CBIR Search Engine for User Designed Query (UDQ)

Tatiana Jaworska
Sysytems Research Institute, Polish Academy of Sciences, 6 Newelska Street, 01-447, Warsaw, Poland

Keywords: CBIR, Search Engine, GUI, User Designed Query.

Abstract: At present, most Content-Based Image Retrieval (CBIR) systems use query by example (QBE), 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. For our CBIR, we prepared the dedicated GUI to construct a user designed query (UDQ). We describe 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 result with those obtained by other search engines.

1 INTRODUCTION

The users of a CBIR system have at their disposal a diversity of methods depending on their goals, in particular, *search by association*, *search for a specific image*, or *category search* (Smeulders et al., 2000). Search by association has no particular aim and implies a highly interactive iterative refinement of the search using sketches or composing images from segments offered by the system. The search for a precise copy of the image in mind, or for another image of the same object, assumes that the target can be interactively specified as similar to a group of given examples. The user requirements are reflected in the query asking methods.

The underlying assumption is that the user has an *ideal* query in mind, and the system's task is to find this ideal query (Urban et al., 2006). So far, the methods that have fulfilled these requirements can be generally divided into:

- Interactive techniques based on feedback information from the user, commonly known as relevance feedback (RF) (Azimi-Sadjadi et al., 2009);
- Automated techniques based on the global information derived from the entire collection known as browser-based;
- Automated (might also be interactive in some cases) techniques based on local information from the top retrieved results, commonly known as local feedback or collaborative image retrieval (CIR) (Zhang et al., 2012) which is used generally as a powerful tool to narrow down the semantic gap between low- and high-level concepts.

Nowadays, for image and video retrieval the scale invariant feature transformation (SIFT) and some of its variants are the most strongly recommended (Lowe, 1999), (Lowe, 2004), (Mikolajczyk and Schmid, 2004), (Tuytelaars and Mikolajczyk, 2007) as their task is: '*to retrieve all images containing a specific object in a large scale image dataset, given a query image of that object*' (Arandjelović and Zisserman, 2012).

Our approach is different, more user oriented, and that is why 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 way of the search engine works. In this paper we present, as our main contribution, a new search engine which takes into account the kind and number of objects, their features, together with different spatial location of 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. Some of such queries can be really unconventional as we can see in (Deng et al., 2012).

An additional contribution is the comparison of our empirical studies of the proposed search engine

Jaworska, T..
CBIR Search Engine for User Designed Query (UDQ).
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 372-379
ISBN: 978-989-758-158-8

with the results of other academic engines.

The system concept is universal. In the construction stage we focus on estate images but for other compound images (containing more than several objects) other sets of classes are needed.

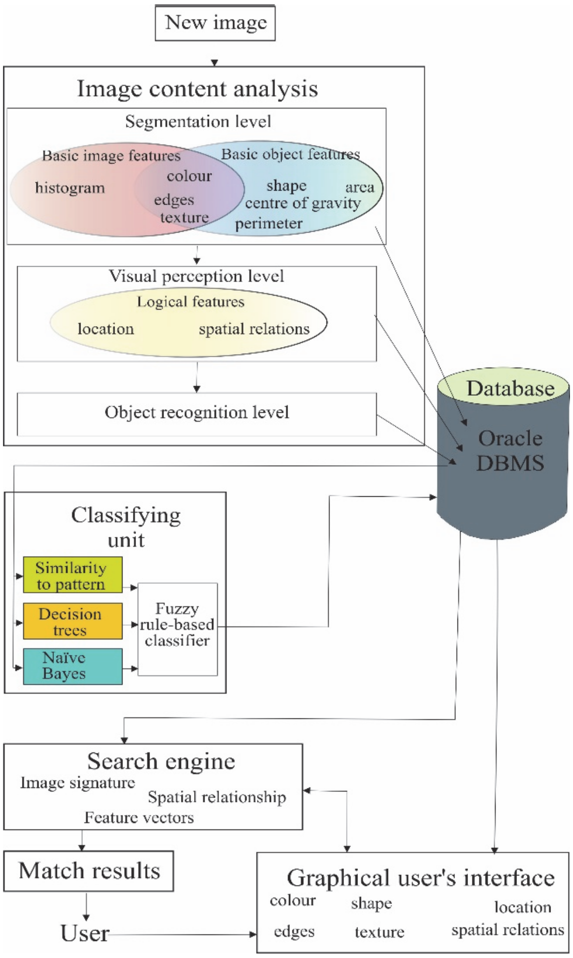


Figure 1: The block diagram of our content-based image retrieval system.

1.1 CBIR Concept Overview

In general, the presented system consists of five main blocks (see Figure 1) applied in Matlab except the database:

- the image pre-processing block, responsible for image segmentation and extraction of image object features, cf. (Jaworska, 2007);
- the classification module comprises similarity to pattern, decision tree, Naïve Bayes and fuzzy rule-based classifiers (Jaworska, 2014), used further by the search engine and the GUI. Classification helps in the transition from rough graphical objects to human semantic elements.
- the Oracle Database, storing information about whole images, their segments (here referred to as graphical objects), segment attributes, object location, pattern types and object identification, cf. (Jaworska, 2008). We decided to prepare our own DB for two reasons: (i) when the research began (in 2005) there were few DBs containing buildings which were then at the centre of our attention and (ii) some existing benchmarking databases offered separate objects (like the Corel DB) which were insufficient for our complex search engine concept. At present, our DB contains more than 10 000 classified objects of mainly architectural elements, but not only;
- the search engine (Jaworska, 2014) responsible for the searching procedure and retrieval process, based on a number of objects, the feature vectors and spatial relationship of these objects in an image. A query is prepared by the user with the GUI;
- the user-friendly semantic graphical user's interface (GUI) which allows users to compose the image they have in mind from separate graphical objects, as a query (in detail in sec. 3).

2 SEARCH ENGINE CONCEPT

2.1 Graphical Data Representation

A classical approach to CBIR consists in image feature extraction (Zhang et al., 2004). 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 (Jaworska, 2011), is described by some low-level features, such as: average colour k_{av}, texture parameters T_p, area A, convex area A_c, filled area A_f, centroid $\{x_c, y_c\}$, eccentricity e, orientation α, moments of inertia m_{11}, m_{12}, m_{21}, m_{22}, major axis length m_{long}, minor axis length m_{short}, solidity s and Euler number E and Zernike moments $Z_{00},\ldots,Z_{33}$.

All features, as well as extracted images of graphical objects, are stored in the DB. Let F_O be a set of features $F_O = \{k_{av}, T_p, A, A_c, \ldots, E\}$. We collected 45 features for each graphical object. For an object, we construct a feature vector **F** containing the above-mentioned features.

2.2 Object Classification

Thus, the feature vector **F** 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. So far, four classifiers have been implemented in this system:

- a comparison of features of the classified object with class patterns;
- decision trees (Fayyad and Irani, 1992). In order to avoid high error rates resulting from as many as 40 classes, we use the hierarchical method. A more general division is achieved by dividing the whole data set into five clusters, applying *k*-means clustering. The most numerous classes of each cluster constituting a meta-class are assigned to five decision trees, which results in 8 classes for each one.
- the Naïve Bayes classifier (Rish, 2001);
- a fuzzy rule-based classifier (FRBC) (Ishibuchi and Nojima, June 27-39, 2011), (Jaworska, 2014) is used in order to identify the most ambiguous objects. According to Ishibuchi, this classifier decides which of the three classes a new element belongs to. These three classes are taken from the three above-listed classifiers.

2.3 Spatial Object Location

Thanks to taking into account spatial object location, the gap between low-levelled and high-levelled 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 (Sharma and Jurie, 2011), spatial arrangements of regions (Smith and Chang, 1999), (Candan and Li, 2001). In some approaches image matching is proposed directly, based on spatial constraints between image regions (Wang et al., 2004).

Here, spatial object location in an image is used as the global feature (Jaworska, 2014). 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 (Chang and Wu, 1995) and later modified by Guru and Punitha (Guru and Punitha, 2004) to determine the first principal component vectors (PCVs). The idea is shown in the side boxes in Figure 2.

2.4 Search Engine Construction

Now, we will describe how the similarity between two images is determined and used to answer a query. Let the query be an image I_q, such as $I_q = \{o_{q1}, o_{q2},\ldots, o_{qn}\}$, where o_{ij} are objects. An image in the database is denoted as I_b, $I_b = \{o_{b1}, o_{b2},\ldots, o_{bm}\}$. Let us assume that there are, in total, $M = 40$ classes of the objects recognized in the database, denoted as labels L_1, L_2, …, L_M. Then, by the image signature I_i we mean the following vector:

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

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 in the following way. A query image is obtained from the GUI, where the user constructs their own image from selected DB objects. First of all, we determine a similarity measure sim_{sgn} between the signatures of query I_q and image I_b:

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

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

If the maximum component of (2) is bigger than a given threshold (a parameter of the search engine), then image I_b is rejected, i.e. not considered further in the process of answering query I_q. Otherwise, we proceed to the next step and we find the spatial similarity sim_{PCV} (3) of images I_q and I_b, based on the Euclidean, City block or Mahalanobis distance between their PCVs as:

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

If the similarity (3) is smaller than the threshold (a parameter of the query), then image I_b is rejected. The order of steps 2 and 3 can be reversed because they are the global parameters and hence can be selected by the user.

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, i.e. $L_{qi} = L_{bj}$. If there is no object $o_{bj} \in L_{qi}$, then $\mathrm{sim}_{ob}(o_{qi}, o_b) = 0$. Otherwise, similarity $\mathrm{sim}_{ob}(o_{qi}, o_b)$ between objects of the same class is computed as follows:

$$\mathrm{sim}_{ob}(o_{qi}, o_{bj}) = 1 - \sqrt{\sum_{l} (F_{qil} - F_{bjl})^2} \qquad (4)$$

where l indexes the set of features used to represent an object. Thus, we obtain the vector of similarities between query I_q and image I_b.

In order to compare images I_b with the query I_q, we compute the sum of $\mathrm{sim}_{ob}(o_{qi}, o_{bj})$ 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.

Figure 2 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 it the user is able to set threshold values for the signature, PCV and object similarity. At this stage of system verification it is useful to have these thresholds and metrics at hand. In the final Internet version these parameters will be invisible to the user, or limited to the best ranges. 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 a query miniature, a graph showing the centroids of query components and, further below, there is a graph with PCV components (cf. subsec. 2.3). 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.

The search engine has been constructed to reduce the semantic gap in comparison with CBIR systems based only on low-level features. The introduction of the image signature and object spatial relations to the search engine yields much better matching with regard to human intuition, in spite of the missing annotations.

3 USER DESIGNED QUERY CONCEPT

At present, most systems use query by example (QBE), 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

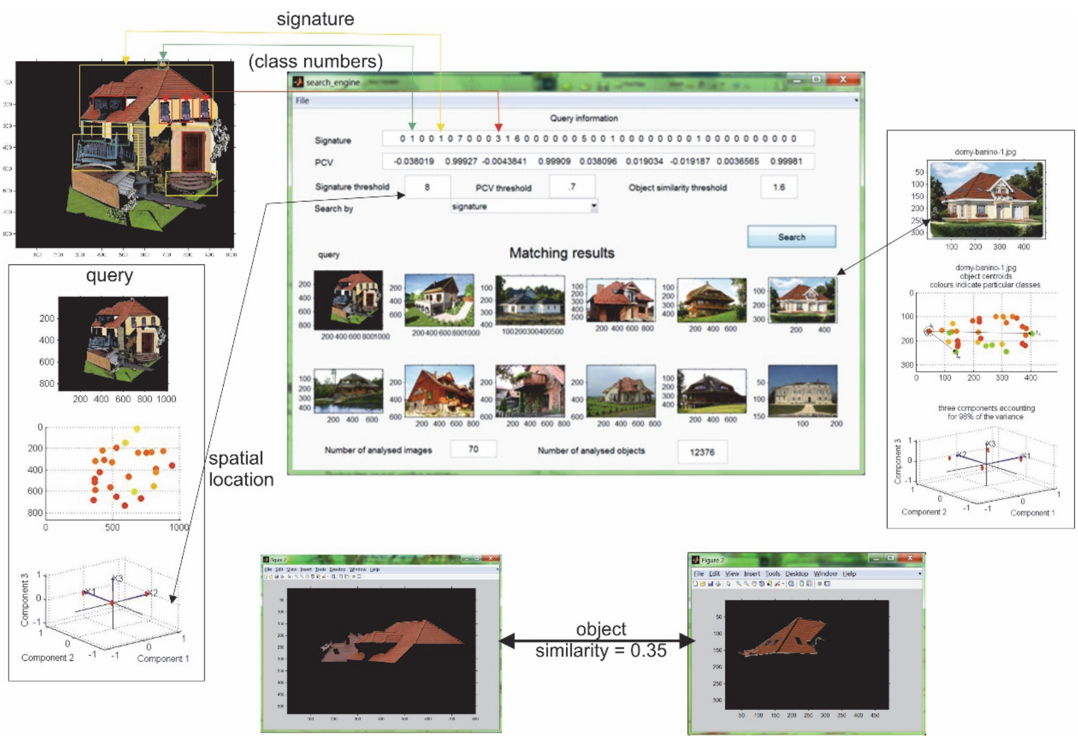


Figure 2: The main concept of the search engine.

mind to feed it to the system as a query by example. An evident example is shown in (Xiao et al., 2011) where face sketches are needed for face recognition.

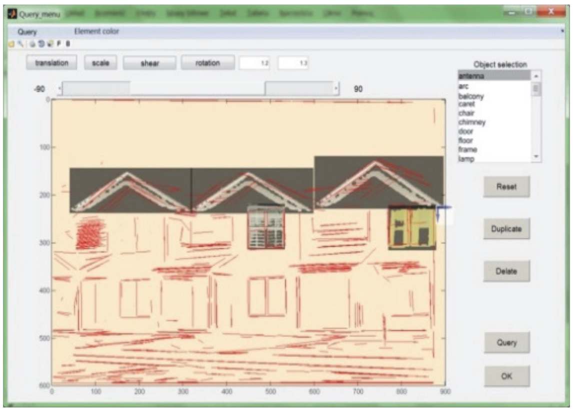


Figure 3: The main GUI window. An early stage of a terraced house query construction.

We propose a graphical editor which enables the user to compose the image he/she has in mind from the previously segmented objects (see Figure 3). It is a bitmap editor which allows for a selection of linear prompts in the form of contour sketches generated from images existing in the DB. The contours are computed as edges based on the Canny algorithm and as a vector model set to the DB during the pre-processing stage. 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 Figure 4 a));
- changing a bitmap colour (Figure 4 c) and d));
- basic geometrical transformation, such as: translation, scale, rotation and shear;
- duplication of repeating fragments;
- reordering bitmaps forwards or backwards.

This GUI is a prototype, so it is not as well-developed as commercial programs, e.g. CorelDraw, nevertheless, the user can design an image consisting of as many elements as they need. The only constraint at the moment is the number of classes introduced to the DB, which now stands at 40 but is set to increase. Once the image has been drafted, the UDQ is sent to the search engine and is matched according to the rules described in sec. 2.

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

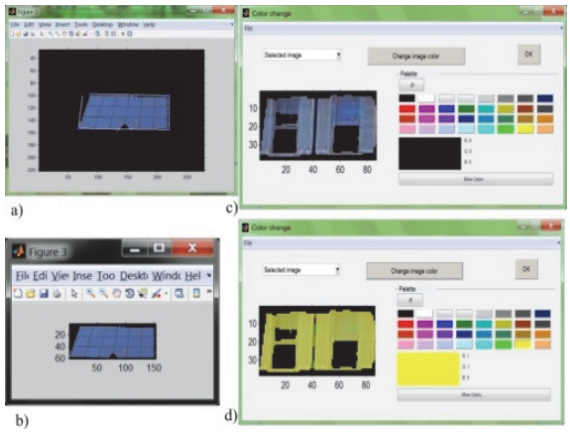


Figure 4: Main components of the GUI. We can draw a contour of the bitmap (see a) and b)) and change the colour of an element (see c) and d)).

4 RESULTS

In this section, we conduct experiments on the colour images generated with the aid of the UDQ, full images taken from our DB and we will compare our results with another academic CBIR system and the Google image search engine.

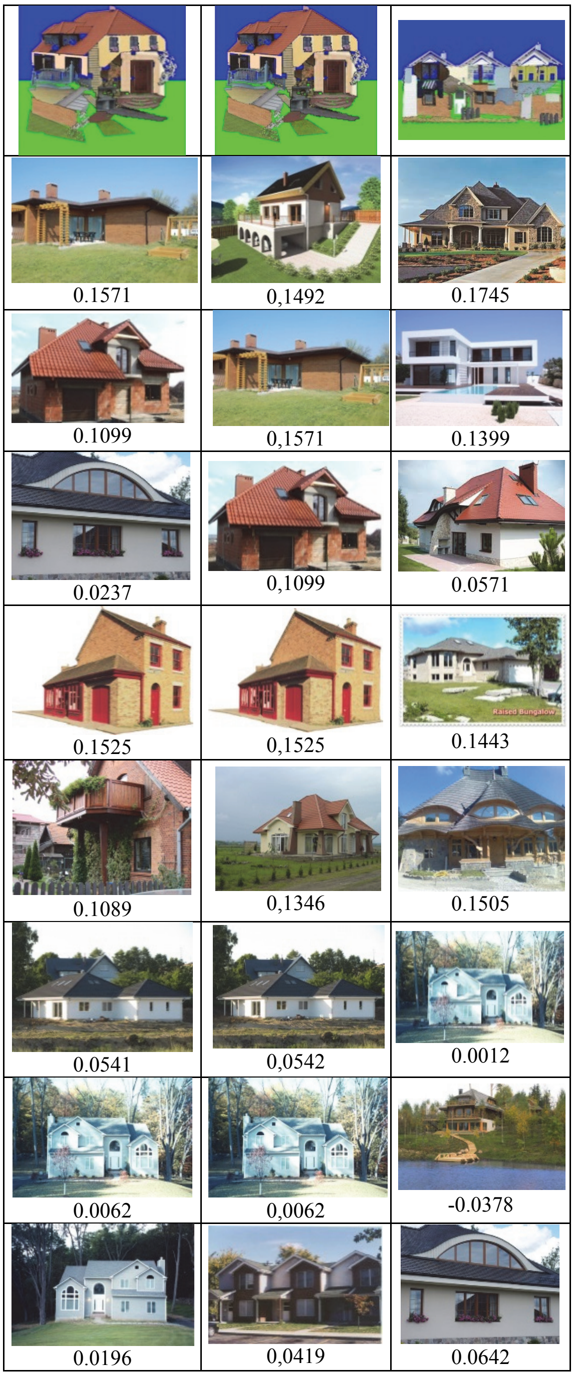
In all tables images are ranked according to decreasing similarity determined by our system. All images are in the JPG format but in different sizes. Although there are different sizes of matched images, all of them are resized to the query resolution.

Only in order to roughly compare our system's answer to the query, we used the universal image similarity index (SSIM) proposed by Wang and Bovik (Wang et al., 2004), 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 result in a negative index.

4.1 User Designed Query

A query is generated by the UDQ interface and its size depends on the user's decision, as well as the number of elements (patches). The search engine displays a maximum of 11 best matched images from our DB. Although the user designed few details, the search results are quite acceptable (see Table 1).

Table 1: The matching results for queries (in the first row) and the universal image similarity index for these matches when PCV similarity is calculated based on: (column 1) the Euclidean distance, (column 2) the City block distance (for thresholds: signature = 17, PCV = 3.5, object = 0.9), (column 3) the City block distance (for thresholds: signature = 20, PCV = 4, object = 0.9).



0.1571	0,1492	0.1745
0.1099	0,1571	0.1399
0.0237	0,1099	0.0571
0.1525	0,1525	0.1443
0.1089	0,1346	0.1505
0.0541	0,0542	0.0012
0.0062	0,0062	-0.0378
0.0196	0,0419	0.0642

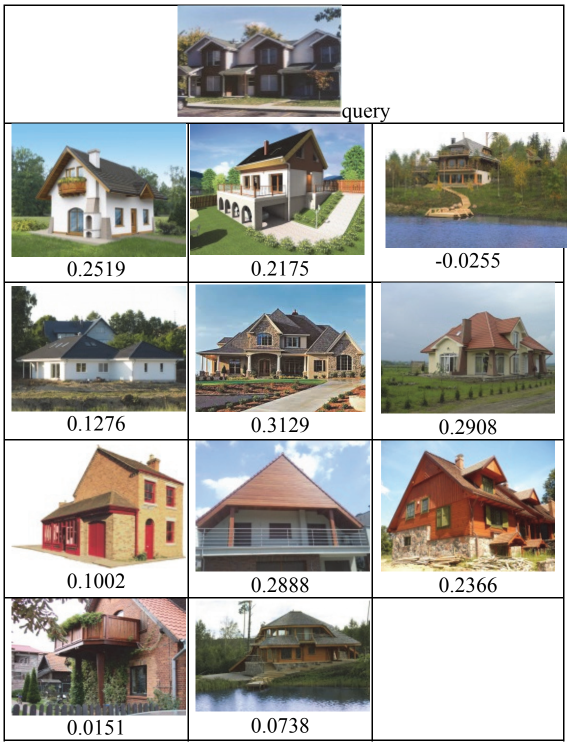
Table 1: The matching results for queries (in the first row) and the universal image similarity index for these matches when PCV similarity is calculated based on: (column 1) the Euclidean distance, (column 2) the City block distance (for thresholds: signature = 17, PCV = 3.5, object = 0.9), (column 3) the City block distance (for thresholds: signature = 20, PCV = 4, object = 0.9). (cont.)

0.1149	0,1497	0.2009
0,1496	0,1154	0.0833

4.2 Full Image

Applying the UDQ is not obligatory. The user can choose their QBE from among the images of the DB if they find an image suitable for their aim. Then the matching results are presented in Table 2.

Table 2: The matching results for QBE when PCV similarity is calculated based on the Euclidean distance. Images are ranked according to our search engine; below each there is the SSIM.



query		
0.2519	0.2175	-0.0255
0.1276	0.3129	0.2908
0.1002	0.2888	0.2366
0.0151	0.0738	

4.3 Comparison to Another Academic CBIR System

We decided to compare our results with the Curvelet Lab system which is based on the Fast Discrete Curvelet Transform (FDCT), developed at Caltech and Stanford University (Candes et al., 2006) as a specific transform based on the FFT. The FDCT is, among others, dedicated to post-processing applications, such as extracting patterns from large digital images and detecting features embedded in very noisy images.

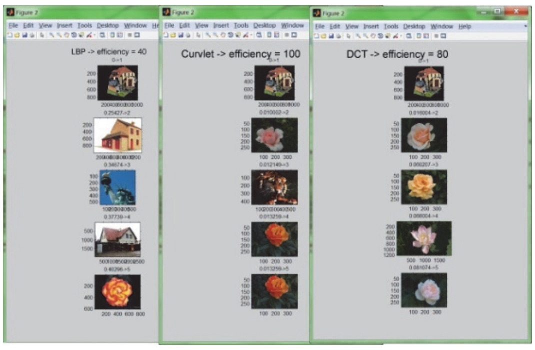


Figure 5: An example of the Curvelet Lab system retrieval for our query. (Efficiency according to Curvelet Lab system).

The Curvelet Lab system additionally offers image retrieval, based on such transforms as: DCT (Discrete Cosine Transform), LBP (Local Binary Pattern), colour and combine. FigureFigure 5 presents the results obtained for a joint set of images, namely ours and Curvelet Lab system's.

4.4 Comparison with the Google Image Search Engine

We also decided to compare our results with the Google image search engine. The results are presented in Table 3.

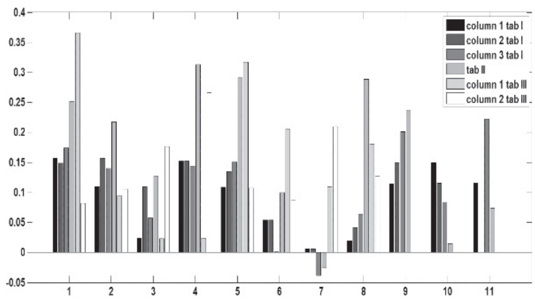


Figure 6: The comparison of SSIMs for the above-presented results from all tables.

In order to better visualise the obtained results, we compare only the SSIMs from tab. 1, 2 and tab. 3 in the form of a bar chart (see Figure 6).

Table 3: Matches for the Google image search engine and their SSIM. (Queries in the first row.)

	
 0.3658	 0.0821
 0.0939	 0.1054
 0.0232	 0.1765
 0.0240	 0.2666
 0.3174	 0.1076
 0.2056	 0.0876
 0.1095	 0.2089
 0.1807	 0.1267

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.

As for the prospects for future work, to evaluate a method more qualitative than the SSIM should be prepared. Next, 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

Arandjelović, R. & Zisserman, A., 2012. *Three things everyone should know to improve object retrieval.* Providence, RI, USA, s.n., pp. 2911-2918.

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.

Candan, S. K. & Li, W.-S., 2001. On Similarity Measures for Multimedia Database Applications. *Knowledge and Information Systems*, Volume 3, pp. 30-51.

Candes, E., Demanet, L., Donoho, D. & Ying, L., 2006. *Fast Discrete Curvelet Transforms,* pp. l-44.: raport.

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.

Deng, J., Krause, J., Berg, A. & Fei-Fei, L., 2012. *Hedging Your Bets: Optimizing Accuracy-Specificity Trade-offs in Large Scale Visual Recognition.* Providence, RI, USA, s.n., pp. 1-8.

Fayyad, U. M. & Irani, K. B., 1992. *The attribute selection problem in decision tree generation.* s.l., s.n., pp. 104-110.

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.

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.

Jaworska, T., 2007. Object extraction as a basic process for content-based image retrieval (CBIR) system. *Opto-Electronics Review*, Dec., 15(4), pp. 184-195.

Jaworska, T., 2008. *Database as a Crucial Element for CBIR Systems.* Beijing, China, World Publishing Corporation, pp. 1983-1986.

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.

Jaworska, T., 2014. *Application of Fuzzy Rule-Based Classifier to CBIR in comparison with other classifiers.* Xiamen, China, IEEE, pp. 1-6.

Jaworska, T., 2014. Spatial representation of object location for image matching in CBIR. In: A. Zgrzywa, K. Choros & A. Sieminski, eds. *New Research in Multimedia and Internet Systems.* Wroclaw: Springer, pp. 25-34.

Lowe, D. G., 1999. *Object Recognition from local scale-invariant features.* Corfu, Greece, s.n., pp. 1150-1157.

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp. 91-110.

Mikolajczyk, K. & Schmid, C., 2004. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, pp. 63-86.

Rish, I., 2001. *An empirical study of the naive Bayes classifier.* s.l., s.n., pp. 41-46.

Sharma, G. & Jurie, F., 2011. *Learning discriminative spatial representation for image classification.* Dundee, s.n., pp. 1-11.

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.

Smith, J. R. & Chang, S.-F., 1999. Integrated spatial and feature image query. *Multimedia Systems*, Issue 7, pp. 129–140.

Tuytelaars, T. & Mikolajczyk, K., 2007. Local Invariant Feature Detectors: A Survey. *Computer Graphics and Vision*, 3(3), pp. 177–280.

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.

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.

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.

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.

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.

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.