International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 16, No. 1, pp. 68 – 85, 2019

IMPROVED QUERY BY APPROXIMATE SHAPES IMAGE RETRIEVAL METHOD

ROMAN STANISŁAW DENIZIAK

Kielce University of Technology, al. Tysiaclecia Panstwa Polskiego 7,
25-314 Kielce, Poland,
s.deniziak@tu.kielce.pl

TOMASZ MICHNO

Kielce University of Technology, al. Tysiaclecia Panstwa Polskiego 7,
25-314 Kielce, Poland,
t.michno@tu.kielce.pl

In this paper we present a new sketch-based Content Based Image Retrieval method. The main idea of the algorithm is based on representation of objects using a set of predefined shapes, called primitives: line segments, arches, polylines, polygons, polyarches (a chain of connected arches) and arc sided polygons (a looped chain of connected arches). Each type is defined by its attributes - for example as a line segment attribute its slope is used and for an arc its angle. To improve the efficiency of the shapes extraction from the image, the HSL color space is used. After detection, all shapes are connected into a graph in order to store also the mutual relations between them. As the storage for graphs, a structure based on a tree is used. In order to reduce the height of the tree, a special types of leaf nodes are used.

Keywords: CBIR; sketch; multimedia databases.

1. Introduction

Nowadays the area of usage of the multimedia databases is growing very rapidly. They are used for example in social media portals, authorizing or monitoring systems. Moreover, many applications must deal with rotated or distorted images and with very huge amount of data to store. Thus, they need effective storage and querying methods in order to provide the proper results for users.

In this paper we present a new sketch-based Content Based Image Retrieval method. The main idea of the algorithm is based on representation of objects using a set of predefined shapes, called primitives: line segments, arches, polylines, polygons, polyarches (a chain of connected arches) and arc sided polygons (a looped chain of connected arches). Each type of the predefined shape is defined by its attributes- for example as a line segment attribute its slope is used and for an arc its angle. To improve the efficiency of the shapes extraction from the image, the HSL color space is used. After detection, all shapes are connected into a graph in order to store also the mutual relations between them. As the storage for graphs, a structure based on a tree is used. In order to reduce the height of the tree, a special types of leaf nodes are used. Comparing to our previous research results, this paper improves the shapes extraction algorithm and the results precision coefficient.

The Query by Approximate Shapes provide for user easy to drawn queries, without need of high drawing skills. Moreover, if a user has example image, it may be used also as a query using the same database data. Since the database structure is based on a tree, the querying process is faster because all similar nodes are stored in the same part of a subtree or in the same leaf node. Moreover, the parallel query processing is also possible, which may additionally improve the computation time.

The paper is organized as follows: the first section is an introduction. Next image retrieval from multimedia databases algorithms are described. The third section describes our motivation. Next section presents the proposed object representation. The fifth section describes the shapes extraction algorithm with improvements and the sixth possible usages of Edge Boxes algorithm. The eight section presents the experimental results. The next section is the summary and future research section. The last one section is a list of references used in this paper.

2. Related works

The image retrieval from multimedia databases problem has been researched for many years. In this area, there can be distinguished three types of algorithms:

- the Keywords Based Image Retrieval (KBIR) algorithms,
- the Content Based Image Retrieval (CBIR) algorithms,
- the Semantic Based Image Retrieval (SBIR) algorithms.

The Keyword Based Image Retrieval algorithms use annotations (most often a set of keywords) to store the description of details present in the image. The main advantage of such approach is the simplicity of both storage and query. The user use as a query keywords, which then are compared with keywords assigned with each image. For objects which are easy to describe textually, the results are very precise. For example a car may be annotated using the brand, model, color and year of production. When objects are harder to annotate by keywords, e.g. they contain unknown details for a user or details which may be described mistakenly the results are not as precise as using other methods [Deniziak and Michno, 2017].

The Content Based Image Retrieval algorithms use the content present in the image without creating any textual annotations. Due to that fact, they are suitable for users who does not have complete knowledge about searched objects, when objects are hard to describe by keywords or there exists an example image which may be used as a query. The CBIR algorithms may be divided into low-level and high-level algorithms [Deniziak and Michno, 2017].

The low-level CBIR algorithms use global, statistical image features which may be for eaxample a contrast, entropy [Kriegel et al., 2006] or a normalized histogram of colors [Mocofan et al., 2011]. There are also approaches which uses shape, color and texture descriptors [Evgeniou et al., 2003]. When an image is added to the database, most often the features are extracted and also stored. As a query, a user use an example image, from which the same types of features are extracted. Then they are compared with all images features stored in the database. The low-level CBIR algorithms provide precise results for querying for similar images (e.g. paintings). When the query is executed in order to find images with an object which may have different background and neighborhood, the precision of the results is much lower and other methods (e.g. high-level) should be used.

The high-level CBIR algorithms does not use the global image features, like low-level, but tries to decompose it into smaller parts. One of the most commonly used approach is based on gathering similar pixels into so called regions. A region is a part of the image which most often contains pixels with similar colors. After regions extraction they are transformed into a graph which stores them as nodes. The information about connected regions is stored using edges between nodes. During queries the graphs of an image query and images stored in the database are compared using e.g. Maximum Likelihood [Li and Hsu, 2008]. The main disadvantage of region-based methods is the need of example image as a query. Users often does not have full knowledge about searched object, thus they may not have such an image. Moreover, preparing such an image by users may be complicated and need high drawing skills, because it should contain all details in colors.

There are also CBIR algorithms which tries to minimize the need of detailed, colored image as a query using a simple rough sketch. Thanks to that, most of users are able to draw a query without having high drawing skills and even when they does not have full knowledge about searched object. In some Sketch Based CBIR methods Standard Histogram of Oriented Gradients, Edge Relation Histogram or SIFT (Scale-invariant feature transform) algorithms are used. They are sensitive to inaccuracies during drawing, because even the same person is not able to create the same sketch in the same way more than once. Moreover, all sketches are extracted directly from the whole image and may need more storage space. Other methods use for example Hidden Markov Models which are extracted from colors and shapes present in the image. Then they are connected into a stochastic model [Müller et al., 1999]. The disadvantage of the method is querying by filled sketches without any inner details.

One of the most advanced Sketch Based CBIR method and the most similar to our approach is [Parui and Mittal, 2015] which is based on line segment detection. All found segments are transformed into an object outline choosing the longest chain. Additionally, as an assistance to the object outline creation, the Geodesic Object Proposals (GOP) [Krähenbühl and Koltun, 2014] algorithm is used to detect the area of the object in the image. When the outline is created, it is transformed into a descriptor which is compared with descriptors stored in the database. Each descriptor contains the information about relations of line segments lengths to each others and angles between all segments. In comparison with our approach, the [Parui and Mittal, 2015] does not allow drawing inner details in object sketches and does not cover other types of outline than line segments which may lead to inaccuracies and problems with describing some types of objects. Moreover our approach is designed to allow using both types of queries - hand drawn sketches and sketches extracted from the example input image without rebuilding the whole database structure.

There are also some Sketch Based algorithms which use neural networks, e.g. deep convolutional neural networks [Wang et al., 2016]. In this approach both sketches and images are used to train the structure. The main disadvantage is the need of learning the network when a new class of object is added. Moreover, the sketches used for training are not extracted directly from the image which are connected with them.

The Semantic Based Image Retrieval algorithms tries to minimize the difference between the data present in the image and the information which can be deduced by a human [Wang et al., 2010; Singh et al., 2012]. There are methods which use textual queries and graphical queries. Most of Semantic Based algorithms use neural networks or other AI methods, due to that fact very often they need time consuming learning, also after adding a new class of object.

Some algorithms in order to extract the information about objects present in the image use dedicated algorithms which tries to find and mark the areas where there are some objects. One of the most popular algorithm in this area is the Geodesic Object Proposals [Krähenbühl and Koltun, 2014] which identifies a set of object candidates in the image. In order to create the proposals, geodesic distance transforms are computed using classifiers which are optimized to object discovery. The resulted proposals may have different shapes and covers the whole region of pixels. Another algorithm in this area is the Edge Boxes [Zitnick and Dollár, 2014]. The algorithm is based on the idea that the number of contours which are wholly enclosed in the bounding box may point that it contains an object. If many contours are enclosed, the likelihood of an object enclosed by a box is high, when there is a less number, the likelihood is low. The method uses Structured Edge detector to extract the edge map from the image. Then the edges are evaluated using a sliding window approach and efficient data structures which improves the computation time. As an algorithm results, the proposals in a form of bounding box are given.

3. The motivation

Querying the multimedia database is a very complicated process because objects may be distorted, rotated, have different sizes or backgrounds. Furthermore, each new image inserted into database have to be added automatically without need of a human interaction. Therefore, our motivation to proposed a new sketch-based CBIR method which tries to overcome such problems was following:

- simple sketch used as a query which does not users to have need high drawing skills,
- ability to use also an example image as a query,
- each object should be represented in order to store also its curved shapes,
- insertion of a new image without need of learning or rebuilding the whole database structure,

- efficient database structure, even for huge number of stored data,

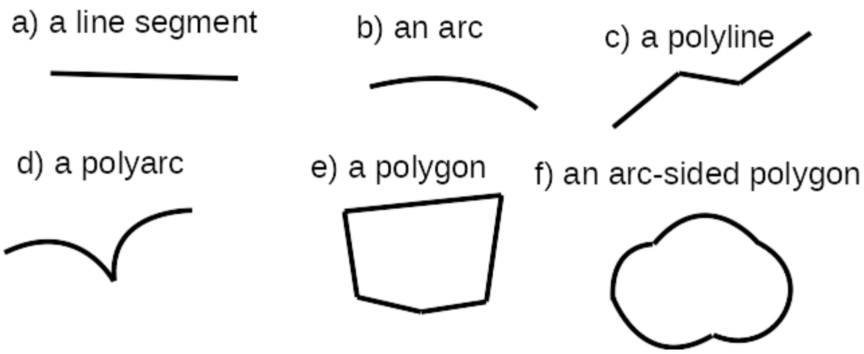

Fig. 1. The proposed primitives: a) a line segment, b) an arc, c) a polyline, d) a polyarc, e) a polygon, f) an arc-sided polygon.

-
- ability to parallel queries,the information about the similarity of each result with the query object should be also returned to the user.

4. The object representation

One of the aims of our previous researches was to find the method of the description of an object in the multimedia database. We realized that most objects may be represented using a sketch, which causes that there is no need of an example image. Moreover, if a user have such an image, it may be transformed into a sketch.

The sketch may be represented by an image (e.g. using black pixels as sketch edges [Kato et al., 1992]), an object outline (e.g. as in [Parui and Mittal, 2015]) or, as proposed in our research, as a set of predefined shapes. The advantage of our approach is the simplicity to draw a sketch by users, ability to store additional information for each shape (e.g. information about material or color). The predefined, simple shapes are called primitives. During our research we noticed, that line segments (Fig. 1 a)) and arches (Fig. 1 b)) are suitable for describing a sketch of most objects. Moreover, during experiments, we realized that storing the information about connected segments and arches improves the efficiency of the algorithm. Thus, additional primitives are proposed:

- based on line segments: polylines (Fig. 1 c)) and polygons (Fig. 1 e)),

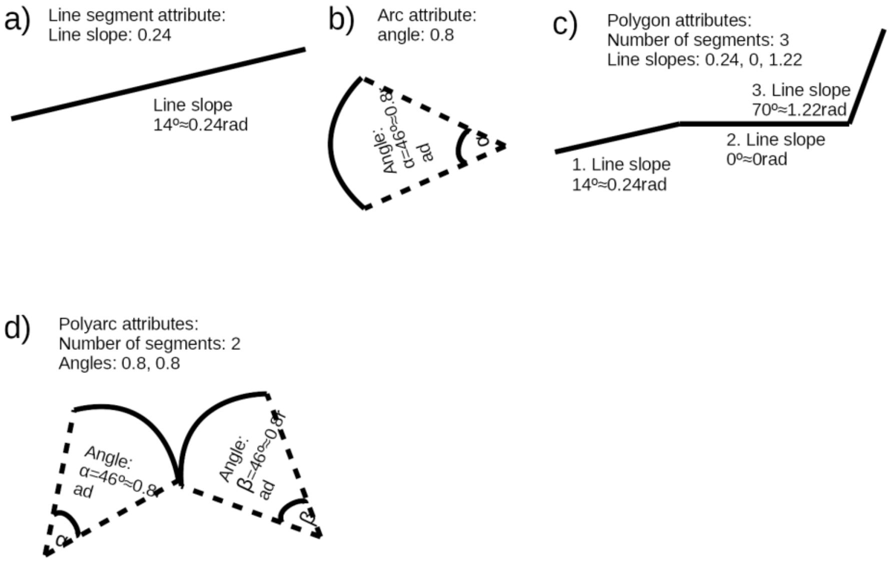

Fig. 2. The attributes of different primitives: a) a line segment, b) an arc, c) a polyline, d) a polyarc.

-

based on arches: polyarches (a chain of connected arches, Fig. 1 d)) and arc-sided polygons (a looped chain of connected arches, Fig. 1 f)).Since all primitives are based on line segments or arches, they are called base primitives, afterwards all primitives created using them are called complex primitives. Each primitive is defined by its type and an attribute or set of attributes, which are defined as follows:

- a line segment attribute is its line slope (Fig. 2 a)),
- an arc attribute is its angle (Fig. 2 b)),
- a polyline and a polygon attributes are number of segments and their line slopes (Fig. 2 c)),
- a polyarc and an arc-sided polygon attributes are number of segments and their angles (Fig. 2 d)).

During our research we realised that not only the information about primitives is needed but also an information which shapes are connected. Thus, such an information is stored using a graph where nodes are used to store primitives and edges used to store connections between them. This approach is similar to graphs of regions in region-based CBIR algorithms. Moreover storing the information about mutual positions between connected primitives also improves the efficiency of the image retrieval from the multimedia database [Deniziak and Michno, 2016]. Therefore, for each connection, there is stored the information about positions using the geographical windrose (N, S, E and W directions). The example of the graph is shown in the Fig. 3.

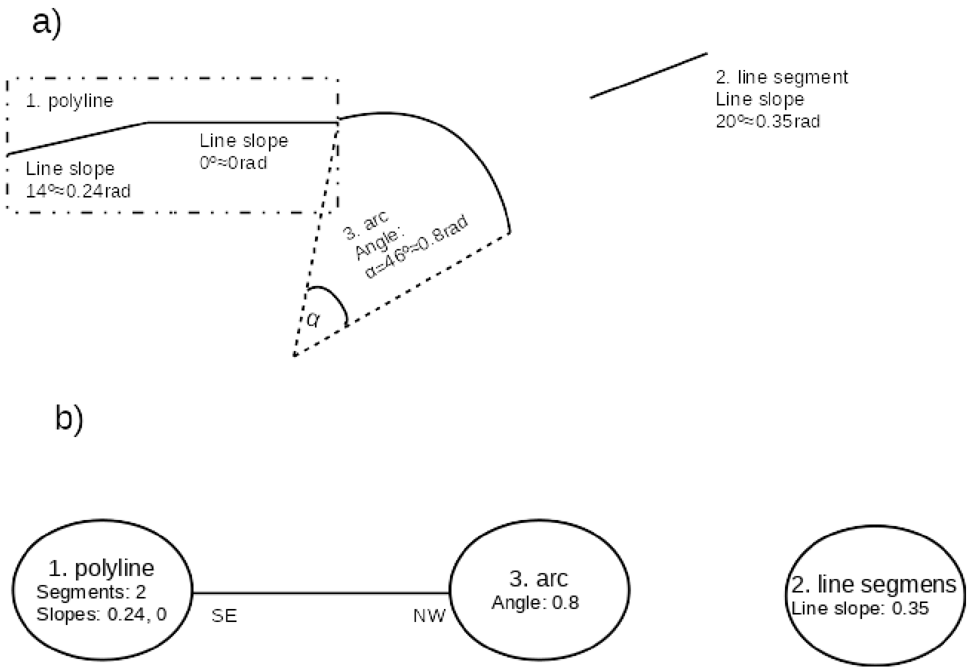

Fig. 3. The example of the graph which contains 3 primitives: a) primitives and their attributes(a polyline, an arc and a line segment), b) the graph: a polyline (1) and an arc (3) are connected, no. 3 lies on the right bottom (SE) of no. 1 and consequently no. 1 lies on the top left (NW) of no. 3; the node no. 2 does not have any connection.

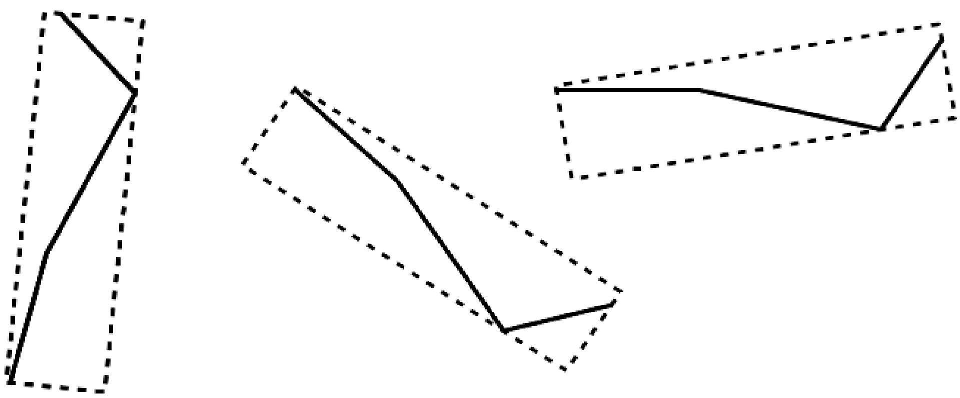

Fig. 4. The example polyline object with its Object Oriented Bounding Box (marked by dotted lines) in three different orientations.

Since an image may contain many objects, in order to clearly emphasize that they are separated in a graph, a structure called complex shape was introduced [Deniziak and Michno, 2016]. It may be used optionally or mandatory based on types of stored images in the multimedia database. In order to improve the algorithm resistance to different objects orientations, as an addition to the previous works, we propose to store primitives attributes in axis independent manner. During the stage of attributes values computing, line slopes may be computed in relation to the longest side of the Object Oriented Bounding Box built on top of the object. The example bounding box of a polyline is shown in the Fig. 4. Such an approach allows comparisons of differently oriented primitives which are strictly the same or very similar with high precision. Without such solution, there may be some problems with proper image retrieval from the database when users draw a sketch which is e.g. rotated in comparison with sketches stored in the database. During previous experiments we found that storing the information about mutual positions of connected nodes highly improves the precision of the image retrieval algorithm results. Due to that fact, we propose to store the same information also for each pair of nodes (not only connected), similarly to coincidence matrix used in graphs. In this paper we test if such an approach improves the precision of the results.

In order to improve the algorithm resistance to different objects orientations, as an addition to the previous works, we propose to store primitives attributes in axis independent manner. During the stage of attributes values computing, line slopes may be computed in relation to the longest side of the Object Oriented Bounding Box built on top of the object. The example bounding box of a polyline is shown in the Fig. 4. Such an approach allows comparisons of differently oriented primitives which are strictly the same or very similar with high precision. Without such solution, there may be some problems with proper image retrieval from the database when users draw a sketch which is e.g. rotated in comparison with sketches stored in the database. During previous experiments we found that storing the information about mutual positions of connected nodes highly improves the precision of the image retrieval algorithm results. Due to that fact, we propose to store the same information also for each pair of nodes (not only connected), similarly to coincidence matrix used in graphs. In this paper we test if such an approach improves the precision of the results.

5. The image shapes extraction algorithm based on HSL image representation

In this paper we propose to enhance the shapes extraction algorithm in comparison with the previous version used in [Deniziak and Michno, 2017]. The main idea of the improvement is based on converting an image into a HSL color space. Due to the conversion three images are created: first with only hues present in the image, the second one with lightness information and the last one which stores saturation of different areas (the example image representation is shown in the Fig. 5). Thanks to that, the algorithm is able to detect line segments and arches with the same hue, the same lightness level and the same saturation (Fig. 6).

Similarly to other objects detection algorithms, morphological operations should be performed on the image in order to remove unnecessary details or noise.

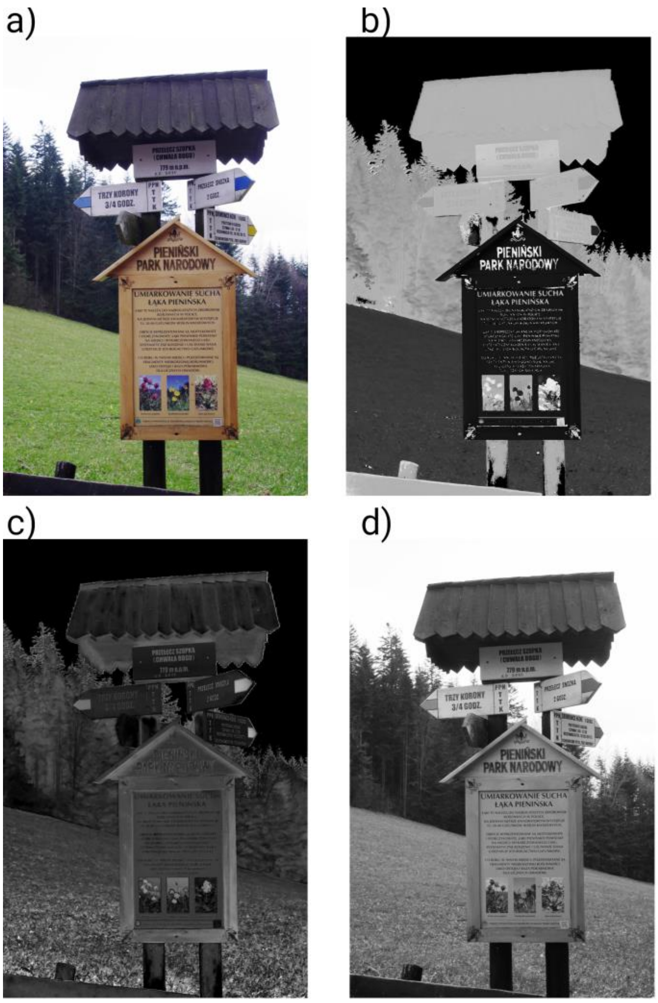

Fig. 5.The representation of image using HSL color space: a) initial image, b) hue channel, c)saturation channel, d) lightness channel which is most often the same as grayscale image representation.

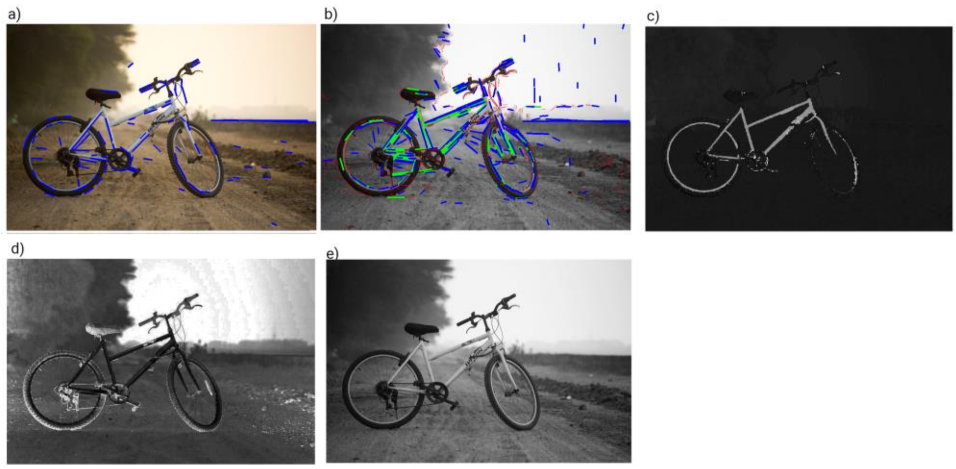

Fig. 6.The example line segment detection: a) version based on shapes detecting using RGB color space (previous approach), b) version based on detecting using HSL color space (new approach),c) hue channel, d) saturation channel, e) lightness channel.

The first algorithm step is the detection of base primitives- line segments and arches. Firstly all line segments are detected in hue channel using Line Segment Detector algorithm. Next, for each channel thresholding with different values is performed in order to create uniform areas of similar pixels and reduce the slight differences on the same shapes. After that, line segment detection is performed again for each channel. Thanks to that more line segments are detected. Next all detected line segments are merged if possible, checking if they have the same line slope and connection of very near endings (Fig. 7). As a next step, all segments shorter than defined threshold are removed from the list. The algorithm is shown in the Alg. 1.

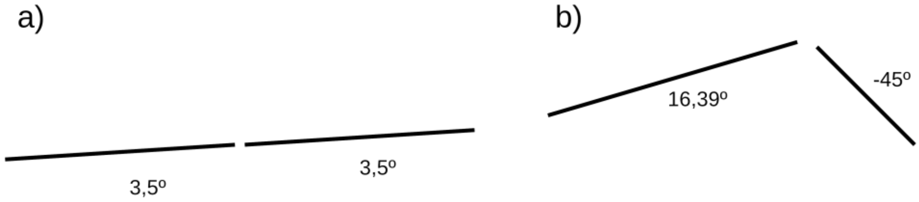

Fig. 7.The example of line segments merging with close endings: a) two segments with line slopes close to 3.5◦ which can be merged, b) two segments with completely different line slopes.

Algorithm 1 Line segment detection algorithm

Ensure: *img* - the image which has to be processed, *lsList* - list which stores line segments

imgH ← hue channel of *img*;
detect all line segments in *imgh* and add them to the *lsList*
for each *imgx* ∈ H,L,S channel of *img* **do**
process img_x as follows:

for each *pixel* of img_x **do**
round *pixel value* to the nearest *threshold* value
end for
detect all line segments in img_x and add them to the *lsList*
end for
for each $ls1 \in lsList$ **do**
for each $ls_2 \in \{\{lsList\} \setminus \{ls_1\}\}$ **do**
if ls_1 and ls_2 endings are close enough **then**
$angle_{LS1} \leftarrow ls_1$ *line slope*
$angle_{LS2} \leftarrow ls_2$ *line slope*
if $angle_{LS1}$ and $angle_{LS2}$ are similar enough **then**
merge ls_1 and ls_2
end if
end if
end for
end for
for each $ls \in lsList$ **do**
$len \leftarrow length(ls)$
end for

When all line segments are found, then the list is searched in order to find the candidates of arches. In order to find line segments which may construct an arc, we check if in the list there is a chain of connected line segments with length equal or greater than 3. Then, if a chain is found, angles between each segments are checked if they are equal or very similar. Moreover a test if their values are in the range $(0^\circ, 180^\circ)$ (Fig. 8) is performed. Additionally, lengths of segments are checked in order to detect if they are very similar. The arc detection algorithm is shown in the list is processed: firstly chains of arches are detected and then looped ones are transformed into arc-sided polygons and all others into polyarches. Alg. 2. In practical implementation, the algorithm is assisted with Circular Hough Transform in order to improve the detection of circles. The last shapes extraction algorithm step is the creation of more complex primitives defined as follows: polylines, polygons, polyarches and arc-sided polygons. Firstly all detected line segments are processed and checked for chains. After chains detection, they are tested in order to detect looped chains from which polygons are created. Then, all others are transformed into polylines. Similarly, founded arches list is processed: firstly chains of arches are detected and then looped ones are transformed into arc-sided polygons and all others into polyarches.

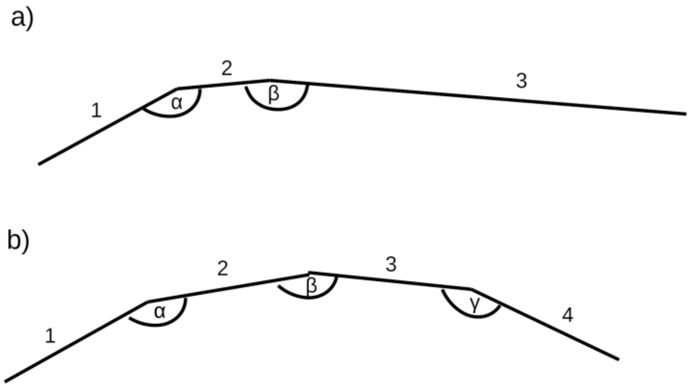

Fig. 8.Different chains of line segments: a) the chain from which an arc cannot be created: segments lengths are different (1=138.7, 2=85.4, 3=371) and angles are not equal or very close (α = 157˚, β = 170˚), b) a chain from which an arc may be created: segments lengths are very similar (1=146, 2=145, 3=146,8, 4=145,8) and angles α,β,γ are equal to 160˚.

Algorithm 2 Arches detection algorithm

Ensure: *lsList* - list which stores line segments; *chains* - list of chains of line segments; *archesList* - list which stores arches

find all connected line segments from *lsList* longer than 3 segments and add them to *chains*
compare lengths of all segments in each *chain* ∈ *chains*
if lengths are similar enough **then**
  for each *ls* ∈ *chain* **do**
    compute an angle between *ls* and next segment from the same chain
  end for
  if angles are similar enough **then**
    create new arc based on chain
    compute arc angle and center point
    add arc to *archesList* and remove all *chain*’s line segments from *lsList*
  end if
end if

6. Shapes extraction improvement using Edge Boxes

The shapes extraction algorithm performance could be improved by a usage of object proposal algorithms. One of the most suitable algorithm for our approach is the Edge Boxes [Zitnick and Dollár, 2014]. It is based on the idea that when there are many contours enclosed by bounding box, there is a high likelihood that it contains an object. As an edge extractor a Structure Edge detector is used. All contours are examined using a sliding window approach. In order to improve the computation time, efficient data structures are used. The result of the algorithm is a set of bounding box objects proposals.

In this paper we propose to firstly extract all shapes present in the image and then to remove all of them, which are not fully enclosed by the object box. Thanks to that, for example all unnecessary line segment which are not connected with the object are not added to the graph. For some cases, when the object's box is not detected precisely, some correct shapes may be removed, but this problem will be examined as a future research.

7. The database structure

The database structure is based on a tree which stores similar object's graphs in the same part of a subtree. Two types of tree nodes are defined:

- common graph nodes
- data nodes

The common graph nodes are used in order to store the common parts of graphs which are stored in its children. For example, if there are two children with a car and a bicycle graphs, as a common graph, wheels are used. The common graph nodes does not store any image information and are only used to check if a whole subtree should be tested or abandoned during the query.

The data nodes are used to store graphs of objects, images and metadata connected with them. In order to reduce the height of the tree, all similar graphs are stored in the same node in a structure called a slice. A slice is a vector of very similar graphs where the first element is the most similar to the parent common node graph and the last one the least. There may be more than one slices in the data node in order to provide the ability to process queries in parallel.

The example simplified tree structure is shown in the Fig. 9. The database contains a car, a scooter and bicycles graphs. There are five common graph nodes and three data nodes which store image graphs, images and metadata. The tree database structure and operations which may be performed (adding, deleting and querying graphs) were described more detailed in [Deniziak and Michno, 2017].

8. Experimental results

In order to examine the proposed approach, an experimental application was written in C++ language for database implementation and python language with Django, HTML and JavaScript for GUI. The database stores images of three types of objects: cars, bicycles and motorbikes. Firstly the computation of primitives attributes in relation to the object oriented bounding boxes was examined. In order to compare the results, two commonly used coefficients were used:

$$precision = \frac{number of relevant results images}{total number of results images}. \tag{1}$$

$$recall = \frac{number of relevant results images}{total number of relevant images \in the database}. \tag{2}$$

The experimental results are presented in the Table 1. As can be seen, when line slopes values are computed in relation to the longest side of the object-oriented bounding box, both precision and recall reach much higher values in comparison to the previous version where slopes are computed in relation to the X axis. It can be noted that even if there are used the same images but with rotations, there are still some differences in precision and recall values for them. This problem is caused i.e. by inaccuracies during shapes extraction stage and will be taken into consideration in the future research.

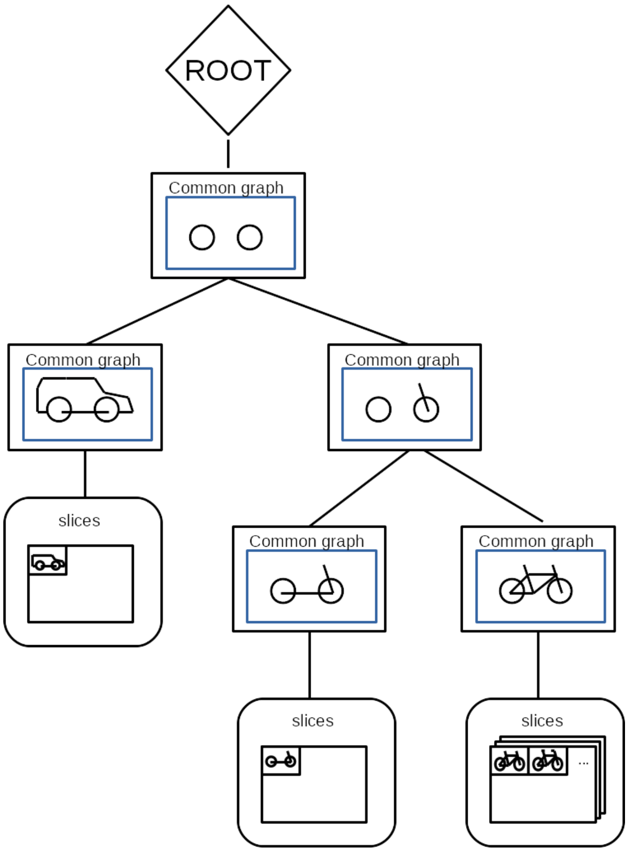

Fig. 9.The overview of the proposed tree database structure [Deniziak and Michno, 2017].

Table 1. The comparison of precision and recall results for bicycle images (normal, aligned to the X-axis and slightly rotated) for computing line slopes values in relation to the X-axis and to the longest side of the object oriented bounding box (OOB).

object	**TOP 10 precision**		**Recall**	
	X-axis	**OOB**	**X-axis**	**OOB**
normal	0.5	0.9	0.26	0.63
rotated	0.3	0.7	0.16	0.42

Table 2. The comparison of precision and recall results for bicycle images (normal, aligned to the X-axis and slightly rotated) for computing line slopes values in relation to the X-axis and to the longest side of the object oriented bounding box (OOB).

object	**TOP 10 precision**		**Recall**	
	previous	**mutual positions**	**previous**	**mutual positions**
Bicycle normal	0.5	0.6	0.26	0.79
Bicycle rotated	0.3	0.4	0.16	0.95
Car	0.8	0.6	1	1

Another tests were performed in order to evaluate the usage of the additional information about mutual positions of all nodes during queries. The results are presented in the Table 2. The experiments does not prove that adding such an information improves the precision of the query results. For the bicycles images the precision values are only slightly higher, but for the car it is lower than in previous version. Moreover, the computation time was much longer. Contrary, for bicycle images, the recall values are much higher than in previous version. Due to that fact, the usage of additional information about mutual positions of all nodes in a graph is doubtful and more tests should be performed.

Another types of experiments were performed in order to evaluate the shapes detection improvement by a usage of HSL color space. The results are presented in the Table 3. There can be noted that for all objects the usage of HSL color space increased the number of detected primitives. The recognition of the third arc in a bicycle is not a mistake, because not only wheels were detected, but also a gearwheel. In the motorbike image also other parts were detected as circles correctly. It can be seen that the number of polylines is much higher when using HSL color space. This is caused by higher number of detected base line segments and more precise detection which allowed to create more complex primitives.

Moreover, additional tests were performed in order to evaluate the proposed method performance in comparison with other CBIR algorithms such as: a simple region-based algorithm, a sketch-based algorithm described by Kato et al. [Kato et al., 1992] and simplified sketch contour-based algorithm based on [Parui and Mittal, 2015]. The experimental results for the example images (Fig. 10) are shown in the Table 4. It can be seen that the precision coefficient values for the proposed method are much higher than for other algorithms. For the region-based method bicycle objects were problematic due to the small area of uniform colors. For the sketch-based methods the highest precision values were obtained for cars objects. Lower values were achieved for the scooter and

Table 3. The comparison of number of detected primitives for previous version (RGB) and version which uses HSL color space.

primitive	Bike		car		motor	
	RGB	HSL	RGB	HSL	RGB	HSL
line segments	24	36	73	54	64	71
arches	2	3	2	2	2	3
polylines	0	3	8	11	2	10

motorbikes classes due to their similarity between each others and to the bicycle object class. The recall coefficients for some objects were lower for the proposed method than for the other algorithms. It is caused by lower precision values obtained for the region-based, sketch-based [Kato et al., 1992] and contour-based methods.

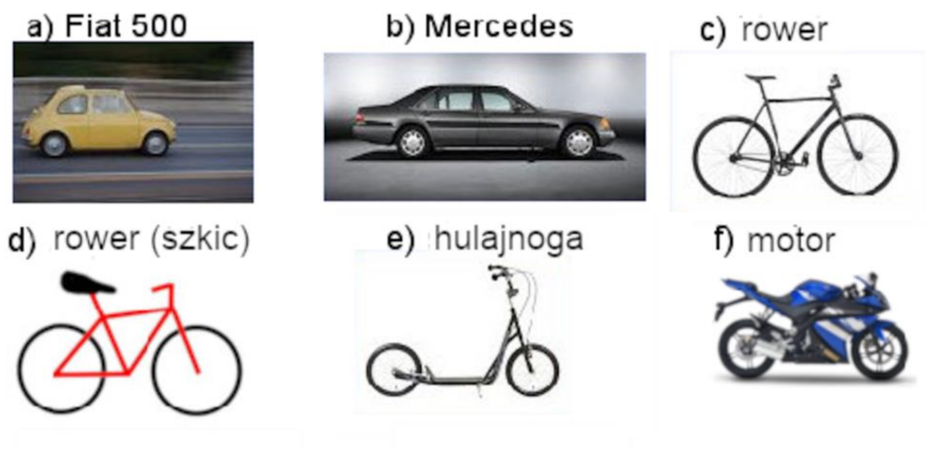

Fig. 10.Example objects used for comparison test with other methods (region-based, sketch-based[Kato et al., 1992] and contour-based.

Table 4. Example results for the comparison with other methods.

object	Query by Approximate Shapes		Region-based		Kato et al. [Kato et al., 1992]		contour-based	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
car(Fiat 500)	0.89	0.33	0.53	0.75	0.58	0.83	0.46	0.87
Car (Mercedes Benz)	0.79	0.73	0.51	0.5	0.47	0.7	0.43	0.78
bicycle	0.93	0.37	0.23	0.42	0.4	0.5	0.37	0.64
bicycle (sketch)	1.0	0.60	0.28	0.47	0.39	0.73	0.32	0.64
motorbike	0.86	0.40	0.75	0.4	0.25	0.5	0.05	0.67
scooter	0.67	1.0	0.21	1.0	0.22	0.4	0.11	1.0

9. Conclusions and future works

This paper presented a new sketch-based Content Based Image Retrieval method which is based on a new object representation. Each object is approximated by a set of simple shapes, called primitives: line segments, polylines, polygons, arches, polyarches and arc-sided polygons. Since the information about mutual positions and connections are important, all primitives with such an information, are stored as a graph.

In order to improve the shapes extraction algorithm, we propose to use the HSL color space, which provided more precise results. Furthermore, to increase the precision of the results, we proposed also to use computation of the line slope in relation to the longest side of the object oriented bounding box.

Performed experiments showed that the usage of HSL color space, Edge Boxes and Object Oriented Bounding Boxes increased the precision and recall coefficients values. The usage of the information about mutual positions between all primitives is doubtful and more tests should be performed.

The future improvement of the proposed method includes preforming tests using higher number of images and improving the recall coefficient value. Moreover, different database structure implementations should be done (e.g. using NoSQL data stores, such as SD2DS or SQL relational databases). Additionally other graph comparison algorithms may be evaluated, e.g. using the optimization methods with constraints [Sitek and Wikarek, 2016]. Also, the shape extraction algorithm may be improved using e.g. approaches proposed in [Woźniak and Połap, 2018]. Moreover, additional work on the Edge Boxes algorithm should be done.

References

Deniziak, S. and Michno, T. (2016). Content based image retrieval using query by approximate shape. In 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), pages 807–816.

Deniziak, S. and Michno, T. (2017). New content based image retrieval database structure using query by approximate shapes. In 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), pages 613–621.

Evgeniou, T., Pontil, M., Papageorgiou, C., and Poggio, T. (2003). Image representations and feature selection for multimedia database search. IEEE Transactions on Knowledge and Data Engineering, 15(4):911–920.

Kato, T., Kurita, T., Otsu, N., and Hirata, K. (1992). A sketch retrieval method for full color image database-query by visual example. In 11th IAPR International Conference on Pattern Recognition, Vol.I. Conference A: Computer Vision and Applications, pages 530–533.

Krähenbühl, P. and Koltun, V. (2014). Geodesic object proposals. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, Computer Vision – ECCV 2014, pages 725–739, Cham. Springer International Publishing.

Kriegel, H. P., Kroger, P., Kunath, P., and Pryakhin, A. (2006). Effective similarity search in multimedia databases using multiple representations. In 2006 12th International Multi-Media Modelling Conference, pages 4 pp.–.

Li, C. Y. and Hsu, C. T. (2008). Image retrieval with relevance feedback based on graph-theoretic region correspondence estimation. IEEE Transactions on Multimedia, 10(3):447–456.

Mocofan, M., Ermalai, I., Bucos, M., Onita, M., and Dragulescu, B. (2011). Supervised tree content based search algorithm for multimedia image databases. In 2011 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pages 469–472.

Müller, S., Eickeler, S., and Rigoll, G. (1999). Multimedia database retrieval using handdrawn sketches. In Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, 20-22 September, 1999, Bangalore, India, pages 289–292.

Parui, S. and Mittal, A. (2015). Sketch-based image retrieval from millions of images under rotation, translation and scale variations. CoRR, abs/1511.00099.

Singh, A., Shekhar, S., and Jalal, A. (2012). Semantic based image retrieval using multiagent model by searching and filtering replicated web images. In Information and Communication Technologies (WICT), 2012 World Congress on, pages 817–821.

Sitek, P. and Wikarek, J. (2016). A hybrid programming framework for modeling and solving constraint satisfaction and optimization problems. Scientific Programming, 2016.

Wang, H. H., Mohamad, D., and Ismail, N. A. (2010). Approaches, challenges and future direction of image retrieval. CoRR, abs/1006.4568.

Wang, X., Duan, X., and Bai, X. (2016). Deep sketch feature for cross-domain image retrieval. Neurocomput., 207(C):387–397.

Woźniak, M. and Połap, D. (2018). Adaptive neuro-heuristic hybrid model for fruit peel defects detection. Neural Networks, 98:16 – 33.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, Computer Vision – ECCV 2014, pages 391–405, Cham. Springer International Publishing.