Application of Fuzzy Rule-Based Classifier to CBIR in comparison with other classifiers

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

Abstract—**At present a great deal of research is being done in different aspects of Content-Based Image Retrieval (CBIR). Image classification is one of the most important tasks that must be dealt with in image DB as an intermediate stage prior to further image retrieval. The issue we address is an evolution from the simplest to more complicated classifiers. Firstly, there is the most intuitive one based on a comparison of the features of a classified object with a class pattern. Next, the paper presents decision trees and Naïve Bayes as another option in a great number of classifying methods. Lastly, to assign the most ambiguous objects we have built fuzzy rule-based classifiers. We propose how to find the ranges of membership functions for linguistic values for fuzzy rule-based classifiers according to crisp attributes. Experiments demonstrate the precision of each classifier for the crisp image data in our CBIR. Furthermore, these results are used to describe a spatial object location in the image and to construct a search engine taking into account data mining.**

Keywords-CBIR, classification, decision trees, fuzzy rule-based classifiers

I. INTRODUCTION

In recent years, the availability of image resources and large image datasets has increased dramatically. This has created a demand for effective and flexible techniques for automatic image classification and retrieval. Although attempts to construct the Content-Based Image Retrieval (CBIR) in an efficient way have been made before [3, 9, 11], a major problem in this area, i.e. the extraction of semantically rich metadata from computationally accessible low-level features, still poses a tremendous scientific challenge and constitutes a topic open to research. A prominent trend in scientific interests focuses on matching the most similar set of images to the query image. In this field, the latest achievements using relevance feedback and collaborative image retrieval seem to be very promising [21, 29, 30, 31,32]. However, our motivation was to give the user a more powerful tool, namely, to give him/her the opportunity to construct their own images based on different image segments. The image constructed in that way which, in fact, consists of elements similar to different images, is our query.

Seemingly, the SIFT method introduced by Lowe [22] and developed by Cho et al. [8] is the most powerful one because it matches separate objects from one image to another. This approach requires immense calculating capacity. But our aim is to give the user classified images (segments of images) for the GUI for their own image construction, thus we cannot avoid image classification.

There are a number of standard classification methods in use, such as: *k*-NN [10], SVM [13], Naïve Bayes classifier [25], neural network [28], and others [2]. Having surveyed these algorithms, we started our classification from the simplest one, namely, the similarity to the pattern which compares the features of a classified object with the set of pattern features which define classes.

Object classification is so important in the context of CBIR because it is used for several purposes, for example [20]:

- to compare whole images. Specifically, an algorithm which describes a spatial object location needs classified objects (see sec. VI);
- to help the user form a query in the GUI. The user forms a query choosing graphical objects semantically collected in groups;
- to compare image objects coming from the same class as a stage in the image retrieval process.

A. CBIR concept overview

In general, our system consists of five main blocks (Fig. 1), implemented in Matlab ver. R2013b:

- the image preprocessing block, responsible for image segmentation and extraction of image object features, (cf. [18]);
- the database, which is implemented in the Oracle Database (DB), stores information about whole images, their segments (here referred to as graphical objects), segment attributes, object location, pattern types and object identification, (cf. [19]);
- the classification module, which is used by the search engine and the GUI. The algorithms applied in this module will be described in the following sections.
- the search engine responsible for the searching procedure and retrieval process based on feature vectors of objects and spatial relationship of these objects in an image;

978-1-4799-5148-2/14/$31.00 ©2014 IEEE

- the graphical user's interface (GUI), which allows users to compose their own images, consisting of separate graphical objects as a query. Classification helps in the transition from rough graphical objects to human semantic elements which is expedient in creating a user-friendly semantic system.

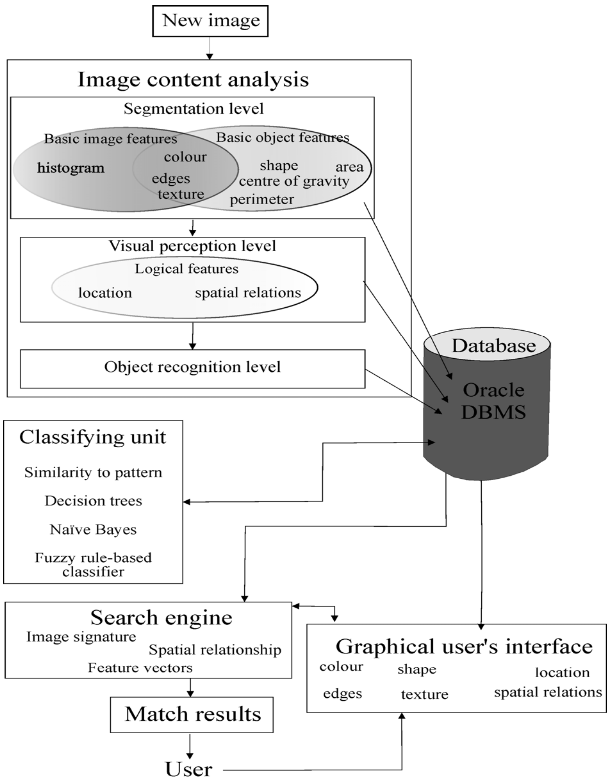

Figure 1. Block diagram of our content-based image retrieval system.

B. Representation of graphical data

As we have mentioned above, we are interested in image segmentation as the first stage of preprocessing. At present, there are tendencies to use more natural thus simpler methods to separate foreground objects from the monolithic background using wavelets [24] or morphological operations [7], whereas in our system the foreground objects, mainly houses, sometimes overlapped by trees, are found against complicated, multi-objected backgrounds. Surely, such images are more challenging and the segmentation process forced us to use different methods to obtain the desired features.

In a nutshell, a new image is segmented, yielding as a result a collection of objects from several up to more than 100. Both the image and the extracted objects are stored in the database. Each object, selected according to the algorithm presented in detail in [18], is described by features, such as: average colour k_{av}, texture parameters T_p (based on wavelet algorithm), area A, convex area A_c, filled area A_f, centroid $C=\{x_c, y_c\}$, eccentricity e, orientation α, moments of inertia $m_{11},\ldots, m_{22}$, bounding box $\{bb_1(x,y), ..., bb_4(x,y)\}$, major axis length m_{long}, minor axis length m_{short}, solidity s and Euler number E and complex Zernike moments $Z_{00},\ldots,Z_{33}$ [26].

Let Fo be a set of features where $F_O = \{k_{av}, T_p, A, A_c, ..., E\}$. Hence, for an object, we construct a feature vector: $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, where $n = 45$ is the number of the above-mentioned features.

II. Standard Classification Methods

A. Similarity to Pattern

The simplest approach to classification is the comparison of an object feature vector $\mathbf{x}$ to the previously prepared patterns P_k for each class used, for instance, Euclidean or Minkowski's metrics. We tried to design classes which attribute (the) objects in accordance with human perception. In our system the main set contains house images, which prompts division of objects into $M = 32$ architectural classes (technical details in App. A).

B. Decision Trees

In the construction of decision trees [12], a measure of discrimination is used in order to rank attributes and select the best one. Each vertex of a binary tree is associated with an attribute [5]. In order to avoid high error rates resulting from as many as 32 classes we use the hierarchical method. A more general division is created by dividing the whole data set into four clusters, applying k-means clustering. The most numerous classes of each cluster, constituting a meta-class are assigned to four decision trees, which results in 8 classes for each one. The process of tree construction is very prone to the unequal numbers of elements in learning subsets for each class.

Additionally, the Naïve Bayes classifier [25] has been implemented and now it seems to be as good as decision trees.

III. Fuzzy Classification

The results presented in sec. V indicate that there are objects difficult to classify. Some difficulties arise from mistakes in object segmentation. Some others come from the fact that there are imbalanced classes because some architectural elements are very numerous or they are always found in images (for instance, windows) but others are very rare (for example, some kinds of lamps, pillars or satellite dishes). All this motivated us to use fuzzy sets [27], specifically, the fuzzy rule-based classifiers.

A. Fuzzy rule-based classifiers

Let us consider an M-class classification problem in an n-dimensional normalized hyper-cube $[0,1]^n$. For this problem, we use fuzzy rules of the following type [16]:

$$\text{Rule } R_q: \quad \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then Class } L_q \text{ with } CF_q, \quad (1)$$

where R_q is the label of the q^{th} fuzzy rule, $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional feature vector, A_{qi} is an antecedent fuzzy set ($i = 1,...,n$), L_q is a class label, CF_q is a real number in the unit interval [0,1] which represents a rule weight. The rule weight can be specified in a heuristic manner or it can be adjusted, e.g. by a learning algorithm introduced by Ishibuchi et al. [23, 17].

We use the n-dimensional vector $\boldsymbol{A}_q = (A_{q1}, \ldots, A_{qn})$ to represent the antecedent part of the fuzzy rule R_q in (1) in a concise manner.

A set of fuzzy rules S of the type shown in (1) forms a fuzzy rule-based classifier. When an n-dimensional vector $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$ is presented to S, first the *compatibility grade* of $\mathbf{x}_p$ with the antecedent part $\boldsymbol{A}_q$ of each fuzzy rule R_q in S is calculated as the product operator

$$\mu_{A_q}(\mathbf{x}_p) = \mu_{A_{q1}}(\mathbf{x}_{p1}) \times \ldots \times \mu_{A_{qn}}(\mathbf{x}_{pn}) \quad \text{for } R_q \in S \qquad (2)$$

where $\mu_{A_{qi}}(.)$ is the membership function of A_{qi}. Then a single winner rule $R_{w(\mathbf{x}_p)}$ is identified for $\mathbf{x}_p$ as follows:

$$w(\mathbf{x}_p) = \arg\max_q \ \{CF_q \times \mu_{A_q}(\mathbf{x}_p) \mid R_q \in S\}, \qquad (3)$$

where $w(\mathbf{x}_p)$ denotes the rule index of the winner rule for $\mathbf{x}_p$.

The vector $\mathbf{x}_p$ is classified by the single winner rule $R_{w(\mathbf{x}_p)}$ belonging to the respective class. If there is no fuzzy rule with a positive *compatibility grade* of $\mathbf{x}_p$ (i.e., if $\mathbf{x}_p$ is not covered by any fuzzy rules in S), the classification of $\mathbf{x}_p$ is rejected. The classification of $\mathbf{x}_p$ is also rejected if multiple fuzzy rules with different consequent classes have the same maximum value on the right-hand side of (3). In this case, $\mathbf{x}_p$ is on the classification boundary between different classes. We use the single winner-based fuzzy reasoning method in (3) for pattern classification.

An ideal theoretical example of a simple three-class, two-dimensional pattern classification problem with 20 patterns from each class is considered by Ishibuchi and Nojima [16]. There three linguistic values (*small*, *medium* and *large*) are used as antecedent fuzzy sets for each of the two attributes, and 3×3 fuzzy rules are generated. S_1 is the fuzzy rule-based classifier with nine fuzzy rules shown below (Fig. 2):

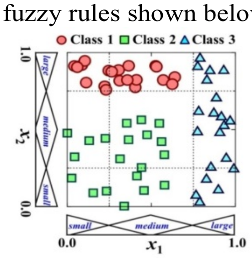

Figure 2. An ideal example of a fuzzy rule-based classifier S_1 developed by Ishibuchi and Nojima [16].

S1: fuzzy rule-based classifier with nine fuzzy rules

R_1: If x_1 is *small* and x_2 is *small* then Class2 with 1.0,
R_2: If x_1 is *small* and x_2 is *medium* then Class2 with 1.0,
R_3: If x_1 is *small* and x_2 is *large* then Class1 with 1.0,
R_4: If x_1 is *medium* and x_2 is *small* then Class2 with 1.0,
R_5: If x_1 is *medium* and x_2 is *medium* then Class2 with 1.0,
R_6: If x_1 is *medium* and x_2 is *large* then Class1 with 1.0,
R_7: If x_1 is *large* and x_2 is *small* then Class3 with 1.0,
R_8: If x_1 is *large* and x_2 is *medium* then Class3 with 1.0,
R_9: If x_1 is *large* and x_2 is *large* then Class3 with 1.0.

B. Construction of membership functions

The theoretical method presented by Ishibuchi does not answer the question how to construct membership functions, especially those corresponding to linguistic values. Hamilton and Stashuk [15] gave a suggestion for the construction of membership functions based on the standardized residual analysis but they applied it to continuous data.

For our discrete data, we solved this problem calculating the mean value $\bar{x}$ and standard deviation σ for the elements of each of the three classes. The membership function of each class is constructed as a symmetrical trapezoidal function in respect to the mean value $\bar{x}$ where the smaller basis has the σ length and the longer one - 2σ. Then, we divide the ranges of features x_1 and x_2 into three equal intervals. Next, we assign the mean value of a particular class to correspondent intervals which represent the proper linguistic values. The effect is visible in Fig. 3 for the horizontal and vertical axes in the side subplots, respectively.

In each case, the fuzzy rule-based classifier is constructed automatically by matching the membership function related to the proper linguistic value, resulting in the right class for each rule. The classifier S_2 corresponds to the example seen in Fig. 3. In this case, x_1 is orientation and x_2 the real part of Zernike's moment.

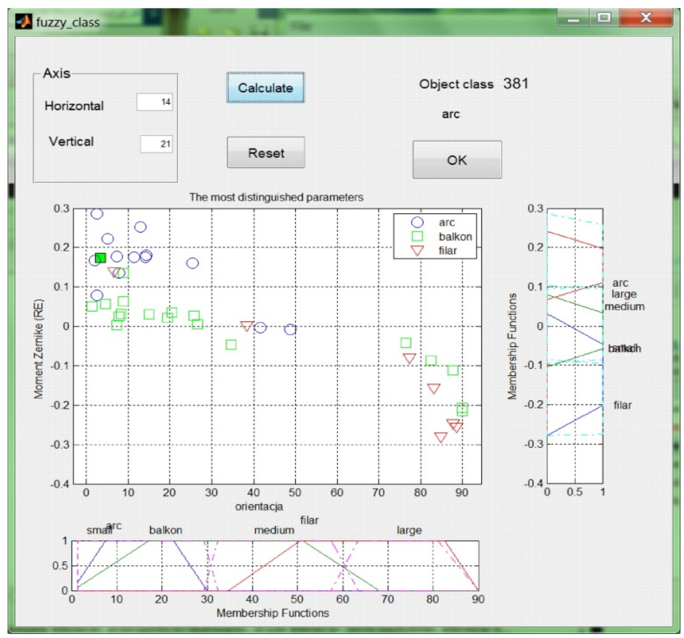

Figure 3. Classification example. The new element marked by the full green square is recognized as an arc among classes: arc, pillar and balcony. Membership functions are represented by solid colour lines and linguistic intervals are drawn in dashed lines.

***S*₂: fuzzy rule-based classifier with nine fuzzy rules**

R_1: If x_1 is *small* and x_2 is *small* then non-defined with 1.0,
R_2: If x_1 is *small* and x_2 is *medium* then balkon with 1.0,
R_3: If x_1 is *small* and x_2 is *large* then arc with 1.0,
R_4: If x_1 is *medium* and x_2 is *small* then non-defined with 1.0,
R_5: If x_1 is *medium* and x_2 is *medium* then balkon with 1.0,

R_6: If x_1 is *medium* and x_2 is *large* then non-defined with 1.0,
R_7: If x_1 is *large* and x_2 is *small* then filar with 1.0,
R_8: If x_1 is *large* and x_2 is *medium* then non-defined with 1.0,
R_9: If x_1 is *large* and x_2 is *large* then non-defined with 1.0.

The winner is the rule for which the product operator is maximum (cf. (2)), as follows:

$$\mu_{R_3}(\mathbf{x}_p) = \mu_{small}(\mathbf{x}_1) \times \mu_{large}(\mathbf{x}_2) =$$
$$= \mu_{small}(3.2383) \times \mu_{large}(0.1806) = 1 \times 1 = 1$$

The fuzzy rule-based classifier is stable, irrespective of attribute selection. Fig. 4 presents a classification of the next object. In this case, there are different attributes (x_1 is orientation and x_2 is the real part of Zernike's moment) and there are different classes in comparison with the example from Fig. 3. Hence, in this case, there is the construction of classifier S_3 as follows:

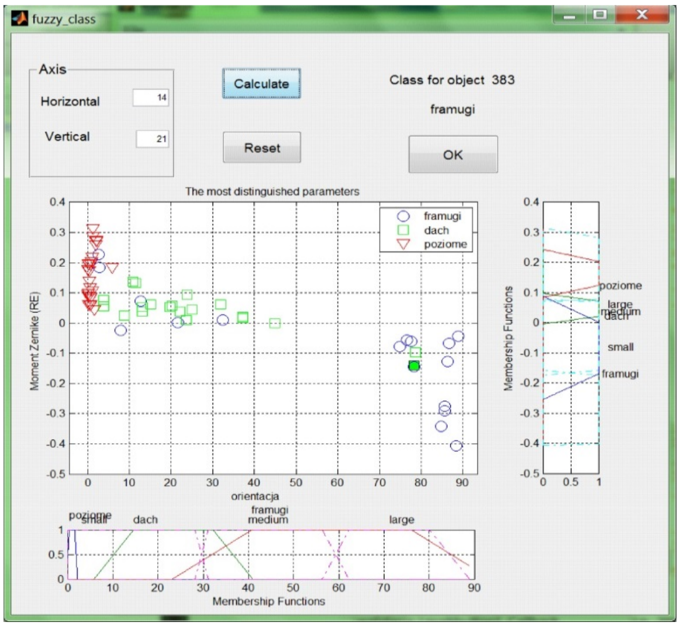

Figure 4. Classification example. The new element marked by the full green square is recognized as a frame among classes: frame, horizontal lines and roof. Membership functions are represented by solid colour lines and linguistic intervals are drawn in dashed lines.

S_3: fuzzy rule-based classifier with nine fuzzy rules

R_1: If x_1 is *small* and x_2 is *small* then non-defined with 1.0,
R_2: If x_1 is *small* and x_2 is *medium* then roof with 1.0,
R_3: If x_1 is *small* and x_2 is *large* then horizontal lines with 1.0,
R_4: If x_1 is *medium* and x_2 is *small* then non-defined with 1.0,
R_5: If x_1 is *medium* and x_2 is *medium* then roof with 1.0,
R_6: If x_1 is *medium* and x_2 is *large* then non-defined with 1.0,
R_7: If x_1 is *large* and x_2 is *small* then frame with 1.0,
R_8: If x_1 is *large* and x_2 is *medium* then frame with 1.0,
R_9: If x_1 is *large* and x_2 is *large* then non-defined with 1.0.

We treat a fuzzy rule-based classifier as a "decisive vote" in the case of differences between Euclidean, decision tree or Naïve Bayes classifiers.

IV. Results

Our learning set consists of 672 objects, which gives about 20 objects per each of the 32 classes. Based on it we classified 2547 new objects of all classes and we obtained the total precision of 21.5% for the similarity to the pattern algorithm, 68.6% for decision trees, 53,8% for the Naïve Bayes classifier and 88% for the fuzzy rule-based classifier FRBC (see Tab. I).

TABLE I. Classification Precision

Precision	Similarity to pattern	Naïve Bayes	Decision trees	FRBC
Total (for 32 classes)	**21.5%**	**53,8%**	**68.6%**	**88%**
Window-pane	16.1%	31,3%	72%	89.7%
Window	46.7%	82,2%	61%	57.6%
Brick wall	9%	32%	45.5%	90.9%
Arc	63.6%	65%	68.2%	58%
Roof edge	8.4%	61,4%	86.7%	93.9%

The high rate of false classification in the similarity to pattern algorithm results from extensive aggregation of information. Although the weights are used, all the features are involved in the eventual class assignment, whereas in the case of trees, only the most informative features are selected. Our classification process is divided into four trees due to the number of meta-classes which also reduces error rates.

The FRBC has the easiest job because it is used to distinguish only among tree classes that have been pre-classified earlier by the three simpler classifiers.

An additional problem, which we avoided in the learning set construction, arises from imbalanced classes. In proper classification, however, it is inevitable.

V. Example of Using Classification for the Spatial Object Location

In our system spatial object location in an image is used as the global feature. Firstly, it is easy for the user to recognize this spatial location visually. Secondly, it supports full identification based on rules for location of graphical elements. 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 in the middle of the sea, and so on. Thirdly, object location reduces the differences between high-level semantic concepts perceived by humans and low-level features interpreted by computers.

For the comparison of the spatial features of two images an image I_i is interpreted as a set of n objects composing it:

$$I_i = \{o_{i1}, o_{i2}, \ldots, o_{in}\} \quad (4)$$

Each object o_{ij} is characterized by a unique identifier and a set of features discussed earlier. This set of features includes a centroid $C_{ij} = (x_{ij}, y_{ij})$ and a class label L_{ij} indicating the class of an object o_{ij} (such as window, door, etc.), identified in the process described in [20]. For convenience, we number the classes of the objects and thus L_k's are just numbers.

Formally, let I be an image consisting of n objects and k be a number of different classes of these objects, $k \leq N$, because usually there are some objects of the same type in the image, for example, there can be four windows in a house.

Let us assume that there are, in total, M classes of the objects recognized in the database, denoted as class labels $L_1, L_2, \ldots, L_M$. Then, by the signature of an image I_i (4) we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \ldots, \text{nobc}_{iM}] \tag{5}$$

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}.

Additionally, for an image I_i we consider a representation of spatial relationships of the image objects. The objects' o_{ij} mutual spatial relationship is calculated based on the algorithm below. Now we consider one image; let C_p and C_q be two object centroids with $L_p < L_q$, located at the maximum distance from each other in the image, i.e.,

$$\begin{aligned} &\text{dist}\,(C_p, C_q) = \\ &= \max\{\text{dist}\,(C_i, C_j)\ \forall i,j \in \{1,2,\ldots,k\} \text{ and } L_i \neq L_j\} \end{aligned} \tag{6}$$

where: dist(•) is the Euclidean distance between two centroids (see fig. 5 middle). The line joining the most distant centroids is the line of reference and its direction from centroid C_p to C_q is the direction of reference for computed angles θ_{ij} between other centroids. This way of computing angles makes the method invariant to image rotation.

Hence, we received triples (L_i, L_j, θ_{ij}) where the mutual location of two objects in the image is described in relation to the line of reference (see Fig. 5 middle). Thus, there are $T=m(m-1)/2$ numbers of triples, generated to logically represent the image consisting of m objects. For a set of all triples we apply the concept of principal component analysis (PCA) proposed by Chang and Wu [6] and later modified by Guru and Punitha [14]) to determine the first principal component vectors (PCVs). The examples of results are shown in Tab. II.

TABLE II. EXAMPLES OF PRINCIPAL COMPONENT VECTORS

Image name	Component 1	Component 2	Component 3
House-front	-0,001786	-0,003713	0,999992
Domy-banino-1	0,000206	0,003988	0,999992
Houselawn I_1	0,000388	0,001869	0,999998
Houselandscape I_2	0,004109	0,001557	0,999990

VI. CONCLUSIONS

The results presented here seem to be encouraging to move forward to the next stages of the CBIR system preparation, namely, to the GUI and the completion of the search engine. The methods already implemented will be also evaluated in terms of the addition of new classes to the system. GUI development will also enforce introducing subclasses to some of the most numerous classes.

Intensive computational experiments are under way in order to draw some conclusions regarding the choice of parameters for the search engine. The results we have obtained so far, using the simplest configuration, are quite hopeful.

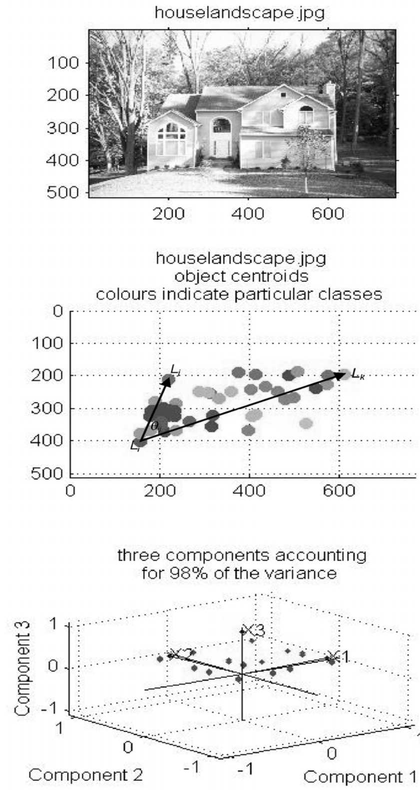

Figure 5. The main stages of the PCV applied to determine the unique object spatial location in an image

As for the prospects for future work, the introduction of iterative classification in respect of object neighborhood should be implemented, as well as the option of the presentation of similar objects to the user's previous choice.

If the classification precision is found insufficient, we will have to apply fuzzy decision trees [4] or other more sophisticated methods.

APPENDIX A
COMPARISON OF REAL AND COMPLEX FEATURE VECTORS

Technically, patterns can be created in different ways. The simplest method is the calculation of the average value of each component of a feature vector x_i. The subsets of objects used to define particular patterns are also used as learning subsets. In order to compare the object vector with a pattern we apply the Euclidean metrics:

$$D(\mathbf{x}, P_k) = \sqrt{\sum_{i=1}^{r} \xi_{P_k}(x_i) \left| \mathrm{x}(x_i) - P_k(x_i) \right|^2} \tag{7}$$

where: k– pattern number, $1 \le i \le n$ and weights $\xi_{P_k}(i) = \sigma(i) / \overline{x}(i)$ (where σ – standard deviation and $\bar{x}$ - mean value for each feature). Weights reflect the dispersion of each feature in the subset selected as a pattern (P_k). All the pattern vectors are normalized. A new object is classified to a class for which D is the minimum [20].

Another problem which we encountered when we built a classifier for this set of data was the existence of complex features. As we mentioned above, Zernike's moments are complex features, hence to obtain the real weight we apply the formula [1]:

$$\xi(i) = \sqrt{\frac{\sigma_{\mathrm{Re}}^2 + \sigma_{\mathrm{Im}}^2}{\overline{x}_{\mathrm{Re}}^2 + \overline{x}_{\mathrm{Im}}^2}} \qquad (8)$$

where standard deviations and means are calculated separately for real and imaginary parts of complex moments.

For all these classes we have created a pattern library (also stored in the DB) which contains information about pattern types, weights and objects belonging to learning subsets [20].

REFERENCES

[1] A. Albert, L. Zhang, "A novel definition of the multivariate coefficient of variation", Biomedical J., Vol. 52 (5), Wiley-VCH, pp. 667–675, 2010.

[2] J. M. Ali, "Content-Based Image Classification and Retrieval: A Rule-Based System Using Rough Sets Framework", Chap. IV, Artificial Intelligence for Maximizing Content Based Image Retrieval, Zongmin Ma (ed.), New York, pp. 68-82, 2009.

[3] F. Berzal, J. C. Cubero, J. Kacprzyk, N. Marin, M. A. Vila, S. Zadrożny, "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, (Suppl.), Feb., World Scientific Publishing Company, Singapore, pp. 111-131, 2007.

[4] M. Boushon-Meunier, Ch. Marsala, "Fuzzy decision tree and databases". Chap. 14, Flexible Query Answering Systems, T. Andreasen et al. (eds.), Kluwer Academic Publisher, pp. 277-288, 1997.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, "Classification and Regression Trees," Chapman and Hall, New York, 1984.

[6] C. C. Chang, T. C. Wu, "An exact match retrieval scheme based upon principal component analysis," Pattern Recognition Letters 16, pp. 465-470, 1995.

[7] J-J. Chen, Ch-R Su, W.E.L. Grimson, J-L Liu, and D-H. Shiue, "Object Segmentation of Database Images by Dual Multiscale Morphological Reconstructions and Retrieval Applications," IEEE Transactions on Image Processing, vol. 21, no. 2, pp. 828-843, 2012.

[8] M. Cho, J. Lee, and K. M. Lee. "Feature correspondence and deformable object matching via agglomerative correspondence clustering." EEE International Conference on Computer Vision (ICCV), pp. 1-8, 2009.

[9] J. C. Cubero, N. Marin, J. M. Medina, O. Pons, M. A. Vila, "Fuzzy Object Management in an Object-Relational Framework," Proceedings of the 10th International Conference IPMU, Perugia, Italy, pp. 1775-1782, 2004.

[10] B. V. Dasarathy, (ed.) "Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques," IEEE Computer Society Press, Los Alamitos, Calif., 1991.

[11] S. Deb, (ed.)., "Multimedia Systems and Content-Based Image Retrieval," Chap. VII and XI, IDEA Group Publishing, Melbourne, pp. 182-199, 2004.

[12] U. M. Fayyad and K. P. Irani "The attribute selection problem in decision tree generation," Proceedings of the 10th National Conference on artificial Intelligence, AAAI, vol. 7, pp. 104-110, 1992.

[13] B. Fei, J. Liu, "Binary Tree of SVM: A New Fast Multiclass Training and Classification Algorithm," IEEE Transaction on Neural Networks, Vol. 17, no. 3, pp. 696–704, 2006.

[14] D. S. Guru, P. Punitha, "An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis," Pattern Recognition Letters, Vol. 25, pp. 73 -86, 2004.

[15] A. Hamilton-Wright, D. W. Stashuk, "Constructing a Fuzzy Rule Based Classification System Using Pattern Discovery," In NAFIPS 2005 Annual Meeting of the Narth American Fuzzy Information Processing Society, IEEE, pp. 460-465, 2005.

[16] H. Ishibuchi and Y. Nojima, "Toward Quantitative Definition of Explanation Ability of fuzzy rule-based classifiers," IEEE International Conference on Fuzzy Systems, June 27-39, Taipei, Taiwan, pp. 549-556, 2011.

[17] H. Ishibuchi, T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," IEEE Transactions on Fuzzy Systems, Vol. 13, no. 4, pp. 428-435, 2005.

[18] T. Jaworska, "Object extraction as a basic process for content-based image retrieval (CBIR) system," Opto-Electronics Review, Association of Polish Electrical Engineers (SEP), Vol. 15. No. 4, Warsaw, Poland, pp. 184-195, 2007.

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

[20] T. Jaworska, "A Search Engine Concept Based on Multi-feature Vectors and Spatial Relationship," LNAI Vol. 7022, Springer, pp. 137-148, 2011.

[21] Y. Liu, D. Zhang, G. Lu, W-Y. Ma, „A survey of content-based image retrieval with high-level semantics," Pattern Recognition, Vol. 40, Elsevier, pp. 262-282, 2007.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[23] K. Nozaki, H. Ishibuchi, H. Tanaka, "Adaptive fuzzy rule-based classification systems," IEEE Transactions on Fuzzy Systems, Vol. 13, no. 4, pp. 238-250, 1996.

[24] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, "Fast Wavelet-Based Image Characterization for Highly Adaptive Image Retrieval," IEEE Transactions on Image Processing, vol.21, no. 4, pp. 1613 - 1623, 2012.

[25] I. Rish, "An empirical study of the Naive Bayes classifier," Proceedings of IJCAI-2001 workshop on Empirical Methods in AI, pp. 41-46, 2001.

[26] M. R. Teague, "Image analysis via the general theory of moments," JOSA, Vol. 70, ed. 8, pp. 920-930, 1980.

[27] L. A. Zadeh, "Fuzzy sets", Information and Control, vol. 8, no. 3, pp. 338-353, 1965.

[28] G. P. Zhang, "Neural Networks for Classification: A Survey," IEEE Transactions on Systems, Man and Cybernetics, part C: Applications and reviews, Vol. 30, no. 4, pp. 451-462, 2000.

[29] L Zhang, L.P. Wang and W. Lin, "Conjunctive patches subspace learning with side information for collaborative image retrieval," IEEE Trans. Image Processing, vol. 21, no. 8, pp. 3707-3720, 2012.

[30] L Zhang, L.P. Wang and W. Lin, "Geometric Optimum Experimental Design for Collaborative Image Retrieval", IEEE Transactions on Circuits and System for Video Technology, vol. 24, pp. 346-359, 2014.

[31] L Zhang, L.P. Wang and W. Lin, "Semi-supervised biased maximum margin analysis for interactive image retrieval," IEEE Trans. Image Processing, vol. 21, no. 4, pp. 2294-2308, 2012.

[32] L Zhang, L.P. Wang and W. Lin, "Generalized biased discriminant analysis for content-based image retrieval," IEEE Trans. System, Man, Cybern, Part B - Cybernetics, vol. 42, no. 1, pp. 282-290, 2012.