An Approach for Image Organization and Retrieval in Realistic Image Databases

Boris Rachev[1], Irena Valova[2], Silyan Arsov[2]
[1]Technical University of Varna, 1, Studentska Str., 9010, Varna, Bulgaria
[2]Technical University "Angel Kanchev", 8, Studentska Str., 7017, Rousse, Bulgaria

Abstract
One of the basic problems in the creating of the Realistic Image Databases (RIDB) structure is the efficient organization of the image retrieval. As a result of the analyses of the retrieval strategies we made, incl. GIS realistic images, we can make the conclusion that the solution of the problem for effective image retrieval is the approach for image sorting or indexing. Text- and content-based indexing methods play different roles in representing image information. The systems for image organization and image retrieval use more and more information from the image visual content instead of being manually annotated by text-based keywords or other relatively simple characteristics from the image, especially from the realistic GIS coverages. The basic characteristics of the image visual content are colour, texture and shape features. The colour feature is probably the most visible visual feature for the most humans.
In this work we propose a development of the usage of the colour as a image feature for *an automatic indexing by image contents*. The process of the indexing is based on the idea of the predominant colours in an image. This approach combines some advantages of the classical methods such as systematically classifications, the organization and retrieval methods in the relational database systems and the new approaches in the organisation and retrieval in image databases. Our approach combines the representation of the colour as the global image feature and the colour as the representation of the spatial information in the images. We propose an image retrieval system that provides users with content-based methods, which contain *automatic indexing capability* and *easy-to-use visual query facilities, based on the artificial intelligence Image Manipulation Language*. It is expected that users may use these proposed content-based facilities to perform specific retrieval tasks without the use of manual indexing and text-based methods

Keywords: Data Bases, Image Databases, Realistic Image Databases, GIS, Content-Based Image Retrieval, Image DBMS.

1. Introduction

Our world is dominated by visual information and a tremendous amount of such information is being added day-by-day. It would be impossible to cope with this explosion of visual data, unless they are organized such that we can retrieve them efficiently and effectively.
The main problem in organizing and managing such visual data is indexing, the assignment of a synthetic descriptor which facilitates its retrieval. It involves extracting relevant entities or characteristics from images as index keys. Then a representation is chosen for the keys and specific meaning is assigned to it.
Visual database systems require efficient indexing to facilitate fast access to the images in the database. Recent Content-Based Image Retrieval (CBIR) techniques cited in the literature [1], [3,4,6,9,8,10] are based on features such as colour, texture, shape, spatial relationships, object motion, etc. As the number of digital images grows, there is a need for automatic image retrieval.
Colour is an important cue for image retrieval. Many CBIR systems have been designed with color as the main feature for retrieval [5,7]. Though a global property, its distribution is inde-

pendent of view and resolution and does not require knowledge of component objects of the image.
In color indexing, given a query image, the goal is to retrieve all the images whose color compositions are similar to the color composition of the query image. Typically, the color content is characterized by color histograms, which are compared using the histogram intersection distance measure.

2. Definition of the Problem

The general computational framework of a CBIR system is depicted in Fig.1. The entire process starts with the construction of an image database. The images to be added to the database are processed by a feature extraction algorithm. The output of this algorithm is a feature representation, which is the data structure actually stored in the database and used to compute similarity. The same feature extraction algorithm is used to process the query image and the images contained in the database. Hence, the same feature representation is computed for the query image as was for each image in the database. The similarity measure then compares the query feature representation with each of the feature representations in the database. Those feature representations deemed "similar" are returned to the user as a result set. It is not strictly necessary that an image be specified as part of the query.
Queries can be specified by sketches or by graphical user interface tools. However, the ultimate result of the query specification must be the same feature representation that is used by the database to store and index images. The specification of the query can be with an example image, a user drawn sketch, or explicit information from the user about the primitive features of interest.
As a result of our research we found that we need a ***new and more effective method for storage and retrieval in databases of realistic images, based on the color features of the images***. Also we need a ***new appropriate limited natural language to specify some types of queries.***

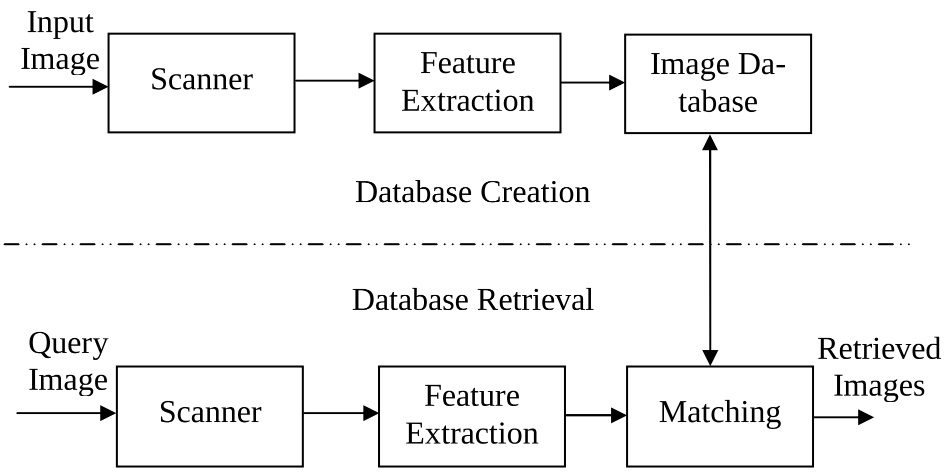


Figure 1. General Computational Framework of a CBIR System

3. Solution of the Problem

Colour is by far the most common visual feature used in CBIR, primarily because of the simplicity of extracting color information from. Colour histograms describe the distribution of pixels of each color in the color space of the image. A commonly accepted color space from which color histograms are represented does not exist. Despite the many color spaces available, the perception of color is subjective. This lack of a commonly accepted color space makes the comparison of the different CBIR methods difficult to compare. Many methods use the Red-Green-Blue (RGB) colour space; at least as a starting point, since image pixel values are commonly stored as RGB 3-tuples. However, the RGB colour space lacks perceptual uniformity, and, hence, it is rarely used as the final color space for color-based feature extraction methods. Because it is impossible to classify all the colors and to store the color for every pixel from the image we select nine basic colors (see Table 1) according to the presented by *Brent Berlin* and *Paul Kay* in 1969, study of *Basic Colour Terms.* They presented eleven basic colour categories, which are white, black, red, green, yellow, blue, brown, purple, pink, orange, and grey.

Table 1

Colour Descriptor	Colour Mapped
0	Uncertain Colours: “very dark” or “very bright”
1	White
2	Grey
3	Black
4	Red, Pink
5	Brown, Dark Yellow, Olive
6	Yellow, Orange, Light Yellow
7	Green, Lime
8	Blue, Cyan, Aqua, Turquoise
9	Purple, Violet, Magenta

To store the color image features we propose ***two type index structures: index key - for the global color features and index matrix - for the spatial information in every image***.
The processes of the image processing operations are illustrated in Fig. 2 and are described as follows.

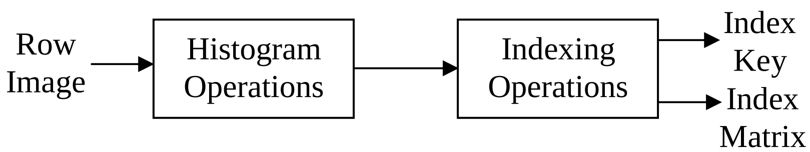


Figure 2. The Processes of Image Processing Operations

The first stage of the scheme is to generate *the histogram of the perceptual colour descriptors for the image*. The histogram is a list of "bins" showing the number of pixels being classified into the different perceptual colour groups.

3.1. Index key – representing the global features

$$IndexKey = [C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9],$$

where C_i is color descriptor for the i color from the perceptual color model in Table 1.

To generate the Index Key the histogram can be used to indicate the different counts for each colour descriptor in an image, the perceptual colour group, which has the largest count in the histogram, may be regarded as the most dominant colour group. Thus the less dominant perceptual colour group should contain the lesser amount in the histogram (see Fig. 3).

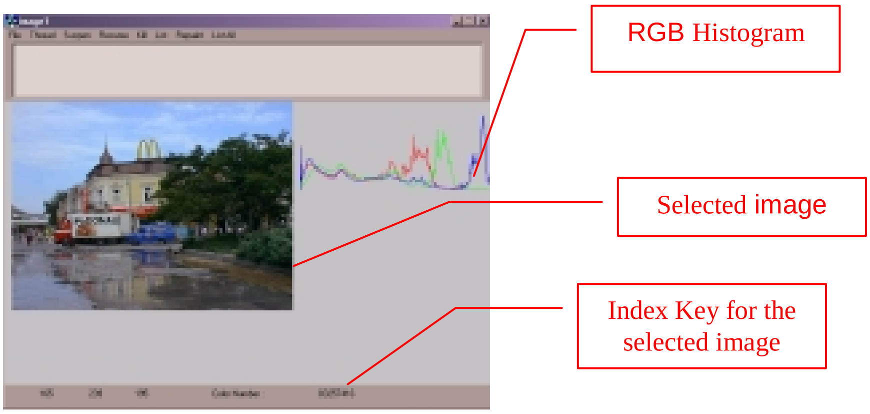


Figure 3. Append new image in the Image Database or select an Image for the Query-by-Image-Example

This structure can store the color features of the images but it does not contain any spatial information. Due to these limitations, which are important for some type of queries, we propose another index structure – index matrix.

3.2. Index Matrix - Representation of Spatial Information

In order to create this index structure the whole image is divided into 256 equal parts. In this index matrix is stored the coefficient of the dominant color in the corresponding part. The original images were 16x16 quantised and were represented as16x16 color blocks (see Fig.4).

$$IndexMatrix = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & \dots & C_{0,15} \\ C_{1,0} & C_{1,1} & & & \dots \\ \dots & & & & \dots \\ \dots & & & & \dots \\ C_{15,0} & \dots & \dots & \dots & C_{15,15} \end{bmatrix},$$

where $C_{i,j}$ is color descriptor for the dominant color in the i,j element of the image

Currently, the size of the Index Matrix, 16x16, was suggested according to the informal investigation and also referred to the previous work of Harmon, 1973; Harmon and Julesz, 1973. Harmon 1973, p.73 stated, "Our informal investigation revealed that a spatial resolution of 16 x 16 squares was very close to the minimum resolution that allows identification."

Further study on the minimum size of the index matrix could be explored to achieve the minimum storage.

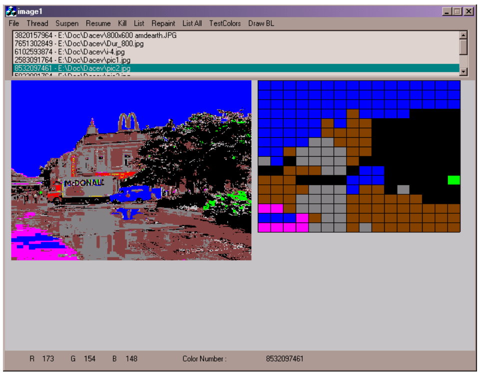


Figure 4. The quantised image

3.3. Image indexing processes

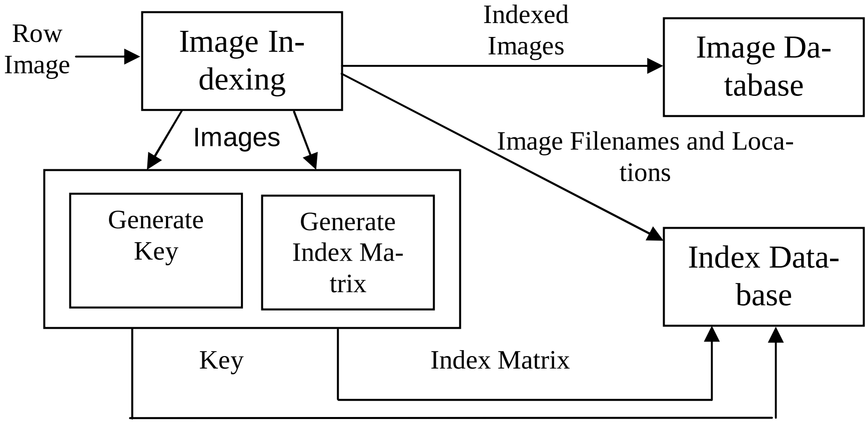
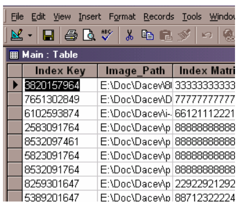


Figure 5. The Processes of Image Indexing

The storage of the index data is currently in a "table-based" structure, which is supported by currently Image DBMS (IDBMS). Such design provides the potential benefits of using the existing features of Relational DBMS, such as the use of SQL. Specifically, it means that the "index database" (see Fig. 5) is based on a RDBMS. For every image it is stored in the table the two types of indexes, the name of the file and the path, some metadata about the image with keywords (see Fig. 6).



Main : Table

Index Key	Image_Path	Index Matri
3820157964	E:\Doc\Dacev\8I	33333333333
7651302849	E:\Doc\Dacev\D	77777777777
6102593874	E:\Doc\Dacev\i-	66121112221
2583091764	E:\Doc\Dacev\p	88888888888
8532097461	E:\Doc\Dacev\p	88888888888
5823091764	E:\Doc\Dacev\p	88888888888
8532091764	E:\Doc\Dacev\p	88888888888
8259301647	E:\Doc\Dacev\p	22922921292
5389201647	E:\Doc\Dacev\p	88712322224

Figure 6. Index Database - structure

The storage of the image data currently preserves the original "file-based" formats, and the original "directories" (or "folders"), which may be arranged by users themselves. Specifically, it means that the "Image Database" (see Fig. 5) is based on "image files".

3.3. Queries

To develop query facilities for retrieving images by accessing the Index matrix of the images in the RIDB, users construct a Query matrix through easy-to-use query facilities. We used algorithms (1) of similarity measures to perform the degree of similarity between the Query matrix and the Index matrix, which provides a ranking to arrange the order of images in the result set. Then let:

$Similarity(IB,Q) = |IB \cap Q|$, it is the simple matching coefficient (1)

$Similarity(IB,Q) = \frac{|IB \cap Q|}{|IB \cup Q|}$, it is Jaccard's coefficient,

where IB is the set of colour descriptors representing the index matrix of the images in the image database, and

Q is the set of colour descriptors representing the index matrix of the query.

The Equation (1) can be used to represent the semantic that the number of colour descriptors matched in the Index matrix and the Query Matrix, which is constructed by users using query facilities.

For example *Measures of Image Query – "Query-by-Image-Example"*:

If we have the matrix (for the example we choose 4X4 matrix to simplify the calculations):

$$IndexMatrix = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 5 & 4 \\ 1 & 5 & 4 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad QueryMatrix = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 5 & 3 \\ 1 & 5 & 4 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix}$$

In accordance with (1) we have:

Simple matching coefficient=12 (12 colour coefficients are equal in the two matrixes)

Jaccard's coefficient = 12/16=0.75 (represent the ratio between the count of the equal coefficients in the two matrixes and the total count of the coefficients in the matrix – now 16, but really in the system 256; in our system we transform for the end user this coefficient in %)

And for *"Query-by-User-Construction"*:

If we have the matrix:

$$IndexMatrix = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 5 & 5 & 4 \\ 1 & 5 & 4 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad QueryMatrix = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 5 & -1 & 3 \\ 1 & 5 & 4 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix}$$

(-1 means the colour descriptor is not available – user in his sketch does not define it)

Simple matching coefficient = 9 (9 colour coefficients are equal in the two matrixes)

Jaccard's coefficient = 9 / 16 = 0.56

The current method of ranking is to rank the Jaccard's coefficient for similarity measures in the descending order. In other word, the lager amount the Jaccard's coefficient is, the higher the rank is (see Fig. 7).

3.4. Limited Natural language queries

We propose a ***Limited Natural Query Language for the queries definition*** to the RIDB and entering to a ***special developed preprocessor***. It includes basic elements, which are shown in Fig. 8. When it is impossible to construct the query by image example, the module for the *User-Database Interaction By The Limited Natural Language Queries* is very useful [2].

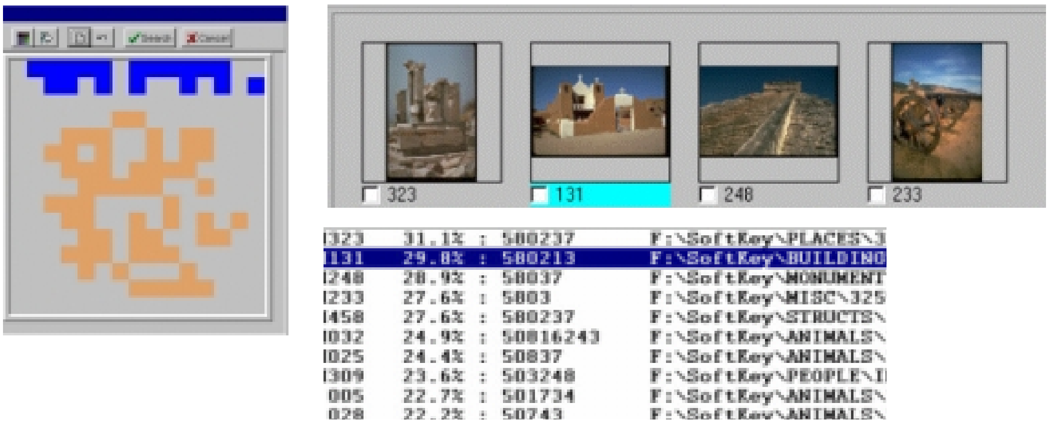


Figure 7. Query-by-User-Construction

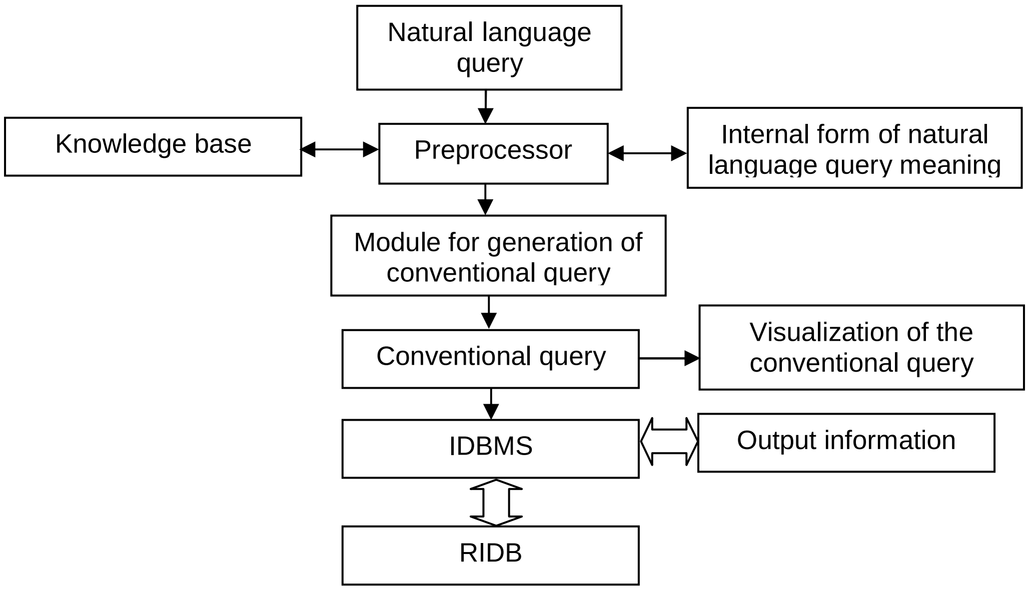


Figure 8. Block diagram of a natural language query processing system

- Module for analysis. It adapts the input query in internal form of her meaning. To this end, this module uses the available knowledge base – dictionaries, syntactic, semantic and pragmatic characteristics of lexical units used in the query.
- Module for comprehension. It basic function is to interact with the knowledge base and to transform the internal form of meaning of the natural language query in internal form of conventional query.
- Module for generation. It generates a query into selected IDBMS, using the internal form, formed by the comprehension module.

The knowledge base includes dictionaries with the names of the attributes and their meaning in the natural language, in which the queries are formulated; the dictionaries with the names of relations and their respective logical operation in the conventional query language and relationship including the name of relations between the basic relation, in which the data is saved.

Once the preprocessor is formed a standard query, it is manipulated from the respective environment for query processing of the selected IDBMS. IDBMS interacts with the database. For chose IDBMS and database it is necessary to formulate knowledge base for query maintenance. The output user information is visualized on the screen.

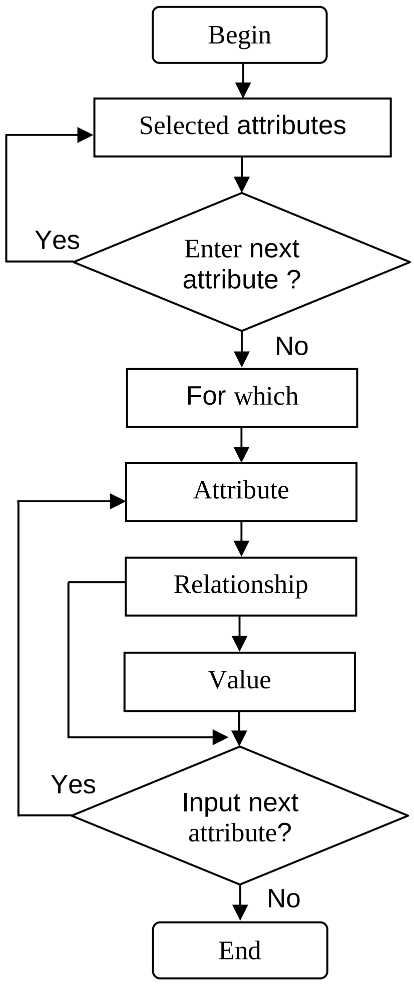


Figure 9. Diagram of natural language query forming

The sentence may to begin with the question word "Who" or with the word "Select" (see Fig. 9). After them the attribute values of which will complete the selection in database, must be selected. After the selection of the first attribute it is possible to input next. If there is no next attribute follows a word "for which". After this connecting unit of the speech, there must be selection of attribute, which will realize the filtration of database. After this attribute the user has to choose relationship between the attribute name and the value, which is seek. After the value selection the sentence is completed, but it is possible to continue and give a final form with the logical operations "or" or "and". Using this method the user can formulate the natural language query by the any DBMS to any database (see Fig. 10).

After the query is formulated it is transformed to the corresponding Query Matrix according to the selected color model and the selected color measures in the limited natural language query. As the result we have the query by user construction and it is executed (see Fig. 9 and Fig. 10).

4. Conclusions

First, the generation of the Key may have a problem that those images visually similar to each other may have close but different Keys with different orders of colour descriptors, since the histograms of those image are only slightly different.

Second, although an image can be reasonably represented by the index, which contains the spatial information of the image, the algorithm of similarity measures would play an important role in image retrieval for using these indexes. However, the current algorithm of similarity measures is only to match the absolute positions of colour descriptors; the relative positions of spatial objects in photos will not be considered.

Third, everyone may have different colour perception and it is ideal to allow users to choose their preferences in the definition of the model, if a suitable user interface is given. Therefore, further research is needed to explore this issue.

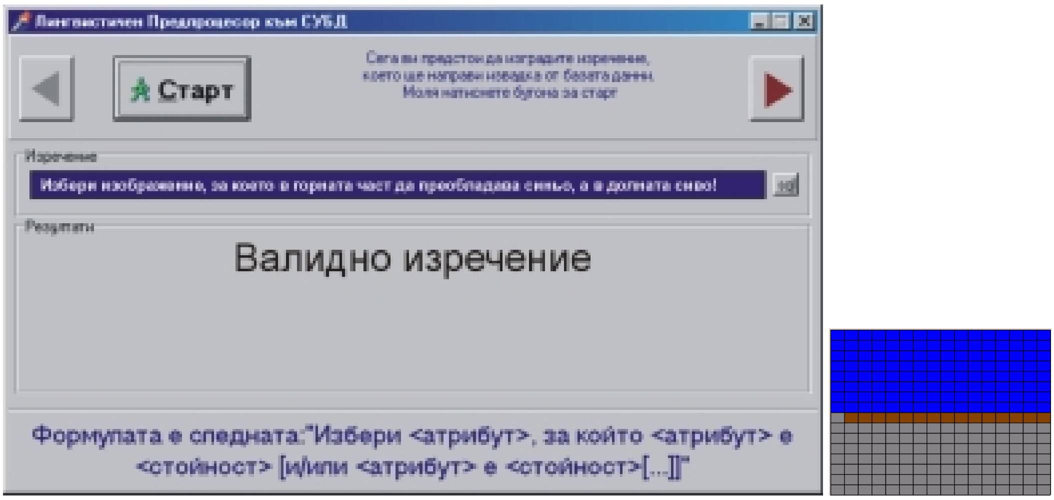


Figure 10. Example for Limited Natural Language Query window and the corresponding query matrix in the right. The Query is: ***Select image where the up part is predominantly blue and down part is predominantly gray*** *(in Bulgarian).*

Fourth, retrieval process from databases with the Limited Natural Language queries is realized on the base of the enhanced data model "entities –relationships – attributes" [2], which was specifically developed for the RIDB implementation. On the base of this model and with the update of the knowledge base it is possible to retrieve data from RIDB.

5. Acknowledgements

This work includes some proposals, which develop the results of INCO Copernicus Project URBAN 960252.

6. References

1. Aigrain, P., Zhang, H.J., Petkovic, D., Content-based representation and retrieval of visual media: A state-of-the-art review, *Multimedia Tools and Applications*, No 3 PP. 179–202, 1996.
2. Arsov, S., Rachev, B., Graphical Specification of Conceptual Database Schemes Using Modified Model "Entity-Relationship", *Proceedings of the CompSysTech'2000*, Sofia, Bulgaria, 22-23 June, PP. III.8.1-III.8.8, 2000.
3. Gray, R.S., Content-based image retrieval: color and edges. *Technical Report PCS-TR 95-252*, Dartmouth College, 1995.
4. Gudivada, V., Raghavan, V.V., Special issue on content based image retrieval systems - guest eds, *IEEE Compute*r, No 28(9):PP. 18–22, 1995.
5. Kankanahalli, M.S., Mehtre, B.M., Wu J.K., Cluster-based color matching for image retrieval, *Pattern Recognitio*n, No 29(4): PP. 701–708, 1996.
6. Marsicoi, M., Cinque, L., Levialdi, S., Indexing pictorial documents by their content: a survey of current techniques, *Image and Vision Computin*g, No 15(2): PP. 119–141, 1997.
7. Mehtre, B.M.et al., Color matching for image retrieval, *Pattern Recognition Letter*s, 16:325–331, 1995.
8. Niblack W. et al, The QBIC project: Querying images by content using color, texture and shape, *In Storage and Retrieval for Image and Video Databases (SPIE*), 1908: PP. 173–187, 1993.
9. Pentland, A., Picard P.W., Sclaroff, S., Photobook: Content-based manipulation of image databases, *International Journal of Computer Visio*n, No 18(3): PP. 233–254, 1996.
10. *Rachev, B., Valova, I., Stoeva, M.*, Image Databases Organization and Data Models, *Proceedings of the CompSysTech'2000*, Sofia, Bulgaria, 22-23 June, PP. III.9.1-III.9.9, 2000.