Query techniques for CBIR

Tatiana Jaworska

Systems Research Institute, Polish Academy of Sciences,
6 Newelska Street, Warsaw, Poland
Tatiana.Jaworska@ibspan.waw.pl

Abstract. One of the fundamental functionalities of a content-based image retrieval system (CBIR) is answering user queries. The survey of query types and examples of systems using these particular queries are presented here. For our CBIR, we prepared the dedicated GUI to construct user designed query (UDQ). We outline the new search engine which matches images using both local and global image features for a query composed by the user. In our case, the spatial object location is the global feature. Our matching results take into account the kind and number of objects, their spatial layout and object feature vectors. Finally, we compare our matching results to some other search engines.

Keywords: query, content-based image retrieval, GUI, composed query, search engine.

1. Introduction

1.1. Query concept overview

One of the fundamental functionalities of a content-based image retrieval system (CBIR) is answering user queries. In traditional, text retrieval system of queries has been highly developed, whereas for the image retrieval a content query has not come up to users' expectations. It stems from the fact that content-based searches have important distinctions compared to traditional searches. The limitations of these systems include both the image representations they use and their methods of accessing those representations to find images. Systems based on keyword querying are often non-intuitive and offer little help in understanding why certain images were returned and how to refine the query.

Independently of a diversity of methods focused on the retrieval, in particular, search by association, search for a specific image, or category search [25] we can generally divide query methods into (see Fig. 1):

1. Query by keywords [36],
2. Query by example [2, 23],
3. Query by canvas [18, 19],
4. Query by sketches [16],

5. Query by spatial icons [17],
6. Designed query for semantic retrieval [6, 12].

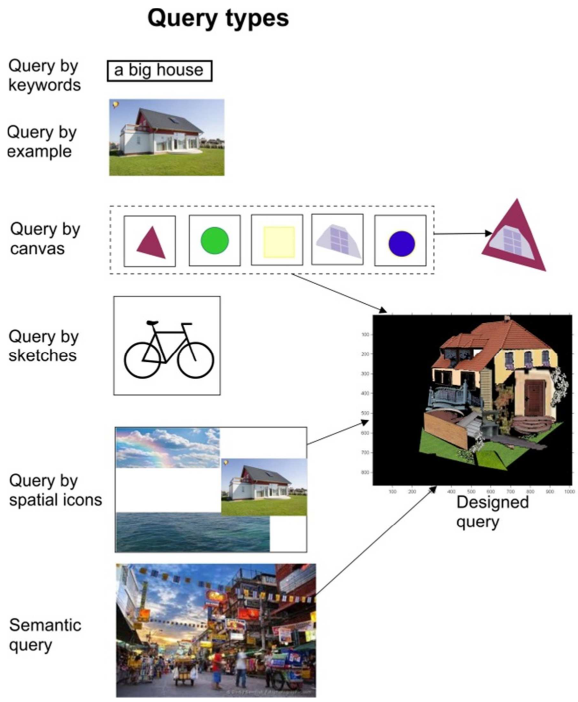


Fig. 1. Query types**.**

Query by keywords: The first method for asking the database a question was using keywords as an analog to queries in alpha-numeric DBs. This method requires text annotations for images collected in DB. It is still used in WWW image search engines but it is notorious for incomplete, inconsistent, context sensitive and the ambiguity of meaning of the keywords. In this case [36], the keyword ambiguity is expanded to the selected reference classes most relevant to the query keyword. For example, the keyword 'apple' can mean: 'apple fruit', 'apple computer', 'apple logo', or 'apple tree' from which reference classes are selected. These attempts try to fill in the semantic gap that exists between the description of an image and the image itself.

Query by example: At present, most systems use query by example (QBE) whose major advantage is the capability to determine a set of attributes or features that describe the contents of the user's desired image [18, 23]. In a nutshell, QBE provides a full image for the search engine, but its drawback is the fact that the user first has to find an image which he wants to use as a query. In some situations, the most difficult task is to find this one proper image which the user keeps in mind to feed it to the system as a query by example.

Query by canvas allows the user to compose a visual query using geometrical shapes, colours, and textures. This approach inherently tends to specify objects of interest in an indirect way using primitive features [19]. Moreover, the similarity matching between query and images relies on effective pre-segmentation of regions in the images, which is generally complex and difficult [18].

Query by sketches enables the user to draw the shape of an object as query but is not popular, perhaps because most users are rather poor at graphic design [23]. For this reason applications have used a query by sketches in a limited form only to images of dominant objects in a uniform background.

Query by spatial icons is represented by the visual icons with spatial constraints. It specifies a query using a higher-level visual semantics representation. A query is composed as a spatial arrangement of image segments [17]. The visual query term specifies the region where a salient image region should appear, and a query forming chains of these terms using logical operators. The spatial constraint for regions defines the location and size of the specified visual semantic segments as drawn on a canvas.

Semantic query contains the associated semantic concept even though is visually dissimilar [21, 36]. Hence, the multiple semantic interpretation results in a retrieval problem because it needs to take into account simultaneously: low-level features, object layout and human associations which an image evokes. For this reason, recently systems have appeared offering designed queries to the user [6, 12].

1.2. Previous works

The data structure and type of query imply the kind of image information that is retrieved. A CBIR, in turn, should meet the user's diverse requirements depending on the interest domain and a particular need.

Chronologically speaking, the first CBIR systems appeared in the early 90's and used queries by keyword which necessitated the image DBs with annotations. These annotations were more or less relevant to the image context and needed human work to prepare them [29]. Later, QBE appeared which has been the most popular manner used for an image query up to now. Approximately in the late 90's a highly interactive refinement of the search using the sketches or composing images from segments was offered by the system. These three kinds of queries share one disadvantage: it is difficult for the user to prepare, find or design an *ideal* query which they have in mind.

About 10 years later interactive techniques based on feedback information from the user, commonly known as relevance feedback (RF) appeared [31] which, step by step, developed into techniques based on local information from the top retrieved results, commonly known as local feedback or collaborative image retrieval (CIR) [38] which is a powerful tool to narrow down the semantic gap between a low- and high-level retrieval concept [1].

Recently CBIR systems have appeared which enable the user to compose a query from different, previously segmented objects, for instance, a query for face retrieval [37] or a query from composed single images and segments of video frames [6].

In this paper we propose a special, dedicated user's GUI which enables the user to compose their ideal image from the image segments. The data structure and the layout of the GUI reflect the manner of our search engine work. In this paper we present, as a main contribution, the new search engine which takes into account the kind and number of objects, their features, together with different spatial location of the segmented objects in the image.

In order to help the user create the query which they have in mind a special GUI has been prepared to formulate composed queries. An additional contribution is the empirical studies for the proposed search engine consisting in benchmarking it against other known engines.

2. User designed query concept for semantic retrieval

Fauqueur and Boujemaa [6] offered the system combining Boolean queries through the region's photometric thesaurus to specify the types of regions which are present and absent in the user's mental/imagined image. The very simple user interaction enables them to combine sophisticated Boolean composition queries with the range query mechanism to adjust the precision of the visual search.

Whereas a front-end module of our CBIR provides a graphical editor which enables the user to compose the image which he/she has in mind from the previously segmented objects. We give the user more tools to design their query than our predecessors.

It is a bitmap editor which allows for the selection of linear prompts in the form of contour sketches (see Fig. 2). These sketches have been generated from images existing in the DB. Next, from the list of object classes the user can select elements to prepare a rough sketch of an imaginary landscape. There are many editing tools available, for instance:

- creating masks to cut off the redundant fragments of a bitmap (see Fig. 3 a));
- changing the bitmap colour (see Fig. 3 c) and Fig. 3 d));
- performing basic geometrical transformation, such as: translation, scale, rotation and shear;
- duplicating of repeating fragments;
- reordering bitmaps forward or backward.

This GUI is a prototype, so it is not as well-developed as commercial programs, e.g. CorelDraw. Nevertheless, generally, the user can design an image consisting of as many elements as they need. The only restriction is the number of classes introduced in the DB. At the moment, there are 40 classes but there is no limit for them. Later, the user designed query (UDQ) is sent to the search engine and is matched according to the rules described in sec. 3. However, in the absence of

UDQ, the search engine can work with a query consisting of a full image downloaded, for example, from the Internet.

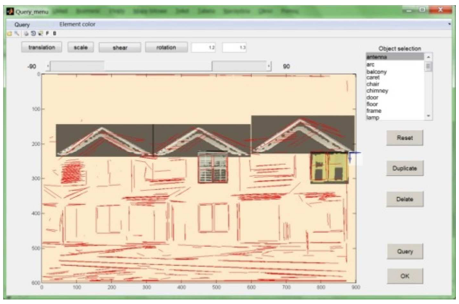


Fig. 2. The main GUI window. An early stage of a terraced house query construction.

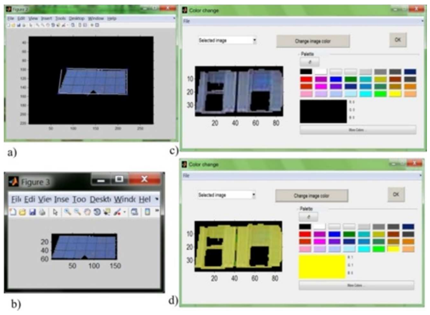


Fig. 3. Main components of the GUI. We can draw a contour of the bitmap (see a) and b)) and change the colour of an element c) and d).

3. CBIR structure

In general, our system consists of four main blocks (Fig. 4): the image preprocessing block [10], the Oracle Database [11], the search engine [13] and the graphical user's interface (GUI). All modules, except the Oracle DBMS, are implemented in Matlab.

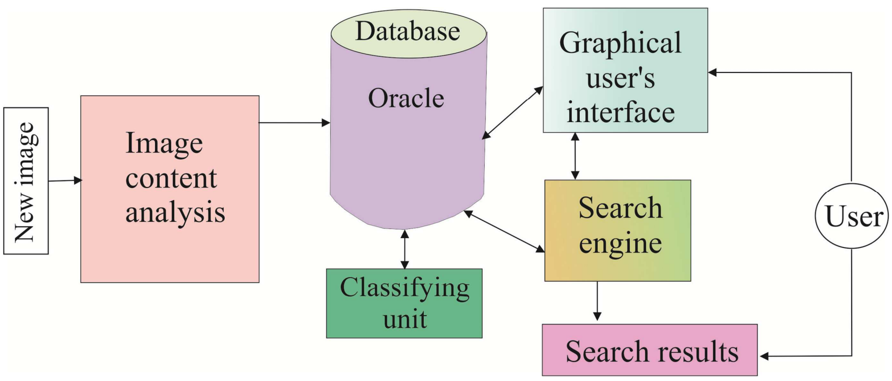


Fig. 4. Block diagram of our content-based image retrieval system.

3.1 Graphical data representation and object classification

A classical approach to CBIR comprises image feature extraction [30, 39]. Similarly, in our system, at the beginning, the new image (e.g. downloaded from the Internet) is segmented, creating a collection of objects. Each object, selected according to the algorithm presented in detail in [13], is described by some low-level features f_i. We collect $r = 45$ features for each graphical object, for which we construct a feature vector $\mathbf{O} = \{f_1, f_2, \dots, f_r\}$.

Next, the feature vector **O** is used for object classification. We have to classify objects in order to use them in a spatial object location algorithm and to offer the user a classified group of objects for the semantic selection. So far, four classifiers have been implemented and they are mutually used in our:

- a comparison of features of the classified object with a class pattern;
- decision trees [20, 7];
- the Naïve Bayes classifier [4, 22];
- a fuzzy rule-based classifier (FRBC) [9, 15].

The FRBC is used in order to identify the most ambiguous objects which means these assigned to different classes according the three first classifiers. According to Ishibuchi, the FRBC decides which of the three classes a new element belongs to [14].

3.2 Spatial object location

Thanks to taking into account a spatial object location the gap between low-level and high-level features in CBIR has diminished. To describe spatial layout of objects, different methods have been introduced, for example: the spatial pyramid representation in a fixed grid [24], others used the spatial arrangements of regions [28], or the object's spatial orientation relationship [40]. In some approaches, image matching is proposed directly, based on spatial constraints between image regions [35].

Here, spatial object location in an image is used as the global feature [15]. The objects' mutual spatial relationship is calculated based on the centroid locations and angles between vectors connecting them, with an algorithm proposed by Chang and Wu [3] and later modified by Guru and Punitha [8], to determine the first principal component vectors (PCVs).

3.3 Search engine construction

Now, we will describe how the similarity between two images is determined and used to answer a query. Let a query be an image I_q, such as $I_q = \{o_{q1}, o_{q2}, \dots, o_{qn}\}$, where o_{ij} are objects. An image in the database is denoted as I_b, $I_b = \{o_{b1}, o_{b2}, \dots, o_{bm}\}$. Let us assume that there are, in total, M classes of the objects recognized in the database, denoted by labels $L_1, L_2, \dots, L_M$. Then, by the image signature I_i we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \dots, \text{nobc}_{iM}] \tag{1}$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i, i.e. such objects o_{ij}.

In order to answer the query I_q, we compare it with each image I_b from the database. First of all, we determine a similarity measure sim_{sgn} between the signatures of query I_q and image I_b:

$$\text{sim}_{\text{sgn}}(I_q, I_b) = \sum_i (\text{nob}_{qi} - \text{nob}_{bi}) \tag{2}$$

computing as the modified Hamming distance between two vectors of their signatures (cf. (1)), such that $\text{sim}_{\text{sgn}} \geq 0$ and $\max_i (\text{nob}_{qi} - \text{nob}_{bi}) \leq \text{tr}$, tr is a limit of element number of a particular class which I_q and I_b can differ. It means that we prefer images composed of the same classes as the query. Similarity (2) is non-symmetric because if classes in the query are missing from the image the components of (2) can be negative.

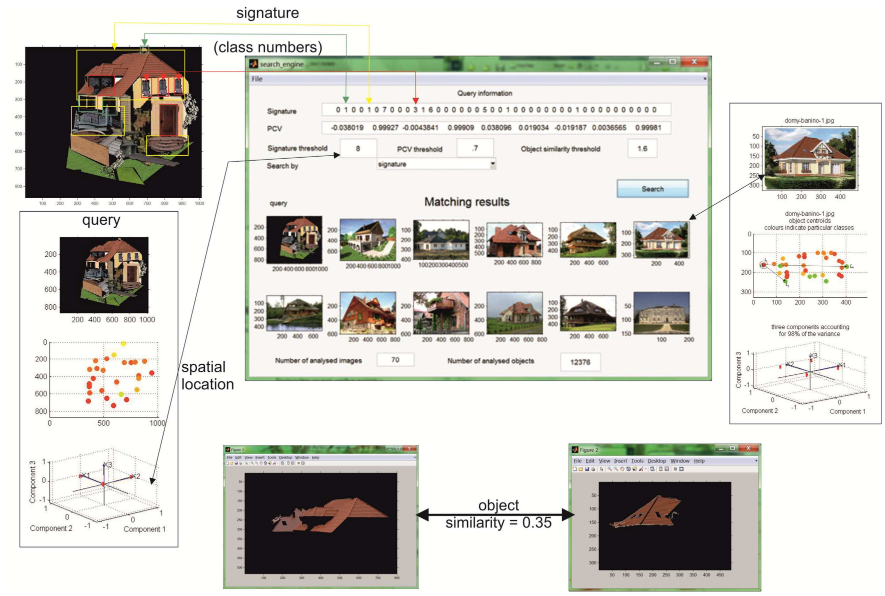


Fig. 5. A graphic concept scheme of our image search engine.

Otherwise, we proceed to the next step and we find the spatial similarity sim_{PCV} of images I_q and I_b computing the Euclidean, City block or Mahalanobis distance between their PCVs as:

$$\text{sim}_{\text{PCV}}(I_q, I_b) = 1 - \sqrt{\sum_{i=1}^{3} (PCV_{bi} - PCV_{qi})^2} \tag{3}$$

If the similarity (3) is smaller than the threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q.

Next, we proceed to the final step, namely, we compare the similarity of the objects representing both images I_q and I_b. For each object o_{qi} present in the representation of the query I_q, we find the most similar object o_{bj} of the same class, sim_{ob} (o_{qi} , o_b) based on the comparison of feature vectors **O** computing the Euclidean distance between o_{qi} and o_b. If there is no object o_{bj} of the class L_{qi}, then sim_{ob} (o_{qi} , o_b) = 0.

The process of searching highly similar objects is realized according to the Hungarian algorithm for the assignment problem implemented by Munkres. Thus, we obtain the vector of similarities between query I_q and image I_b:

$$\text{sim}(I_q, I_b) = \begin{bmatrix} \text{sim}_{\text{ob}}(o_{q1}, o_{b1}) \\ \vdots \\ \text{sim}_{\text{ob}}(o_{qn}, o_{bn}) \end{bmatrix} \tag{4}$$

where n is the number of objects present in the representation of I_q. In order to compare images I_b with the query I_q, we compute the sum of sim_{ob} (o_{qi}, o_b) and then use the natural order of the numbers. Therefore, the image I_b is listed as the first in the answer to the query I_q, for which the sum of similarities is the highest.

Fig. 5 presents the main elements of the search engine interface with reference images which are present in the CBIR system. The main (middle) window displays the query signature and PCV, and below the user is able to set threshold values for the signature, PCV and object similarity. The lower half of the window is dedicated to matching results. In the top left of the figure we can see a user designed query comprising elements whose numbers are listed in the signature line. Below the query there is a box with the query miniature, a graph showing the centroids of query components and further below there is a graph with the PCV components. In the bottom centre windows there are two elements of the same class (e.g. a roof) and we calculate their similarity. On the right side there is a box which is an example of PCA for an image from the DB. The user introduces thresholds to calculate each kind of similarity.

4. Results

In this section, we describe experiments conducted on the colour images generated with the help of the UDQ or full images taken from our DB and we will compare our results with the Google image search engine. All images are in the JPG format but in different sizes. In all tables images are ranked according to decreasing similarity determined by our system.

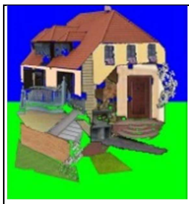
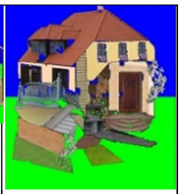
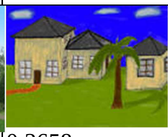
Only in order to roughly compare our system's answer to the query, we used SSIM (Universal image similarity index) proposed by Wang and Bovik [34], being aware that it is not fully adequate to present our search engine ranking. SSIM is based on the computation of three components, namely the luminance, contrast and structural component, which are relatively independent. In case of a big difference of images the components can be negative which may results in a negative index.

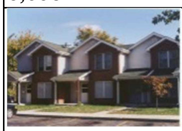
A query is generated with the help of the UDQ interface and its size depends on the user's decision as well as the number of objects. The search engine displays a maximum of 11 best matched images from the DB. Although the user designed a query including few details (see tab. 1 query 1 and 2) the search results are quite acceptable.

Applying the UDQ is not obligatory. The user can choose their QBE from among the images of the DB if they find it suitable for their aim. Then the matching results are presented in tab. 1 (column 3).

We also decided to compare our results with the Google image search engine because our system compared with some academic search systems proved significantly more effective. The results obtained are presented in tab. 1 (columns 4 and 5).

Table 1. The matching results and the SSIM (column 1) for UDQ when PCV similarity is calculated based on the City block distance (for thresholds: signature = 17, PCV = 3.5, object = 0.9), (column 2) matches for UDQ when PCV similarity is calculated based on the City block distance (for thresholds: signature = 20, PCV = 4, object = 0.9), (column 3) matches for QBE when PCV similarity is calculated based on the Euclidean distance, (columns 4 and 5) matches for the Google image search engine.

 query1	 query2	 query3	 query2	 query1
 0,1492	 0.1745	 0.2519	 0.3658	 0.0821
 0,1571	 0.1399	 0.2175	 0.0939	 0.1054

 0,1099	 0.0571	 0.1276	 0.0232	 0.1765
 0,1525	 0.1443	 0.3129	 0.0240	 0.2666
 0,1346	 0.1505	 0.2908	 0.3174	 0.1076
 0,0542	 0.0012	 0.1002	 0.2056	 0.0876
 0,0062	 -0.0378	 -0.0255	 0.1095	 0.2089
 0,0419	 0.0642	 0.2888	 0.1807	 0.1267
 0,1497	 0.2009	 0.2366		
 0,1154	 0.0833	 0.0151		
	 0.2221	 0.0738		

In order to better visualise the obtained results, we compare only the SSIMs from tab. 1 as a bar chart (see Fig. 6). There is one extra bar for query1 representing the matches when PCV similarity is calculated based on the Euclidean distance. The other bars refer to the respective columns in tab. 1 according to the legend.

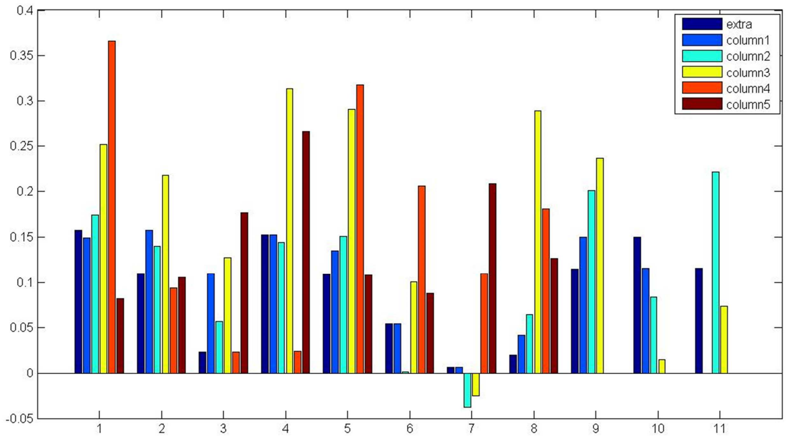


Fig. 6. The comparison of SSIMs for the above-presented results from tab. 1.

5. Conclusions

We built and described a new image retrieval method based on a three-level search engine. The underlying idea is to mine and interpret the information from the user's interaction in order to understand the user's needs by offering them the GUI. A user-centred, work-task oriented evaluation process demonstrated the value of our technique by comparing it to a traditional CBIR.

Intensive computational experiments are under way in order to determine the optimal choice of model parameters. Furthermore, the results of this study have to be verified in a larger scale evaluation, involving long-term usage of the system for real day-to-day user tasks. However, the preliminary results we have obtained so far, using the simplest configuration, are quite promising.

As for the prospects for future work, the implementation of an on-line version should test the feasibility and effectiveness of our approach. Only experiments on large scale data can verify our strategy. Additionally, a new image similarity index should be prepared to evaluate semantic matches.

References

1. Azimi-Sadjadi, M. R., Salazar, J. & Srinivasan, S., 2009. An Adaptable Image Retrieval System With Relevance Feedback Using Kernel Machines and Selective Sampling. *IEEE Transactions on Image Processing*, 18(7), pp. 1645-1659.

2. Carson, C., Belongie, S., Greenspan, H. & Malik, J., 2002. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Aug, 24(8), pp. 1026-1038.
3. Chang, C.-C. & Wu, T.-C., 1995. An exact match retrieval scheme based upon principal component analysis. *Pattern Recognition Letters*, Volume 16, pp. 465-470.
4. Comer, D., 1979. Ubiquitous B-Tree. *ACM Computing Surveys*, 11(2), pp. 21-137.
5. Faloutsos, C. et al., 1994. Efficient and Effective Querying by Image Content.. *Journal of Intelligent Information Systems*, Volume 3, pp. 231-262.
6. Fauqueur, J. & Boujemaa, N., 2006. Mental image search by boolean composition of region categories. *Multimed Tools and Applications*, Volume 31, pp. 95–117.
7. Fayyad, U. M. & Irani, K. B., 1992. *The attribute selection problem in decision tree generation.* s.l., s.n., pp. 104-110.
8. Guru, D. S. & Punitha, P., 2004. An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis. *Pattern Recognition Letters* , Volume 25, pp. 73–86.
9. Ishibuchi, H. & Nojima, Y., June 27-39, 2011. *Toward Quantitative Definition of Explanation Ability of Fuzzy Rule-Based Classifiers.* Taipei, Taiwan, IEEE Society, pp. 549-556.
10. Jaworska, T., 2007. Object extraction as a basic process for content-based image retrieval (CBIR) system.. *Opto-Electronics Review*, December, 15(4), pp. 184-195.
11. Jaworska, T., 2008. *Database as a Crucial Element for CBIR Systems.* Beijing, China, World Publishing Corporation, pp. 1983-1986.
12. Jaworska, T., 2009. Multi-criteria object indexing and graphical user query as an aspect of content-based image retrieval system.. In: L. Borzemski, A. Grzech, J. Świątek & Z. Wilimowska, eds. *Information Systems Architecture and Technology.* Wrocław: Wrocław Technical University Publisher, pp. 103-112.
13. Jaworska, T., 2011. A Search-Engine Concept Based on Multi-Feature Vectors and Spatial Relationship. In: H. Christiansen, et al. eds. *Flexible Query Answering Systems.* Ghent: Springer, pp. 137-148.
14. Jaworska, T., 2014. *Application of Fuzzy Rule-Based Classifier to CBIR in comparison with other classifiers.* Xiamen, China, IEEE, pp. 1-6.
15. Jaworska, T., 2014. Spatial representation of object location for image matching in CBIR. In: A. Zgrzywa, K. Choroś & A. Siemiński, eds. *New Research in Multimedia and Internet Systems.* Wrocław: Springer, pp. 25-34.
16. Lee, Y. J., Zitnick, . L. C. & Cohen, M. F., 2011. Shadowdraw: real-time user guidance for freehand drawing.. *ACM Transactions on Graphics (TOG),*, Volume 30, pp. 27.
17. Lim, J.-H. & Jin, J. S., 2005. A structured learning framework for content-based image indexing and visual query. *Multimedia Systems*, Volume 10, p. 317–331.
18. Niblack, W. et al., 1993. The QBIC Project: Querying Images by Content Using Colour, Texture and Shape. *SPIE*, Volume 1908, pp. 173-187.
19. Ogle, V. E. & Stonebraker, M., 1995. CHABOT: Retrieval from a Relational Database of Images. *IEEE Computer*, September, 28(9), pp. 40-48.
20. Quinlan, J. R., 1986. Induction of Decision Trees. *Machine Learning*, Volume 1, pp. 81-106.
21. Rasiwasia, N., Moreno, P. J. & Vasconcelos, N., 2007. Bridging the Gap: Query by Semantic Example. *IEEE TRANSACTIONS ON MULTIMEDIA*, Aug, 9(5), pp. 923-938.
22. Rish, I., 2001. *An empirical study of the naive Bayes classifier.* s.l., s.n., pp. 41-46.

23. Rubner, Y., Tomasi, C. & Guibas , L. J., 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2), pp. 99-121.
24. Sharma, G. & Jurie, F., 2011. *Learning discriminative spatial representation for image classification.* Dundee, s.n., pp. 1-11.
25. Smeulders, A. W. M. et al., 2000. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec, 22(12), pp. 1349-1380.
26. Smeulders, A. W. M. et al., 2000. Content-Based Image Retrieval at the End of the Early Years. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Dec, 22 (12), pp. 1349 - 1380.
27. Smeulders, A. W. M. et al., 2000. Content-Based Image Retrieval at the End of the Early Years. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGEN*, Dec, 22(12), pp. 1349 - 1380.
28. Smith, J. R. & Chang, S.-F., 1999. Integrated spatial and feature image query. *Multimedia Systems*, Issue 7, p. 129–140.
29. Srihari, R. K., 1995. Automatic Indexing and Content-Based Retrieval of Captioned Images. *Computer*, Sep, pp. 49-56.
30. Tuytelaars, T. & Mikolajczyk, K., 2007. Local Invariant Feature Detectors: A Survey. *Computer Graphics and Vision*, 3(3), p. 177–280.
31. Urban, J., Jose, J. M. & van Rijsbergen, C. J., 2006. An adaptive technique for content-based image retrieval. *Multimedial Tools Applied*, July, Issue 31, pp. 1-28.
32. Urban, J., Jose, J. M. & van Rijsbergen, C. J., 2006. An adaptive technique for content-based image retrieval. *Multimedial Tools Applied*, July, Issue 31, pp. 1-28.
33. Wang , Z. & Bovik, A. C., 2002. A universal image quality index. *Signal Procceing Letter*, 9(3), pp. 81-84.
34. Wang , Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P., 2004. Image Qualifty Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, April, 13(4), pp. 600–612.
35. Wang, T., Rui, Y. & Sun, J.-G., 2004. Constraint Based Region Matching for Image Retrieval. *International Journal of Computer Vision*, 56(1/2), pp. 37-45.
36. Wang, X., Liu, K. & Tang, X., 2011. *Query-Specific Visual Semantic Spaces for Web Image Re-ranking..* s.l., s.n., pp. 1-8.
37. Xiao , B., Gao, X., Tao, D. & Li, X., 2011. Recognition of Sketches in Photos. In: W. Lin, et al. eds. *Multimedia Analysis, Processing and Communications.* Berlin: Springer-Verlag, pp. 239-262.
38. Zhang, L., Wang, L. & Lin, W., 2012. Conjunctive patches subspace learning with side information for collaborative image retrieval. *IEEE Transactions on Image Processing*, 21(8), pp. 3707-3720.
39. Zhang, Y.-J., Gao, Y. & Luo, Y., 2004. Object-Based Techniques for Image Retrieval. In: S. Deb, ed. *Multimedia Systems and Content-Based Image Retrieval.* Hershey, London: IDEA Group Publishing, pp. 156-181.
40. Zhou, X. M., Ang, C. H. & Ling, T. W., 2001. Image retrieval based on object's orientation spatial relationship. *Pattern Recognition Letters*, Volume 22, pp. 469-477.