Fuzzy Oriented Graphical User Interface for Content-Based Image Retrieval System

T. Jaworska

Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland.
(e-mail: Tatiana.Jaworska@ibspan.waw.pl)

Abstract: Computing with graphical objects techniques have proved useful in the management of imperfect graphical information. This paper presents graphical user interface (GUI) constructed for Content-Based Image Retrieval System (CBIR) in order to build a query by image. In our case, this image example is prepared individually by the user from the graphical objects offered by the GUI. The images are described based on a two-staged index and compared, applying compatibility and aggregation operators. We analyze the semantic imprecision and spatial or temporal relationships of graphical objects in the compared images using the fuzzy set theory.

Keywords: CBIR, fuzzy sets, graphical index, GUI, similarity, compatibility.

1. INTRODUCTION

In recent years, the availability of image resources on the WWW has increased tremendously. This has created a demand for effective and flexible techniques for automatic image retrieval. Although attempts to perform the Content-Based Image Retrieval (CBIR) in an efficient way have been made in the past, a major problem in this area has been computer perception. In other words, there remains a considerable gap between image retrieval based on low-level features, such as shape, colour, texture and spatial relations, and image retrieval based on high-level semantic concepts, for example, houses, windows, roofs, flowers, etc. This problem becomes especially challenging when image databases are exceptionally large.

Images and graphical data are complex in terms of visual and semantic contents. Depending on the application, images are modelled and indexed using their

- visual properties (or a set of relevant visual features),
- semantic properties,
- spatial or temporal relationships of graphical objects.

Consequently, retrieval in content-based image retrieval systems is inherently fuzzy because of the imperfection in image feature definition, imperfection in query formulation methods, imperfection in the index structure, etc (Candan, 2001). Some of the main advances in this area are represented by Zadeh's fuzzy set theory and his attempt to develop a computational theory of perceptions. Many researchers have successfully used these theories to represent information with the vagueness usually involved in a natural language (Zadeh, 1999a), (Zadeh, 1999b). However, these techniques applied for computing with graphical objects and image perceptions have turned out to be a very active research area in terms of the management of imperfect graphical information.

During the last decade some concepts of extending data models in order to allow the representation of imperfect data have been used. Proposals can be found for the relational (Pons *et al.*, 2000), object-oriented (Lee, 2001), (Berzal *et al.*, 2007) and object-relational (Cubero *et al.*, 2004) database models, as well as for the CBIR (Deb, 2004), (Flickner *et al.*, 1995), (Niblack *et al.*, 1993), (Ogle, 1995). Nevertheless, programmers have limited tools when they need to develop graphical applications dealing with imperfect pictorial data. Within the scope of semantic properties, as well as graphical object properties the first successful attempt has been made by Candan and Li (Candan, 2001) who constructed the Semantic and Cognition-based Image Retrieval (SEMCOG) query processor for searching images by predicting their semantic and spatial imperfection. This new approach has been very important because earlier and even present-day queries to the database are put as query by example images.

Hence, in order to give the user the possibility of composing their own image consisting of separate graphical objects as a query, we have had to create our own system. We have dealt successfully with numerous problems involved in the CBIR system with one final issue that still requires our attention. Ultimately, we have managed to form a new paradigm in comparing images with graphical objects.

In this paper we present graphical object matching through the comparison of imprecisely described graphical objects. Fuzzily described objects may be put together to form collections (Jaworska, 2009). It is important to note that a collection can also be understood as an image. In order to improve the comparison of two images, we need to construct a feature hierarchy by linking together the global image feature (defined as spatial relationships in our approach) and local image objects, described by their vectors of fuzzy attributes. We can go even further, assuming that video images represent collections of fuzzily described images.

2. CBIR CONCEPT OVERVIEW

In general, the system consists of four main blocks:

1. the image preprocessing block (responsible for image segmentation), applied in Matlab, cf. (Jaworska, 2007);
2. 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);
3. the matching engine responsible for the two-level image indexing procedure and retrieval process based on stored data and the compatibility and aggregation operator, applied in Matlab;
4. the graphical user's interface (GUI), also applied in Matlab.

A query by image allows users to search through databases to specify the desired images. It is especially useful for databases consisting of very large numbers of images. Sketches, layouts or structural descriptions, texture, colour, sample images, and other iconic and graphical information can be applied in this search.

An example query might be: *Find all images with a pattern similar to this one,* where the user has selected a sample query image. In the QBIC system (Flickner *et al.*, 1995) the images are retrieved based on the above-mentioned attributes separately or using distance functions between features. Tools in this GUI include some basic objects such as: polygon outliner, rectangle outliner, line draw, object translation, flood fill, eraser, etc. More advanced systems enable users to choose as a query not only whole images but also some objects. The user can also draw some patterns consisting of simple shapes, colours or textures (Smith, 1999). In the SEMCOG query processor (Candan, 2001) the user could organize an image as a spatial composition of five semantic groups of objects such as: car, woman, man, house and bicycle. Additionally, the user could choose the colour, size and shape of a graphical object. In order to retrieve a matched image, the system integrated an image query statement and non-image operation statement.

2.1 Graphical Data Representation

In our system, Internet images are downloaded. Firstly, the new image is segmented creating a collection of objects. Each object, selected according to the algorithm presented in detail in Jaworska, (2007), is described by some low-level features. The features describing each object include: 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}, bounding box $\{b_1(x,y), ..., b_s(x,y)\}$ (s – number of vertices), major axis length m_{long}, minor axis length m_{short}, solidity s and Euler number E. All features, as well as extracted images of graphical objects, are stored in the DB.

Let F be a set of features where: $F = \{k_{av}, T_p, A, A_c,\ldots, E\}$. For ease of notation we will use $F = \{f_1, f_2, \ldots, f_r\}$, where r – number of attributes. For an object, we construct a feature vector O containing the above-mentioned features:

$$O = \begin{bmatrix} O(k_{av}) \\ O(T_p) \\ O(A) \\ \vdots \\ O(E) \end{bmatrix} = \begin{bmatrix} O(f_1) \\ O(f_2) \\ O(f_3) \\ \vdots \\ O(f_r) \end{bmatrix} \tag{1}$$

This feature vector is further used for object classification. Therefore, we have to classify objects first in order to assign them to a particular class and second in order to compare objects coming from the same class.

The pattern library (one of tables from the DB) contains information about pattern types, shape descriptors, object location and allowable parameter values for an object. We define a model feature vector P_k for each graphical element. We assume weights μ_P characteristic of a particular type of element which satisfy: $\mu_{P_k}(f_i) \in [0,1]$ where: $1 \le i \le r$, k -- number of patterns. These weights for each pattern component should be assigned in terms of the best distinguishability of patterns.

First, each graphical extracted object is classified into a particular category from the pattern library. For this purpose, in the simplest case, we use an L_m metric, where the distance between vectors O and P_k in an r - dimensional feature space is defined as follows:

$$d(O, P_k) = \left[\sum_{i=1}^{r} \mu_{P_k}(f_i) \left| O(f_i) - P_k(f_i) \right|^m \right]^{1/m} \tag{2}$$

where: k – pattern number, $1 \le i \le r$, m is the order of the metric. For $m = 1$ and for $m = 2$, it becomes the Manhattan and the Euclidean distance, respectively.

3. THE GLOBAL FEATURE AS A SPATIAL OBJECT LOCATION

High resemblance values are used for graphical object classification into a particular category. Object features can be described by applying fuzzy sets (Berzal *et al.*, 2007) in order to improve the retrieval efficiency. Images are spatial collections, and that is why they can be analysed as a fuzzy collection of fuzzily described graphical objects. Nevertheless, it is not sufficient for full image identification. There is also a need for assigning a global feature to confirm image similarities.

Chow, Rahman and Wu (Chow *et al.*, 2006) proposed a tree-structured image representation, where the root node contains the global features, while child nodes contain the local region-based ones. This approach hierarchically integrates more information on image contents to achieve better retrieval accuracy than global and regional attributes, compared individually. The next step is an examination of mutual relationships of objects and object position in the whole image. Candan and Li (Candan, 2001) analysed the description of objects' mutual relationships based on different fuzzy operators.

In our system, spatial object location in an image is also used as a global feature. Firstly, it is easy to recognize this spatial location visually by the user. Secondly, it supports full identification based on rules for location of graphical elements (Jaworska, 2009). Let us assume that we analyse a house image. Then, for instance, an object which is categorized as a window cannot be located over an object which is categorized as a chimney. For this example, rules of location mean that all architectural objects must be inside the bounding box of a house. For an image of a Caribbean beach, an object which is categorized as a palm cannot grow from the middle of the sea, and so on. For this purpose, the mutual position of all objects is checked. The location rules are also stored in the pattern library (Jaworska, 2008). Note that object classification reduces the differences between high-level semantic concepts perceived by people and precise value features interpreted by computers.

Thus, we are going further because we are taking into account imprecisions in objects' mutual relationships such as:

- the imprecision of object size scalability which is offered to the user by our GUI;
- the imprecision of object spatial location that the user cannot avoid preparing their query image with the computer mouse.

These imprecisions are described with the use of the fuzzy set theory. Namely, based on the similarity and compatibility in the fuzzy set theory (Cross and Sudkamp, 2002) we can use some compatibility and aggregation operators.

Two of the most popular scoring functions are the *min* and *products* semantics of fuzzy logical operators (∧, ∨, ¬). We can present these two semantics (in the form of a table), as follows: let $P = \{P_1, \ldots, P_m\}$ denote fuzzy sets and $M = \{\mu_1(x), \ldots, \mu_m(x)\}$ corresponding membership functions. Table 1 presents the *min* and *products* semantics.

Table 1. *Min* and *products* semantics for fuzzy logical operators.

Min semantics
$\mu_{P_i \wedge P_j}(x) = \min\{\mu_{P_i}(x), \mu_{P_j}(x)\}$
$\mu_{P_i \vee P_j}(x) = \max\{\mu_{P_i}(x), \mu_{P_j}(x)\}$
$\mu_{\neg P_i}(x) = 1 - \mu_{P_i}(x)$

In our case, spatial information, specifically, the objects' mutual relationships, is presented as vector F_g of the global feature:

$$F_g = \{\min(\mu(x_{c_1}), \mu(y_{c_1})), o_1, \min(\mu(x_{c_2}), \mu(y_{c_2})), o_2, \\ \ldots, \min(\mu(x_{c_N}), \mu(y_{c_N})), o_N\} \tag{3}$$

where: N – the number of all objects in an image, $\{x_{c_i}, y_{c_i}\}$ is an object centroid, $\mu(x_{c_i}), \mu(y_{c_i})$ - trapezoidal membership functions (TMF) describing user's imprecision of object spatial location in the x and y directions, respectively, $\min(\mu(x_{c_i}), \mu(y_{c_i}))$ - is a fuzzy logical operator, o_k – an object label, assigned in the process of identification. By definition, a trapezoidal MF is specified by four parameters {a,b,c,d} as follows:

$$\text{trapezoid}\,(a,b,c,d) = \begin{cases} 0, & x \le a. \\ (x-a)/(b-a), & a \le x \le b. \\ 1, & b \le x \le c. \\ (d-x)/(d-c), & c \le x \le d. \\ 0, & d \le x. \end{cases} \tag{4}$$

The middle of the range [b,c] is located at the point $\{x_{C_i}\}$ for x - axis and $\{y_{C_i}\}$ for y - axis, respectively for particular object. As you can see in Fig. 1, we analyse the mutual spatial location for classified types of objects.

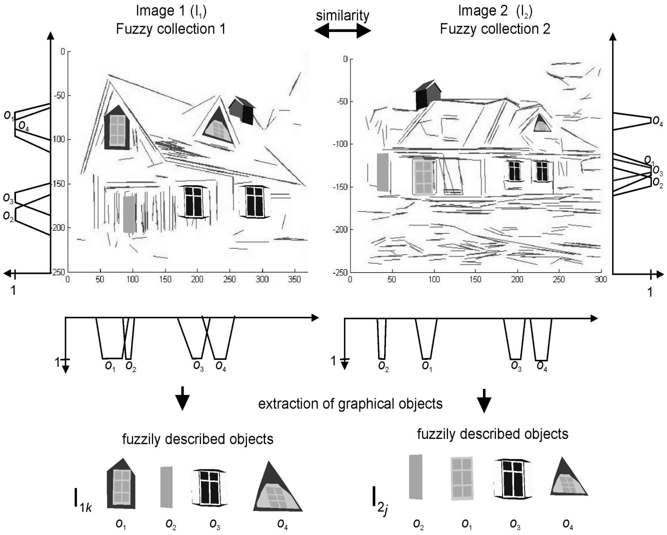

Fig. 1. General scheme of similarity between two images compared as two collections. The model of the spatial object location is described as a global vector F_g. For each object o_i we know its feature vector O. Image I_1 is treated as a query image Q. Supports for fuzzy sets correspond to the uncertainty of the x and y size of objects $o_{i.}$

4. QUERY BY IMAGE AS A CRUCIAL ELEMENT OF CBIR

Graphical User Interface (GUI) is an immanent element of our system as the area of the human-computer interaction (Newman *et al.*, 1996). Hence, we have made an effort to create a useful tool for the user who is interested in designing their own image. This design is treated as a query by image. Fig. 2 presents the main GUI window entitled “Query_ menu”. In the left window the user can choose the image outlines which become visible in an enlarged form in the main window.

Next, the user chooses particular graphical elements from subsequent menus and situates them on the appropriate location in the chosen outline. These elements can be scaled in a limited range. For each element the user can change its colour (see Fig. 3). Moreover, there is a window for changing the texture of an element, if it has one, or adding a texture for non-textured components. For a texture the user can also choose its colour.

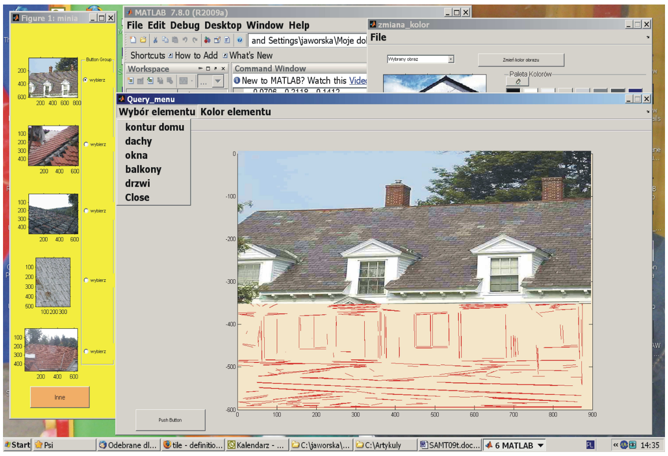

Fig. 2. The user menu applied by the system to design a query by image. The left window is used to present graphical elements, for example house roofs. It is easy to notice that the first roof at the top of the list of miniatures on the left is chosen and located in the house outline.

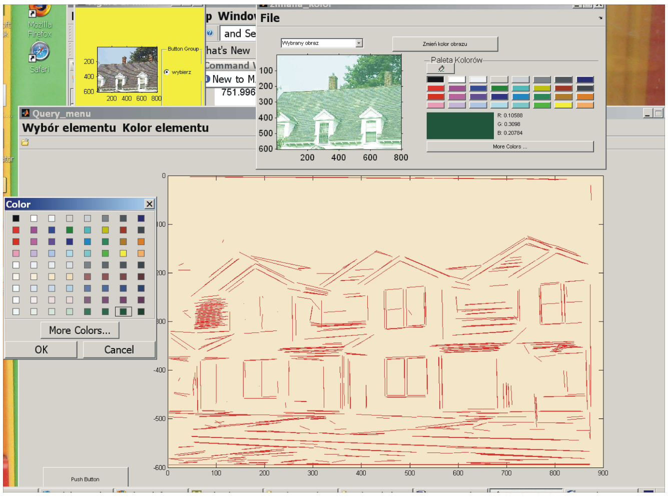

Fig. 3. Menu tools dedicated to changing the element colour. When the user selects a graphical element from the window containing miniatures, they can open the “zmiana_koloru” window in order to change the colour of this element. If the basic colour pallet appears too limited, the user can open the “more colours” window. Having determined the element colour, the user locates the element in the appropriate position in the image outline.

In most query by example systems, the features for retrieval and their importance are estimated by the system. Even in systems where such information can be provided by the user, users cannot always communicate unambiguously what they are looking for. In our system, these constraints are overcome by the user’s selection of specific features from numerous menus. After the designing process, the image is sent as a query to the DB.

For more advanced users, there are additional options in the query interface which enable them to select the most interesting feature. These preferences are implemented in the system as weights μ_{qo} which are taken into account during the final matching. Then we compare a query object with a feature vector $O_q = \mu_{q_{Oi}}(f_i)$ to objects stored in the DB. This fact is especially important when we use fuzzily described object features.

The GUI is strictly dedicated to the CBIR system and consists of the most important components only. In further work some additional menus will be added if a need to improve the retrieval process arises.

5. IMAGE MATCHING STRATEGY

Image matching is conducted with the aid of object recognition and spatial relationships. Query image $Q = \{F_{gq}, O_{q_1}, ..., O_{q_N}\}$ consists of a global feature vector F_{gq} and a set of identified objects O_{q_k}, for which feature vectors are known and where $1 \le k \le N$. First, the relevant images $R = \{F_{gR}, O_{R_1}, ..., O_{R_N}\}$ with N objects are searched for in the database. Next, we check if objects have the same pattern o_k. If the answer is positive, then the global feature vectors F_{gq} and F_{gR} are compared. Their similarities are searched for based on mutual object locations in the images.

This means that objects are not matched based upon fixed positions in the image. For example, in the query which you can see in Fig. 4, object *c* (chimney) is to the right of object *r* (root) and object w_r (root window) is above object *d* (door) or *w* (window). This information is collected and stored in tables as a global feature. For matching images *Q* and *R*, whose spatial information is illustrated in tables 2, 3 and 4, we compare each table cell. The notation used in the tables is as follows: E - object O_1 is to the east of object O_2, W - object O_1 is to the west of object O_2, S - object O_1 is to the south of object O_2, N - object O_1 is to the north of object O_2. Numbers before letters refer to the distances between object centroids, vertically and horizontally, respectively.

Table 2. Spatial information for *query* image from Fig. 4.

	c	***r***	w_r	***d***	***w***
c	0	SE	SE	SE	SE
r	NW	0	0	SE	SW
w_r	NW	0	0	SE	SW
d	NW	NW	NW	0	NW
w	NW	NE	NE	SE	0

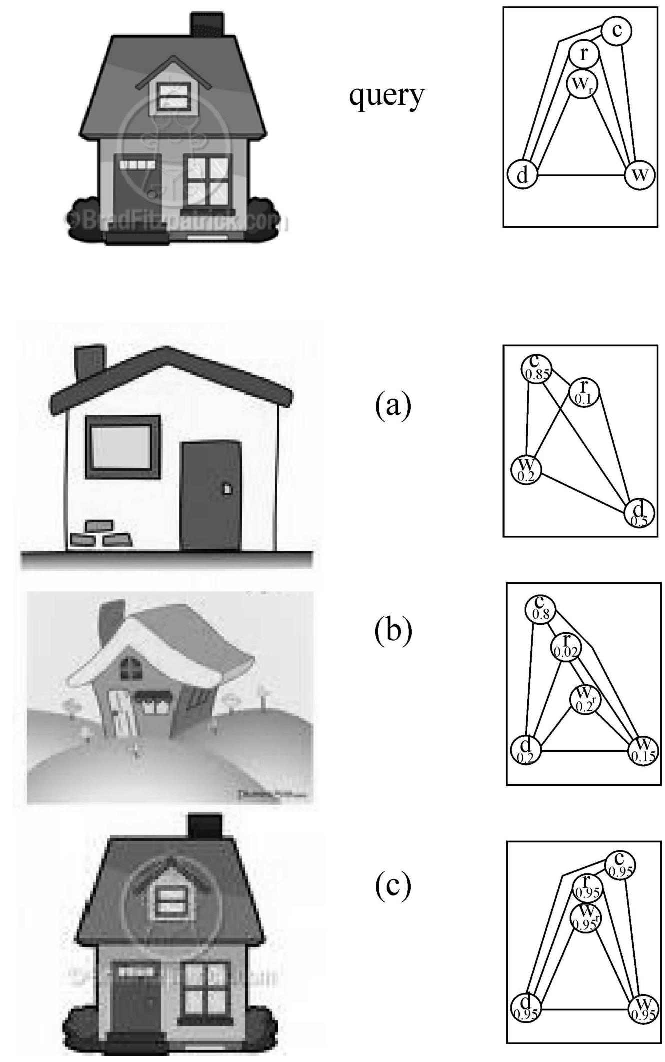

Fig. 4. Query image and candidates for retrieval images (a), (b), (c). Numbers in the graph nodes describe the resemblance degree between query objects and the candidate ones.

Table 3. Spatial information for relevant image (a) from Fig. 4.

	c	***r***	***w***	***d***
0.85 ***c***	0	SW	SW	SW
0.1 ***r***	NE	0	SW	SW
0.2 ***w***	NE	NE	0	SW
0.5 ***d***	NE	NE	NE	0

Table 4. Spatial information for relevant image (b) from Fig. 4.

	c	***r***	w_r	***d***	***w***
0.8 ***c***	0	SW	SW	S	SW
0.02 ***r***	NW	0	S	SE	SW
0.2 w_r	NW	N	0	SE	SW
0.2 ***d***	N	NW	NW	0	NW
0.15 ***w***	NE	NE	NE	SW	0

Numbers in the first column denote the degree of similarity between objects in the query image and relevant image.

5.1 Discussion

As we can see in Fig. 4, the most similar to the query is the house in Fig 4 (c) because only its colour is different. That is why the $F_{gq} = F_{gR}$ and only objects differ slightly.

In case of a lack of relevant images the user can decide if spatial information is the most important for them. If the objects are more important, we can limit the matching procedure only to $Q = \{O_{q_1},...,O_{q_N}\}$. In our example, Fig. 4 b) has the same kind and number of objects as a query. We can also imagine a situation in which the user's preferences enable us to assess weaker constraints for the object matching. In this case we can manipulate values μ_W of membership functions in the object feature vector $O_k(f_i)$.

A tricky situation is when we are trying to find an image which is, for instance, a half of another picture. We could, then, assume that all images for which a half of our query table matched are relevant because we do not know the requirements for the second half. Another situation with similar symptoms occurs when we analyze images of scenes taken from different angles in which objects are nearly the same, but the spatial information does not match. What takes place then is limited to full object identification.

6. CONCLUSIONS

The construction of a CBIR system requires combining some different functional systems linked together and cooperating with each other. Having built the image processing module for automatic segmentation and the database to store the generated information about images and their segments, we face the problem of image retrieval. For this purpose the object classification and identification procedures have been established and the GUI prototype has been constructed. Unfortunately, the whole system has not been tested yet by sufficient numbers of users, so we have not been able to estimate the best criteria and solutions on terms of efficiency of image matching. It is task for the nearest future.

For more flexibility in the whole image matching we have had to take into account some imprecisions that appear with the construction of a query image by the user and other imprecisions that have emerged in the matching process as a result of a lack of our (or the system's) knowledge. For these reasons we have used fuzzy sets and fuzzy collections (Jaworska, 2009) Thanks to this, we can modify some feature vectors according to the user's preferences. We have also achieved some flexibility in spatial image matching by taking into account fuzzy object location in the image.

In the future we hope, among others, to incorporate some real-world knowledge into our implementation which should help the user to avoid mistakes such as the above-mentioned location of a palm in the middle of the sea. Also recognition of upside-down images will be possible. Moreover, we will increase numbers of object classes that are identified by the system. Additionally, we will test different kinds of aggregation operators in order to find the most optimal matching mechanisms.

To sum up, even though we have experienced a few snags, we have succeeded in constructing software that will eventually provide users with an advanced, user-friendly CBIR system

REFERENCES

Berzal, F., Cubero, J. C., Kacprzyk, J., Marin, N., Vila M. A. and Zadrożny, S. (2007). A General Framework for Computing with Words in Object-Oriented Programming. In Bouchon-Meunier B. (ed.), *International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems*, Vol. 15, (Supplement), Feb., pp. 111-131. World Scientific Publishing Company, Singapore.

Candan, K. Selçuk, and Li, Win-Syan, (2001). On Similarity Measures for Multimedia Database Applications. Knowledge and Information Systems, (3), pp. 30 -51.

Chow, T. W., Rahman, M. K. and Wu, S. (2006). Content-based image retrieval by using tree-structured features and multi-layer self-organized map, *Pattern Analysis and Applications,* Vol. 9, pp. 1—20.

Cross, V. V. and Sudkamp, T. A., (2002). *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications.* Studies in Fuzziness and Soft Computing, Vol. 93, Physica –Verlag, Heidelberg and New York.

Cubero, J. C., Marin, N., Medina, J. M., Pons, O. and Vila, M.A. (2004). Fuzzy Object Management in an Object-Relational Framework, In *Proceedings of the 10th International Conference IPMU*, Perugia, Italy, pp. 1775-1782.

Deb, S. (ed.), (2004). *Multimedia Systems and Moment-Based Image Retrieval*, Chap. VII and XI. IDEA Group Publishing, Melbourne.

Flickner, M. and Sawhney, H., et al. (1995). Query by Image and Video Content: The QBIC System. *IEEE Computer* **28** (9), pp. 23 -32.

Jaworska, T., (2007). Object extraction as a basic process for content-based image retrieval (CBIR) system. *Opto-Electronics Review*, Association of Polish Electrical Engineers (SEP), Vol.15 (4), Warsaw, pp. 184-195.

Jaworska, T., (2008). Database as a Crucial Element for CBIR Systems. In *Proceedings of the 2nd International Symposium on Test Automation and Instrumentation*, Vol. 4, World Publishing Corporation, Beijing, China, pp. 1983 -1986.

Jaworska, T., (2009). Graphical object indexing and query by image as an aspect of Content-Based Image Retrieval System, Research Report: RB/41/2009, Systems Research Institute, Polish Academy of Sciences, Warsaw.

Lee, J., Kuo, J-Y. and Xue, N-L. (2001). A note on current approaches to extent fuzzy logic to object oriented modeling. *International Journal of Intelligent Systems*, Vol. 16, pp. 807 –820.

Newman W. M., Lamming M. G. (1996). *Interactive System Design*. Addison-Wesley, Harlow

Niblack W., Flickner M. et al. (1993) The QBIC Project: Querying Images by Content Using Colour, Texture and Shape, *SPIE* **1908**, pp. 173 - 187.

Ogle, V. and Stonebraker, M. (1995). CHABOT: Retrieval from a Relational Database of Images. *IEEE Computer* **28**, (9), pp. 40 -48.

Pons, O., Vila, M. A., and Kacprzyk, J. (2000). *Knowledge management in fuzzy databases,* Studies in Fuzziness and Soft Computing, Vol. 39, Physica –Verlag, Heidelberg and New York.

Smith, J. R. and Chang, Sh-F. (1999). Integrated spatial and feature image query, *Multimedia Systems*, Vol. 7, Springer-Verlag, pp. 129-140.

Zadeh, L. A. and Kacprzyk, J., (ed.), (1999a). *Computing with Words in Information/Intelligent Systems. 1 Foundations*. Studies in Fuzziness and Soft Computing, Vol. 33, Physica –Verlag, Heidelberg and New York.

Zadeh, L. A. and Kacprzyk, J., (ed.), (1999b). *Computing with Words in Information/Intelligent Systems. 2 Applications.* Studies in Fuzziness and Soft Computing, Vol. 33, Physica –Verlag, Heidelberg and New York.