;
    $ObjectsList := ExtractObjects()$;
    foreach $Obj \in ObjectsList$ do
        $ObjAngleHist := CreateAngleHist(Obj)$
        $ObjKeypoints := ExtractKeyPoints(Obj, DetectedKeypoints)$;
        $ObjKeypintsHist := CreateKeyPointHist(ObjKeypoints, Obj)$;
        $AngleKeyPointHist := Concat(ObjAngleHist, ObjKeypintsHist)$;
        $AngleKeypointHistograms.Add(AngleKeyPointHist)$;
    end
end
```

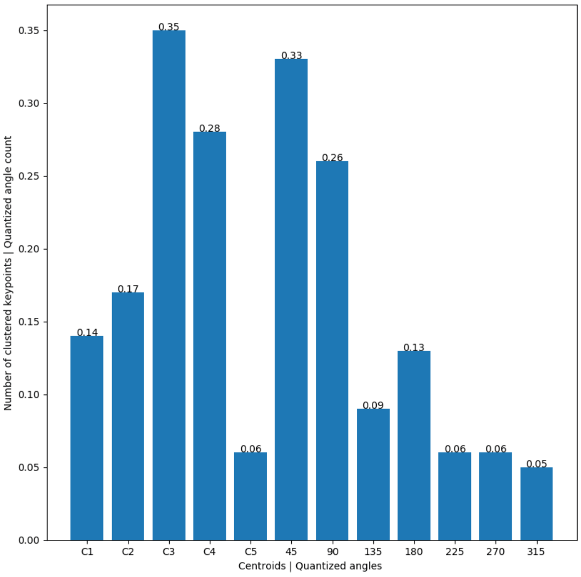

Figure 9. The concatenated histogram. It is composed of keypoint and angle histograms.

The above procedure allows for efficient descriptor extraction. Section 4 shows an application of the descriptor along with its immunity to scale or rotation.

4 Image Retrieval with Novel Descriptors

We designed a system utilizing the image descriptor from the previous Section. A general scheme is presented in Figure 10. The system computes visual features for the existing image set by the method presented in the previous Section. Nearly any data clustering algorithm can be used for histogram generation, and we applied *k*-means clustering [12, 23]. This resulted in a necessity to pick a number of clusters to achieve the right trade-off between the accuracy and speed. Other examples of semi-automatic clustering algorithms are mean shift or DBSCAN. Of course, we can use also other methods for computing local interest points and for edge detection.

The retrieval system is queried by the image which is analysed and described in the same way as the images in the database. Afterwards, we compute the similarity of the histograms of the query image and the database. Sometimes, there is a necessity to add a new set of images to the database. If the new images are from the same domain, it is enough to compute their features with the procedure used in the image retrieval. However, if the new images come from a substantially different domain, the whole existing database must be reindexed with the procedure from Section 3.

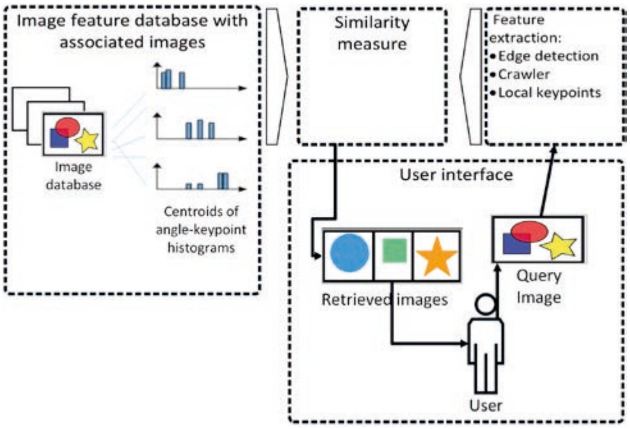

Figure 10. The CBIR architecture utilizing the method proposed in Section 3 as a feature extractor and descriptor.

5 Experiments

The experiments were performed using the software from [21] with to visual benchmarks:

- The PASCAL Object Recognition Database Collection, Unannotated Database - 101 Object Categories [10],
- The COREL Database for Content-based Image Retrieval [40].

We used all the classes in the datasets, and every class was divided into two sets of, respectively, 90% training images and (10%) query (test) images. The performance of the proposed method was evaluated with *Precision* and *Recall* measures [7, 41]. They are computed using: DIC - set of database images for given class of objects, RI - set of retrieved images for query, $RRI(TP)$ - set of relevant retrieved images (true positive), $IRI(FP)$ - irrelevant retrieved images (false positive), $RNRI(FN)$ - relevant not retrieved images (false, negative) and $IRNRI(TN)$ - irrelevant not retrieved images (TN). The general formulas used in machine learning for *Precision* and *Recall* are following [8, 41]

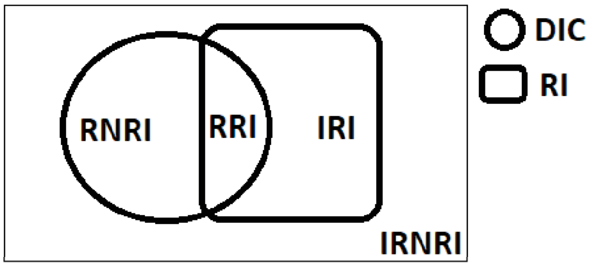

Figure 11. Performance measures diagram.

$$prec = \frac{Total\ number\ of\ retr.\ rel.\ images}{Total\ number\ of\ retrieved\ images}, \tag{2}$$

$$rec = \frac{Total\ number\ of\ retr.\ rel.\ images}{Total\ number\ of\ rel.\ images}. \tag{3}$$

Based on previously described sets and above formulas we can adapt formulas to CBIR needs

$$precision = \frac{|RRI|}{|RRI + IRI|}, \tag{4}$$

$$recall = \frac{|RRI|}{|RRI + RNRI|}. \tag{5}$$

We also define *Accuracy* measure described by the following formula [8]

$$Accuracy = \frac{|RRI + IRNRI|}{|Total|}. \tag{6}$$

Table 1 shows the accuracy measures values for random retrieved images of the Pascal dataset. As can be seen, the results are satisfying and they are better than our previous results obtained in [20]. Due to lack of space, we present only random classes from the whole dataset. The presented results prove that the proposed approach can be useful in fast image retrieval.

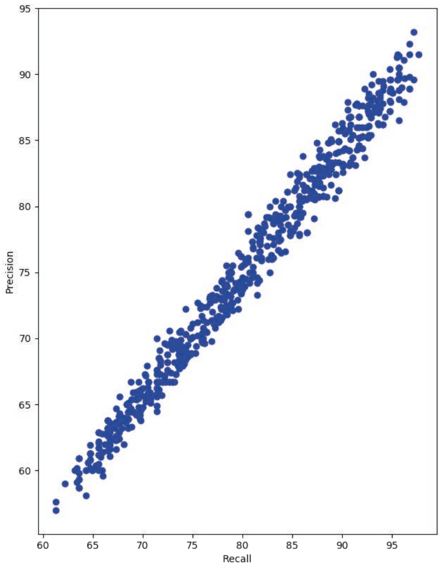

Figure 13. Precision and Recall measures for multi-query, performed on the Pascal dataset.

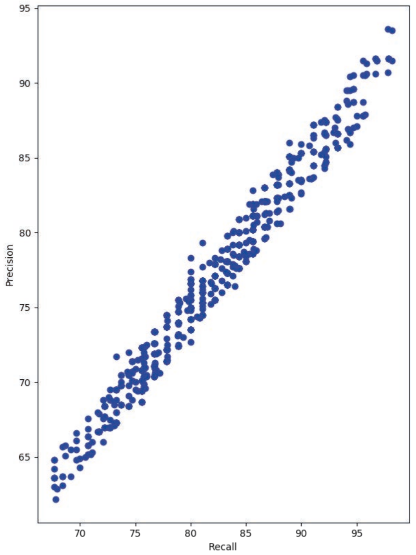

Figure 14. Precision and Recall measures for multi-query, performed on the Corel dataset.

Table 1. Proposed algorithm performance on the Pascal dataset. Due to lack of space, we present only a part of all queries from various classes, although Average Precision is calculated for all query images in the test set. CI_l and CI_h determine low and high confidence interval.

Id(Class)	RI	DIC	RRI (TP)	IRI (FP)	RNRI (FN)	IRNRI (TN)	Precision	Recall	Accuracy
706(Faces-easy)	381	353	275	106	78	6659	72.2	77.9	97.4
738(Faces-easy)	370	353	271	99	82	6666	73.2	76.8	97.5
4(accordion)	48	45	39	9	6	7064	81.2	86.7	99.8
32(Motorbikes)	696	648	419	277	229	6193	60.2	64.7	92.9
49(Motorbikes)	696	648	464	232	184	6238	66.7	71.6	94.2
65(Motorbikes)	689	648	438	251	210	6219	63.6	67.6	93.5
76(Motorbikes)	676	648	412	264	236	6206	60.9	63.6	93
205(Motorbikes)	689	648	470	219	178	6251	68.2	72.5	94.4
281(Motorbikes)	676	648	470	206	178	6264	69.5	72.5	94.6
300(Motorbikes)	676	648	425	251	223	6219	62.9	65.6	93.3
226(Motorbikes)	682	648	431	251	217	6219	63.2	66.5	93.4
238(Motorbikes)	683	648	419	264	229	6206	61.3	64.7	93.1
472(chandelier)	94	88	69	25	19	7005	73.4	78.4	99.4
495(euphonium)	55	53	40	15	13	7050	72.7	75.5	99.6
515(stop-sign)	56	53	48	8	5	7057	85.7	90.6	99.8
537(ferry)	58	55	50	8	5	7055	86.2	90.9	99.8
554(ferry)	59	55	46	13	9	7050	78	83.6	99.7
575(watch)	204	195	142	62	53	6861	69.6	72.8	98.4
591(watch)	212	195	144	68	51	6855	67.9	73.8	98.3
609(watch)	208	195	130	78	65	6845	62.5	66.7	98
725(Faces-easy)	370	353	282	88	71	6677	76.2	79.9	97.8
14(hawksbill)	85	81	69	16	12	7021	81.2	85.2	99.6
28(hawksbill)	87	81	68	19	13	7018	78.2	84	99.6
86(hawksbill)	86	81	65	21	16	7016	75.6	80.2	99.5
117(hawksbill)	89	81	66	23	15	7014	74.2	81.5	99.5
134(hawksbill)	88	81	65	23	16	7014	73.9	80.2	99.5
161(headphone)	37	35	24	13	11	7070	64.9	68.6	99.7
185(headphone)	37	35	25	12	10	7071	67.6	71.4	99.7
201(headphone)	37	35	25	12	10	7071	67.6	71.4	99.7
212(hedgehog)	49	45	33	16	12	7057	67.3	73.3	99.6
232(hedgehog)	49	45	33	16	12	7057	67.3	73.3	99.6
299(helicopter)	78	72	62	16	10	7030	79.5	86.1	99.6
562(ibis)	69	65	56	13	9	7040	81.2	86.2	99.7
578(ibis)	69	65	51	18	14	7035	73.9	78.5	99.6
603(wild-cat)	30	28	24	6	4	7084	80	85.7	99.9
Summary									
Average ≈							**74.97**	**80.02**	**98.66**
$\sigma \approx$							**8.81**	**9.35**	**1.81**
$CI_l \approx$							**57.69**	**61.70**	**95.10**
$CI_h \approx$							**92.25**	**98.34**	**100**

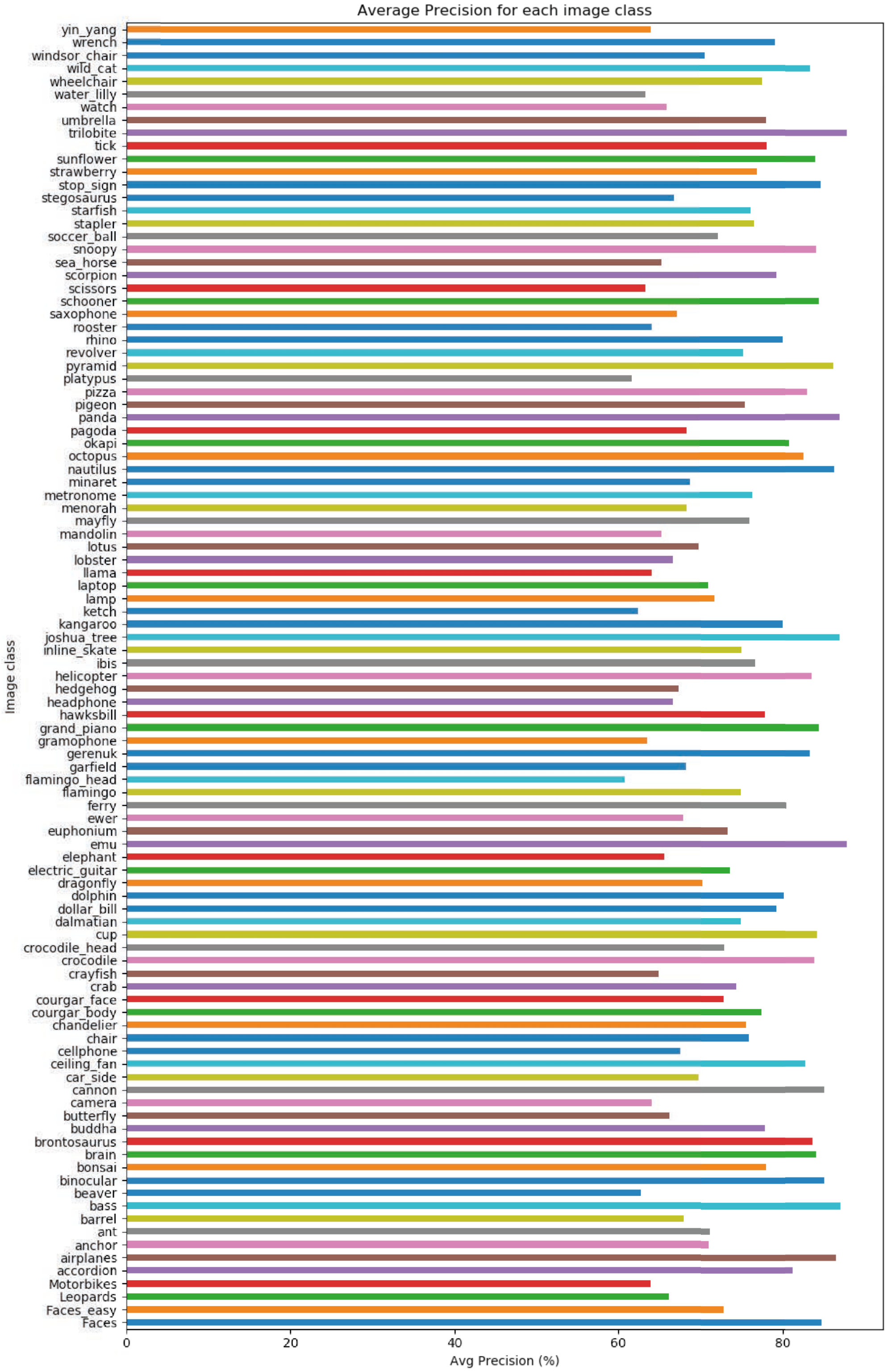

Figure 12. Average Precision for each image class for the proposed algorithm, performed on the Pascal dataset.

Table 2. Experiment results for the proposed algorithm, performed on the Corel dataset. Due to lack of space, we present only a part of all queries from various classes, although Average Precision is calculated for all query images in the test set. CI_l and CI_h determine low and high confidence interval.

Id(Class)	RI	DIC	RRI (TP)	IRI (FP)	RNRI (FN)	IRNRI (TN)	Precision	Recall	Accuracy
671(tiger)	97	90	69	28	21	8294	71.1	76.7	99.4
2(art)	95	90	79	16	11	8306	83.2	87.8	99.7
39(art)	96	90	80	16	10	8306	83.3	88.9	99.7
55(castle)	244	230	205	39	25	8143	84	89.1	99.2
80(castle)	246	230	202	44	28	8138	82.1	87.8	99.1
103(castle)	242	230	205	37	25	8145	84.7	89.1	99.3
122(decoys)	95	90	76	19	14	8303	80	84.4	99.6
138(decoys)	98	90	83	15	7	8307	84.7	92.2	99.7
177(streameng)	96	90	65	31	25	8291	67.7	72.2	99.3
197(streameng)	95	90	65	30	25	8292	68.4	72.2	99.3
213(train)	308	290	267	41	23	8081	86.7	92.1	99.2
259(train)	304	290	249	55	41	8067	81.9	85.9	98.9
332(drinks)	95	90	75	20	15	8302	78.9	83.3	99.6
359(drinks)	97	90	81	16	9	8306	83.5	90	99.7
381(elephant)	98	90	74	24	16	8298	75.5	82.2	99.5
403(elephant)	94	90	71	23	19	8299	75.5	78.9	99.5
168(balloon)	151	140	109	42	31	8230	72.2	77.9	99.1
188(balloon)	147	140	104	43	36	8229	70.7	74.3	99.1
233(bob)	53	50	43	10	7	8352	81.1	86	99.8
256(bob)	55	50	42	13	8	8349	76.4	84	99.8
456(lion)	98	90	79	19	11	8303	80.6	87.8	99.6
474(lion)	98	90	83	15	7	8307	84.7	92.2	99.7
13(dog)	316	290	206	110	84	8012	65.2	71	97.7
27(dog)	307	290	220	87	70	8035	71.7	75.9	98.1
514(mineral)	97	90	74	23	16	8299	76.3	82.2	99.5
145(flags)	95	90	74	21	16	8301	77.9	82.2	99.6
409(bus)	94	90	72	22	18	8300	76.6	80	99.5
427(bus)	95	90	73	22	17	8300	76.8	81.1	99.5
451(car)	435	400	307	128	93	7884	70.6	76.8	97.4
524(car)	423	400	335	88	65	7924	79.2	83.8	98.2
326(easteregg)	94	90	78	16	12	8306	83	86.7	99.7
443(deer)	207	190	182	25	8	8197	87.9	95.8	99.6
53(moleculr)	95	90	84	11	6	8311	88.4	93.3	99.8
495(aviation)	94	90	71	23	19	8299	75.5	78.9	99.5
196(flags)	95	90	79	16	11	8306	83.2	87.8	99.7
Summary									
Average ≈							**78.56**	**83.80**	**99.17**
$\sigma \approx$							**6.53**	**7.06**	**0.91**
$CI_l \approx$							**65.75**	**69.95**	**97.37**
$CI_h \approx$							**91.36**	**97.66**	**100**

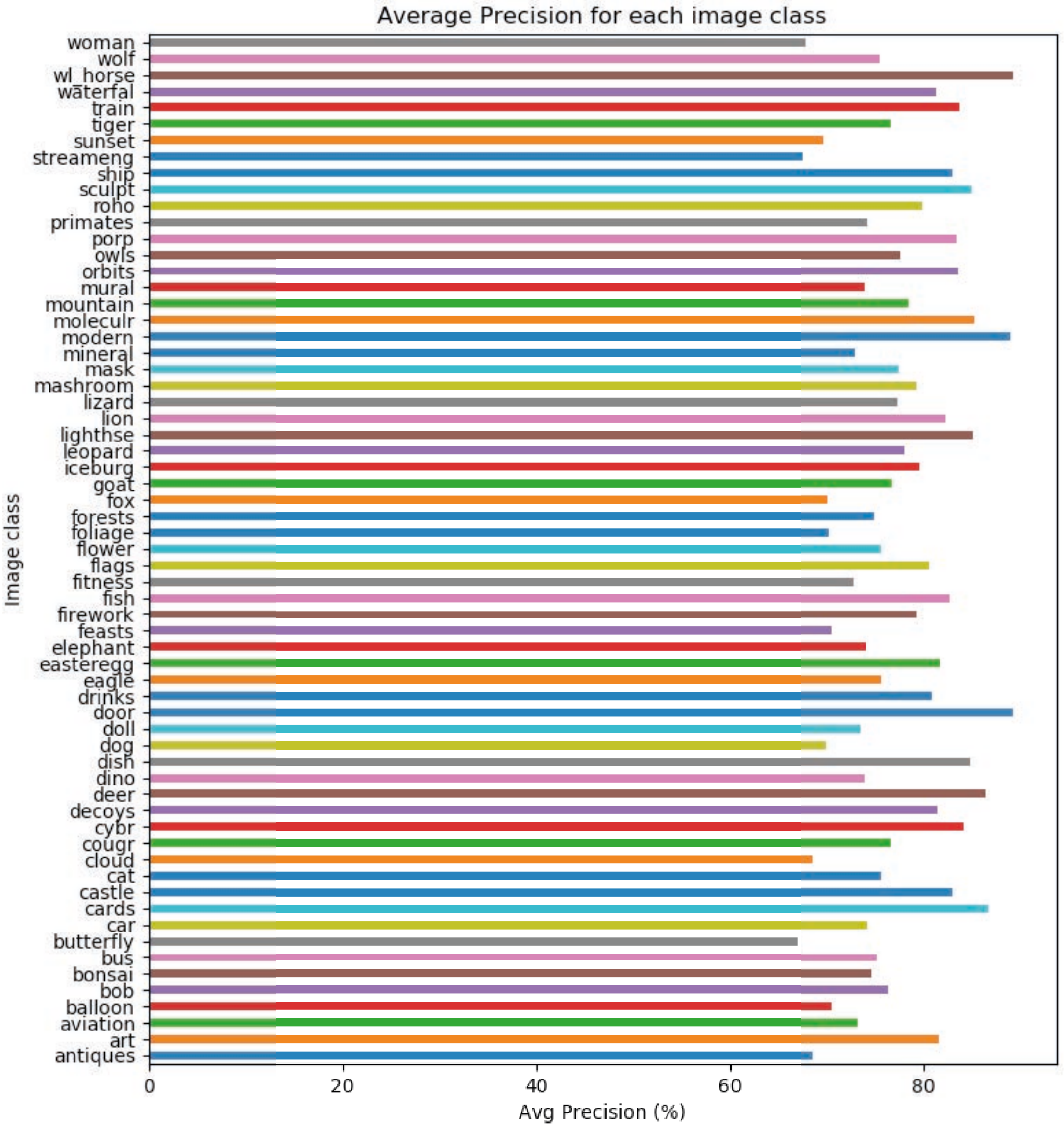

Figure 15. Average Precision for each image class for the proposed algorithm, performed on the Corel dataset.

The obtained results were compared with the current state of the art. The comparison is presented in Table 3.

In Table 2 results for randomly chosen classes of retrieved images from the Corel dataset are presented. As can be seen in the example query 537($ferry$) most images were correctly retrieved (50). Only eight of them are improperly recognized. The *precision* value for this experiment equals 86.2 and *recall* equals 90.9. The average precision equals 78.58 which is a very good result on the Pascal dataset. Figures 12 and 15 show the average precision for each class for both datasets. In Figure 13 and Figure 14 we present a point distribution of *precision* and *recall* for the given dataset.

Table 3. Comparison of Average Precision for the proposed method with the state of the art. The comparison was performed for the Corel dataset.

Method	Avg. prec.	Avg. recall
Prop.method	**0.785**	**0.838**
[9]	0.630	N/A
[27]	0.727	N/A
[42]	0.783	0.156
[43]	0.697	0.130
[1]	0.602	0.449
[35]	0.640	0.416
[33]	0.683	0.540
[38]	0.707	0.490
[31]	0.750	0.560

The time performance of the proposed method is determined by the size of images, hardware, and the algorithm parameters, e.g. in the case of SURF keypoints on the value of the $minHessian$ coefficient. We used one 12-core Intel Xeon E5-2620 2.0 Ghz processor, and the indexation step (without keypoint generation and edge detection) for the Pascal VOC dataset took approximately 10 seconds. The image retrieval step (Section 4) during experiments (i.e. for 10 % of all images) took, respectively, 8 min. 56 s. for the Corel dataset and 20 min. 36 s. for the Pascal VOC dataset.

In Table 3 we compared our method with state of the art methods. The Average Precision is slightly greater than the highest value of the other methods. *Recall*, on the other hand, is much higher than other methods.

6 Conclusions and Future Work

We developed a new method for efficient image description and retrieval. It relies on automatically detected objects and finding salient local image keypoints. Objects are found by crawling edges and described by local interest point descriptors and edge angles. Then, a histogram is built from quantized object outline angles and the keypoints. The method provides the angle-keypoint histogram as an object descriptor; thus, the comparison in the indexation phase (see Figure 10) is relatively fast and straightforward. We showed that the framework is more accurate than similar solutions from the literature. The approach can be used in various applications that need fast image retrieval. Although slightly less accurate, the proposed algorithm is much faster than the solutions based on convolutional neural networks [26]. The retrieval speed is at least ten times faster in the case of the presented framework.

Acknowledgements

This job is supported by the National key R&D Program of China under Grant NO. 2018YFB0203901 and the Key Research and Development Program of Shaanxi Province (No.2018ZDXM-GY-036) and Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data (No.IPBED7).

References

[1] Agarwal, M., Maheshwari, R.: Á trous gradient structure descriptor for content based image retrieval. International Journal of Multimedia Information Retrieval 1(2), 129-138 (2012)

[2] Alfanindya, A., Hashim, N., Eswaran, C.: Content based image retrieval and classification using speeded-up robust features (surf) and grouped bag-of-visual-words (gbovw). In: 2013 International Conference on Technology, Informatics, Management, Engineering and Environment, pp. 77-82 (2013). DOI 10.1109/TIME-E.2013.6611968

[3] An, Y., Riaz, M., Park, J.: Cbir based on adaptive segmentation of hsv color space. In: Computer Modelling and Simulation (UKSim), 2010 12th International Conference on, pp. 248-251. IEEE (2010)

[4] Bao, P., Zhang, L., Wu, X.: Canny edge detection enhancement by scale multiplication. IEEE transactions on pattern analysis and machine intelligence 27(9), 1485-1490 (2005)

[5] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision, pp. 404-417. Springer (2006)

[6] Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence 24(4), 509-522 (2002)

[7] Buckland, M., Gey, F.: The relationship between recall and precision. Journal of the American society for information science 45(1), 12 (1994)

[8] Das, S., Garg, S., Sahoo, G.: Comparison of content based image retrieval systems using wavelet and curvelet transform. The International Journal of Multimedia & Its Applications 4(4), 137 (2012)

[9] Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Information retrieval 11(2), 77-107 (2008)

[10] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2), 303-338 (2010)

[11] Fang, Y., Wang, J., Yuan, Y., Lei, J., Lin, W., Callet, P.L.: Saliency-based stereoscopic image retargeting. Information Sciences 372(Supplement C), 347-358 (2016)

[12] Ferdaus, M.M., Anavatti, S.G., Garratt, M.A., Pratam, M.: Development of c-means clustering based adaptive fuzzy controller for a flapping wing micro air vehicle. Journal of Artificial Intelligence and Soft Computing Research 9(2), 99-109 (2019). DOI 10.2478/jaiscr-2018-0027

[13] Gabryel, M.: The bag-of-words methods with pareto-fronts for similar image retrieval. In: R. Damaševičius, V. Mikašytė (eds.) Information and Software Technologies, pp. 374-384. Springer International Publishing, Cham (2017)

[14] Gabryel, M., Korytkowski, M., Scherer, R., Rutkowski, L.: Object detection by simple fuzzy classifiers generated by boosting. In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, J. Zurada (eds.) Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 7894, pp. 540-547. Springer Berlin Heidelberg (2013)

[15] Gopal, N., Bhooshan, R.S.: Content based image retrieval using enhanced surf. In: 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1-4 (2015). DOI 10.1109/NCVPRIPG.2015.7490035

[16] Grossmann, A.: Wavelet transforms and edge detection. In: Stochastic processes in physics and engineering, pp. 149-157. Springer (1988)

[17] Grycuk, R.: Novel visual object descriptor using surf and clustering algorithms. Journal of Applied Mathematics and Computational Mechanics 15(3), 37-46 (2016)

[18] Grycuk, R., Gabryel, M., Korytkowski, M., Scherer, R.: Content-based image indexing by data clustering and inverse document frequency. In: Beyond Databases, Architectures and Structures 2014, Communications in Computer and Information Science, pp. 374-383. Springer Berlin Heidelberg (2014). Manuscript accepted for publication

[19] Grycuk, R., Gabryel, M., Korytkowski, M., Scherer, R., Romanowski, J.: Improved digital image segmentation based on stereo vision and mean shift algorithm. In: Parallel Processing and Applied Mathematics 2013, Lecture Notes in Computer Science, pp. 433-443. Springer Berlin Heidelberg (2014). Manuscript accepted for publication

[20] Grycuk, R., Gabryel, M., Scherer, M., Voloshynovskiy, S.: Image descriptor based on edge detection and crawler algorithm. In: International Conference on Artificial Intelligence and Soft Computing, pp. 647-659. Springer International Publishing (2016)

[21] Grycuk, R., Gabryel, M., Scherer, R., Voloshynovskiy, S.: Multi-layer architecture for storing visual data based on wcf and microsoft sql server database. In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (eds.) Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 9119, pp. 715-726. Springer International Publishing (2015)

[22] Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pp. 762-768. IEEE (1997)

[23] Koren, O., Hallin, C.A., Perel, N., Bendet, D.: Decision-making enhancement in a big data environment: Application of the k-means algorithm to mixed data. Journal of Artificial Intelligence and Soft Computing Research 9(4), 293–302 (2019). DOI 10.2478/jaiscr-2019-0010

[24] Korytkowski, M., Rutkowski, L., Scherer, R.: Fast image classification by boosting fuzzy classifiers. Information Sciences 327, 175-182 (2016)

[25] Korytkowski, M., Senkerik, R., Scherer, M.M., Angryk, R.A., Kordos, M., Siwocha, A.: Efficient image retrieval by fuzzy rules from boosting and metaheuristic. Journal of Artificial Intelligence and Soft Computing Research 10(1), 57-69 (2020)

[26] Kumarratneshk, R., Weilleweill, E., Aghdasi, F., Sriram, P.: A strong and efficient baseline for vehicle re-identification using deep triplet embedding. Journal of Artificial Intelligence and Soft Computing Research 10(1), 27-45 (2020). DOI 10.2478/jaiscr-2020-0003

[27] Lin, C.H., Chen, R.T., Chan, Y.K.: A smart content-based image retrieval system based on color and texture feature. Image and Vision Computing 27(6), 658-665 (2009)

[28] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91-110 (2004)

[29] Luo, Y., Duraiswami, R.: Canny edge detection on nvidia cuda. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, pp. 1-8. IEEE (2008)

[30] Ma, W.Y., Manjunath, B.: Netra: A toolbox for navigating large image databases. Multimedia Syst. 7(3), 184-198 (1999)

[31] Memon, M.H., Li, J.P., Memon, I., Arain, Q.A.: Geo matching regions: multiple regions of interests using content based image retrieval based on relative locations. Multimedia Tools and Applications 76(14), 15,377-15,411 (2017)

[32] Meskaldji, K., Boucherkha, S., Chikhi, S.: Color quantization and its impact on color histogram based image retrieval accuracy. In: 2009 First International Conference on Networked Digital Technologies, pp. 515-517. IEEE (2009)

[33] Murala, S., Maheshwari, R., Balasubramanian, R.: Directional local extrema patterns: a new descriptor for content based image retrieval. International journal of multimedia information retrieval 1(3), 191-203 (2012)

[34] Park, D.K., Jeon, Y.S., Won, C.S.: Efficient use of local edge histogram descriptor. In: Proceedings of the 2000 ACM Workshops on Multimedia, MULTIMEDIA '00, pp. 51-54. ACM, New York, NY, USA (2000). DOI 10.1145/357744.357758. http://doi.acm.org/10.1145/357744.357758

[35] Saadatmand-Tarzjan, M., Moghaddam, H.A.: A novel evolutionary approach for optimizing content-based image indexing algorithms. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37(1), 139-153 (2007)

[36] da Silva Júnior, J.A., Marçal, R.E., Batista, M.A.: Image retrieval: Importance and applications. In: Workshop de Visao Computacional-WVC, pp. 311-315 (2014)

[37] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on pattern analysis and machine intelligence 22(12), 1349-1380 (2000)

[38] Sumana, I.J., Islam, M.M., Zhang, D., Lu, G.: Content based image retrieval using curvelet transform. In: Multimedia Signal Processing, 2008 IEEE 10th Workshop on, pp. 11-16. IEEE (2008)

[39] Šváb, J., Krajník, T., Faigl, J., Přeučil, L.: Fpga based speeded up robust features. In: Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on, pp. 35-41. IEEE (2009)

[40] Tao, D.: The corel database for content based image retrieval (2009)

[41] Ting, K.M.: Precision and recall. In: Encyclopedia of machine learning, pp. 781-781. Springer (2011)

[42] Walia, E., Pal, A.: Fusion framework for effective color image retrieval. Journal of Visual Communication and Image Representation 25(6), 1335-1348 (2014)

[43] Wang, C., Zhang, B., Qin, Z., Xiong, J.: Spatial weighting for bag-of-features based image retrieval. In: International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making, pp. 91-100. Springer (2013)

[44] Wang, J.Z., Li, J., Wiederhold, G.: Simplicity: semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(9), 947-963 (2001). DOI 10.1109/34.955109

[45] Wang, L., Chang, Y., Wang, H., Wu, Z., Pu, J., Yang, X.: An active contour model based on local fitted images for image segmentation. Information Sciences 418-419(Supplement C), 61-73 (2017)

Rafal Grycuk received his M.Sc. and Ph.D. degrees in computer science from Czestochowa University of Technology in 2012 and 2017, respectively. Currently, he is an assistant professor at the Czestochowa University of Technology. His scientific interests cover computer vision, image description, content-based image retrieval and unsupervised learning.

Adam Wojciechowski received an M.Sc. degree in computer science from the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, in 1999. In 2005 he received a Ph.D. degree in computer science, specialization: computer graphics and vision, from the Faculty of Automatics, Electronics and Computer Science, Silesian University of Technology. The degree of D.Sc. in computer science, he received in 2014 from the Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology. Since 1999 he has been working in the Institute of Information Technology, Lodz University of Technology. Since 2016, he is a vice-Dean for Development and Business Cooperation, and since 2019 he is the Head of the Institute of Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

His research interests encompass human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry, and machine learning.

Wei Wei is a senior member of IEEE, an associated professor of School of Computer Science and Engineering. Xi'an University of Technology, Xi'an 710048, China. He received his Ph.D. and M.S. degrees from Xian Jiaotong University in 2011 and 2005, respectively. He ran many funded research projects as principal investigator and technical member. His research interest is in the area of wireless networks, wireless sensor networks Application, Image Processing, Mobile Computing, Distributed Computing, and Pervasive Computing, Internet of Things, Sensor Data Clouds, etc. He has published around one hundred research papers in international conferences and journals. He is an editorial board member of FGCS, AHSWN, IEICE, KSII, etc. He is a TPC member of many conferences and regular reviewer of IEEE TPDS, TIP, TMC, TWC, and many other Elsevier journals.

Agnieszka Siwocha received M.Sc. the degree from Lodz University of Technology, Faculty of Technical Physics, Computer Science and Applied Mathematics, and her a Ph.D. in 2015 in the field of computer science, computer graphics at the University of Social Sciences, Łódź, Poland. She is currently an assistant professor at the Social Academy of Sciences in Łódź. She has a title of Adobe Certified Expert, and provides training on Adobe software and computer graphics. Author of over 20 publications related to various problems of computer science, computer graphics and IT applications. Her present research interests include fractal coding, compression, and the quality of digital images, computer graphics, machine learning, and multifractal analysis.