The inner structure of database for the CBIR system

Tatiana Jaworska
Systems Research Institute,
Polish Academy of Sciences
6 Newelska Street, 01-447 Warsaw, Poland
E-mail: Tatiana.Jaworska@ibspan.waw.pl

Abstract

In this article we focus on the presentation of the inner structure of the database for the Content-Based Image Retrieval (CBIR) system containing house images. The part devoted to image processing is signalled to the extent necessary for further database (DB) structure explanations to the reader. In this paper standard DB project methods in the Oracle 10g system have been used to generate a database for storage and retrieval of house images. The main body of the article covers the DB description and explanation of the information flow between Oracle and Matlab. For our system, Matlab is applied in image preprocessing as well as retrieval of images, which are answers to user's queries.

1. Introduction

Determining how to store images in large databases, and later, how to retrieve information from them, is an active area of research in many sectors of computer science, including graphics, image processing, information retrieval and databases.

Although attempts have been made to perform CBIR in an efficient way based on shape, colour, texture and spatial relations, it has yet to reach maturity. A major problem in this area is computer perception. In other words, there remains a big gap between retrieval images based on low-level features such as shape, colour, texture and spatial relations and retrieval images based on high-level semantic concepts such as windows, roofs, flowers, etc.

In the last 20 years, CBIR techniques have drawn much interest, and image retrieval techniques have been proposed in the context of searching for information in image databases. In the 90's the Chabot project at UC Berkeley [12] was initialized to study retrieval and storage of a vast collection of digitized images. Also, at IBM Almaden Research Centre CBIR was prepared with a very interesting query interface which enabled the users to define their own graphical queries [7, 11]. This approach allowed the users to find images on the basis of extraction of a set of features. These features were defined for the whole images. This approach was improved in later researches which made possible finding the foreground object in the image on the basis of colour or shape [14, 8]. There also exists a database in which shape is used as the main distinguishing feature, containing about 1100 images of marine creatures, created by F. Mokhtarian, S. Abbasi and J. Kittler [10] at the Department of Electronic and Electrical Engineering, University of Surrey, UK. Each image shows one distinct species in a uniform background. Every image is processed to recover the boundary contour, which is then represented by three global shape parameters and the maxima of the curvature zero-crossing contours in its Curvature Scale Space image.

Although much effort has been put to CBIR, many techniques have been proposed and many prototype systems have been developed [5], the problems with retrieving images according to image content are far from being solved. In the field of CBIR, new researches still tend to use the low-level features which are characteristic of the whole image. Other researches are based on the user's implicit feedback with the use of, for instance, the ostensive relevance approach [15].

In contrast with the approach mentioned above, our system takes into account not only low-level features but mutual localization of elements in the image as well. We also highlight the fact that the whole system, currently under construction, is intended to be entirely automatic. The purpose of this paper is to present the DB system which retrieves images based on individual elements of image, according to graphical user's query. The system consists of 3 main blocks:

1. the image preprocessing block (responsible for image segmentation) applied in Matlab;

2. the Oracle Database, storing information about whole images, their segments (here referred to as image objects), segment attributes and object localization;
3. the graphical user's interface on a WWW site.

2. CBIR conception overview

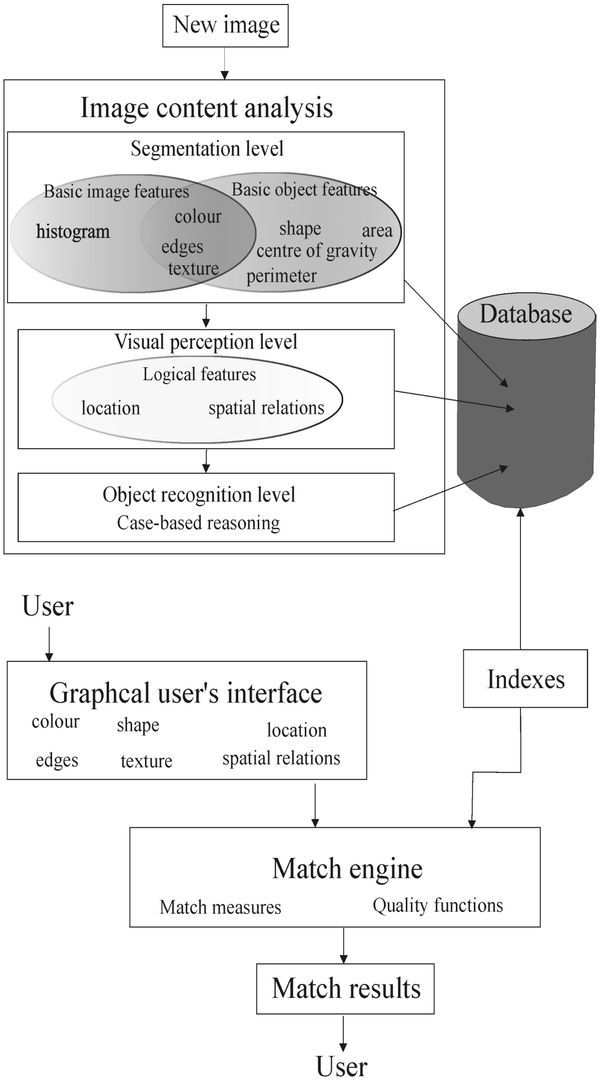

Figure 1. Block diagram of our content-based image retrieval system.

Our CBIR system is built to support estate agencies. In the estate database there are images of houses, bungalows, and other buildings. To be effective in terms of presentation and choice of houses, the system has to be able to find the colour image of a house with defined architectural elements, for example: windows, roofs, doors, etc. Images are downloaded from the Internet in the JPEG format. Figure 1 shows the block diagram of our CBIR system. As can be seen, the top part of the diagram illustrates the image content analysis block of our system. This block is implemented in Matlab (version 2008a) with the support of the following toolboxes: Image Processing, Fuzzy, Statistics, Wavelet and Database. The DB system is applied in Oracle Designer, version 6i and it is implemented in Oracle 10g, whereas the graphical user's interface is created in Perl for WWW users.

Each new image added to the CBIR system must be preprocessed, which is presented in the image content analysis block as a segmentation level frame. All key architectural elements (such as windows, doors, roofs, etc.) must be segmented and extracted from the background at the stage of preprocessing. In the situation when colour images are downloaded from the Internet and their preprocessing is unsupervised, the crucial issue is such an object extraction from the image background that further storage of these objects in the DB can be carried out without supervising.

For this purpose, we apply a two-stage segmentation, enabling us to extract in detail the desired objects from the image. In the first stage, the image is divided into separate RGB colour layers according to three light levels. In the second stage, individual objects (referred to as architectural elements of the house) are extracted from each layer. Next, the low-level features are counted for each object, understood as a fragment of the entire image. These features are: colour, area, centroid, eccentricity, orientation, texture parameters, moments of inertia, etc. The segmentation algorithm, object extraction algorithm, as well as texture parameters finding algorithm are presented in detail in an article by Jaworska [9]. Object storage in the DB takes place after counting the object low-level features and their logical features (juxtaposition, for instance) – details in sec. 3.

The bottom part of figure 1 is dedicated to users and it is processed on-line. Its first element is the graphical query interface block. In our system the user's interface is offered on a WWW page. If the users have a vague target image in mind, they have the possibility of composing their imaginary house and the system presents them with some optional houses based on these chosen elements [6].

The next element of the system is the matching engine, which uses a distance function between the features on the low-level [13], the dissimilarity measures on the visual perception level and case-based reasoning on the high-level to search for “the best matched images”. Retrieval results are presented by the user's interface.

3. The inner structure of database of the CBIR system

3.1.The basic information

The fundamental assumption of the relational model is that all data is represented as mathematical n-ary relations, an n-ary relation being a subset of the Cartesian product of n domains. The relational model of data permits the database designer to create a consistent, logical representation of information. The basic relational building block is the domain or data type. A tuple is an unordered set of attribute values. An attribute is an ordered pair of the attribute and type names. An attribute value is a specific valid value for the type of the attribute. This can be either a scalar value or a more complex type. A table is an accepted visual representation of a relation; a tuple is similar to the concept of a row, but note that in the database language SQL the columns and the rows of a table are ordered [4]. A database management system (DBMS) is a computer software designed for the purpose of managing databases. A DBMS can be characterized by:

1. A modelling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
2. The optimal structure depends on the natural organization of the application's data, and on the application requirements (which include transaction rate or speed, reliability, maintainability, scalability, and cost).
3. Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to a volatile main memory).
4. A database query language and report writer to allow users to interactively interrogate the database, analyse its data and update it according to the users' authorized access to data.
5. The security control unit of the database.
6. Data security preventing unauthorized users from viewing or updating the database. By using passwords, users are granted access to the entire database or its subsets called subschemas.
7. Providing a way to interactively enter and update the database, as well as interrogate it, which enables the user to manage their personal database.
8. Maintaining the integrity of the data in the database.
9. Maintaining the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicating records via unique index constraints.

In recent years more and more photographs, images and video films have been recorded in DB as well as GIS data understood as spatial objects.

For our purpose, of the many databases existing on the market we chose Oracle DB. We based our decision on the fact that Oracle DBMS deals with very large amounts of data, including multimedia data, as well as GIS data. This fact is very important with regards to our application.

All project stages of our DB were carried out in Oracle Designer, version 6i. We used the entity-relationship modeller for generating all schemas. The end-product of the modelling process is an entity-relationship diagram (ERD). ERDs were originally proposed by Pin-Shan Chen in 1976 [3]. In a modern form, an entity-relationship diagram (ERD) is a data model or diagram used to describe conceptual data models by providing graphical notations which document entities and their relationships, and the constraints that bind them. For our purpose, we will not introduce the whole entity terminology connected with entity types, but we shall only emphasise that herein an object is a graphical object understood as a fragment of an image obtained as a result of segmentation (not in the IT meaning).

The requirements analysis, which describes information needs as a preliminary step in the design of a database, has not been difficult because the input data received from the image processing system, as well as data which will be needed for the image retrieval process have been known before [1, 2].

3.2. The inner structure of database of the CBIR system

Image is the crucial information for the CBIR system. Hence, the CALY OBRAZ entity stores the most general information about the image, such as: image file name, disc path and the image itself in graphical form (BLOB). The entity-relationship diagram for our CBIR system is presented in fig. 2. According to the segmentation algorithm, an image is divided into colour layers, so the next entity, called KOLOROWA WARSTWA, stores information about: the layer number, its colour and the layer graphical image. The relationship between the above mentioned entities is of the many-to-one type (Mandatory-Mandatory). OBIEKT is the most capacious entity of all, containing information about parameters of graphical objects which were extracted from particular colour layers. Here there are the following attributes: object ID, object image, solidity, major axis, area of the convex hull, localization, average colour, area, centroid, eccentricity, orientation, texture parameters, moments of inertia and a number of other objects. In case there is a necessity to add more parameters, for instance shape descriptors, for object identification, we can easily increase the number of these entity parameters because Matlab generates as many as 24 features of a graphical object.

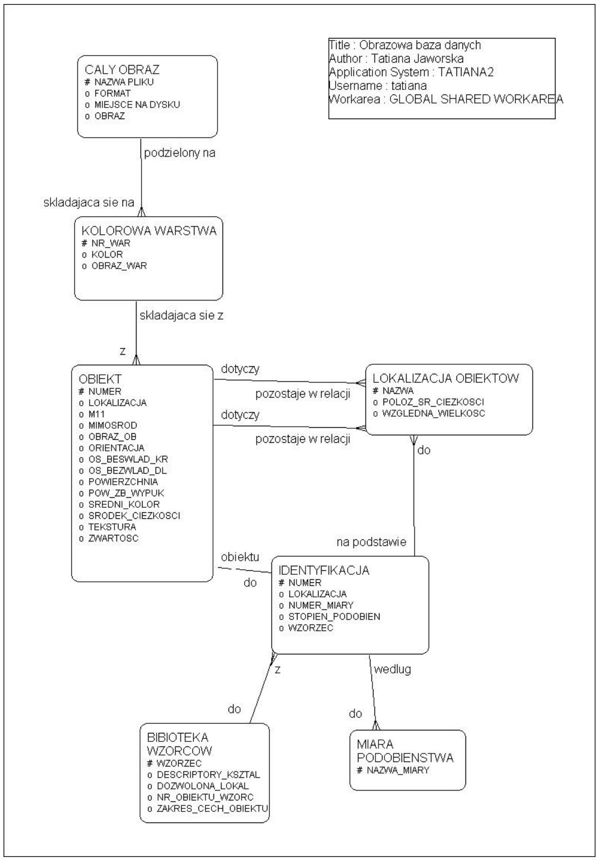

Figure 2. The entity-relationship diagram for our CBIR system.

The next entity focuses on an object location. Therefore, the LOKALIZACJA OBIEKTOW entity consists of information about the mutual location of two object centroids. For this reason, the entity has two relationships with the OBIEKT entity and, additionally, it has a connection with the IDENTYFIKACJA entity, which identifies objects in terms of human perception. Such object identification is based on object parameters and localization, as well as a comparison with the pattern library. The relationship between entities OBIEKT and IDENTYFIKACJA is the one-to-one type because object identification must be unique. Identification is understood as an assignment of each graphical object, consisting of a set of pixels (eg. a square or a triangle) to a pattern name. These patterns are the names of architectural elements applied by the user, such as windows, doors, etc. The assignment of an object to a pattern is based on (dis)similarity measures. Hence, there are two additional entities: BIBLIOTEKA WZORCOW and MIARA PODOBIENSTWA. The BIBLIOTEKA WZORCOW entity contains information about pattern types, shape descriptors, object location and allowable parameter values for an object. The MIARA PODOBIENSTWA entity consists of only names of similarity measures which will be used by Matlab to determine the degree of object (dis)similarity to a pattern and to retrieve an image required by user's query from images stored in the DB.

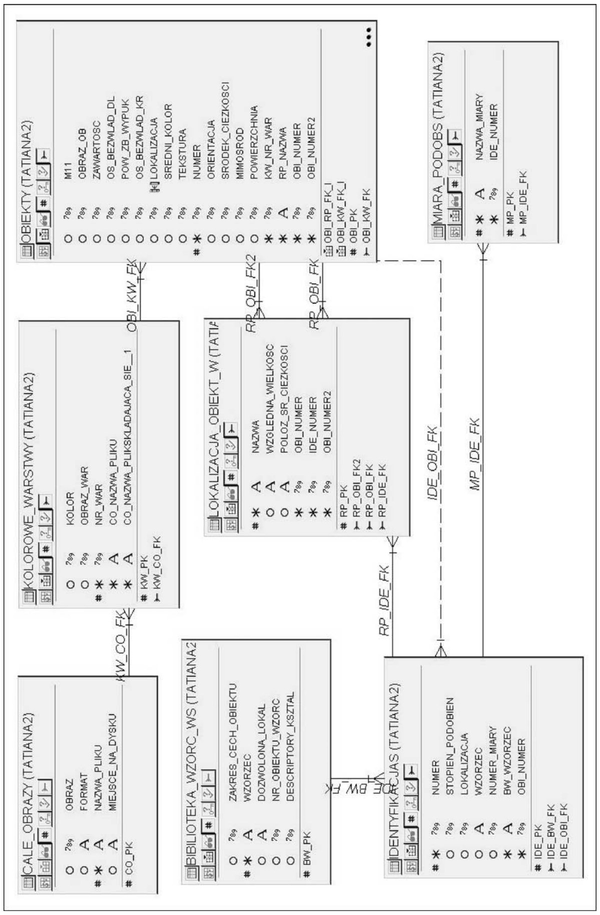

Figure 3. The database server model.

Having transformed the entity-relationships diagram by the DB designer, we obtained the database server model which is presented in fig. 3. In the table models we can also see ID numbers and foreign keys which the Oracle transformer added automatically. The DB discussed here has been generated and tentatively fulfilled with data calculated by Matlab earlier.

3.3. Information flow in the CBIR system

As it has been mentioned in the introduction, the system consists of three main blocks. Each block uses a different standard application which runs on a different operating system (on another physical server). The separation of particular functions among applications, as shown in fig.1, is not self-evident. That is why, the information flow in our CBIR system should be explained. In a graphical way this flow is illustrated in fig. 4.

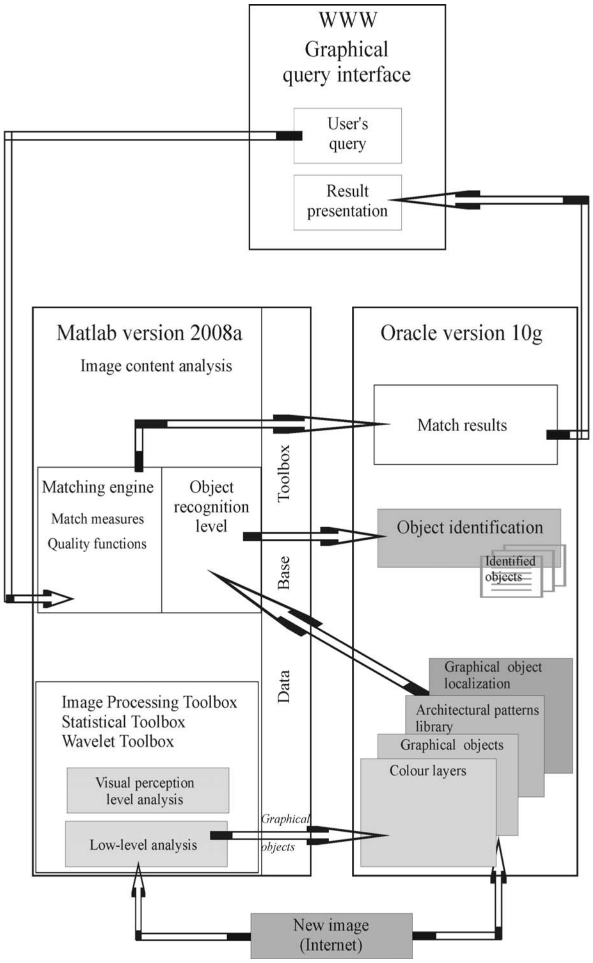

Figure 4. Information flow in the CBIR system.

All the image content analysis is carried out by Matlab, but it is not a sequential process. Firstly, a new image is segmented and parameters of this segmentation are sent to Oracle and stored in the database. This procedure is implemented with the support of the following Matlab Toolboxes: Image Processing, Statistics and Wavelet. DB Toolbox supports the communication between Matlab and Oracle.

Secondly, the stored parameters pertaining to the DB are transferred to the matching engine for object identification. The low-level identification uses shape descriptors: region-based as well as contour-based. Some information such as: solidity, major axes, moments of inertia, etc. is stored in the OBIEKT table. Other shape descriptors must be calculated, especially those using transform spaces. It is significant in so far as if we select pattern parameters in the wrong way, a window, for example, can be recognized as a French window, etc. This is even more difficult because object identification, similarly to image segmentation, is to be a fully automatic process. That is why the information obtained from the images is transferred to the match engine implemented in Matlab.

The next step is finding the similarity of all the above-mentioned parameters describing the object to those stored in the pattern library. It is all designed for the user to receive the most natural and evident answer. People tend to attribute a designation to the objects they can see. If we can see a triangular object, we shall more frequently classify it as a roof than other parts of a house, whereas a dark round object will be usually recognized as a window. The part of the identification process which attributes the objects is based on artificial intelligence algorithms and soft computing. This process is implemented in the object recognition module of the system. The identification is the point where image preprocessing is completed.

The next stage is using the system for answering user’s queries. For this purpose the user's interface on the WWW site cooperates with the matching engine.

Thirdly, the matching engine results are sent to the DB. Then, the DB selects the best parameters received from the matching engine and sends them to the WWW site as an answer for the user.

4. Conclusions and further tasks

The construction of a CBIR system requires combining different systems, which results in various conceptual, as well as purely technical problems. Having built the image processing module for automatic segmentation we had to design the DB to store the generated information about images and their segments. The technical problems which emerged on linking Matlab and Oracle via interfaces were due to the fact that the Matlab server works in the Windows XP environment, whereas the Oracle server works on Unix; the difficulties were further aggravated by the differences between the applications themselves. The ongoing problem is communication of these two servers with the WWW site which is physically located on another server. Therefore, simultaneously with the construction of appropriate applications in each environment, their mutual link must be ensured.

After inputting data on images and their objects in the database, the next vital stage will consist in designing appropriate architectural object patterns. These patterns need to be prepared so that they can be stored in the pattern library. Later, membership functions used in match measures should be adjusted in order to obtain the best matching. Additionally, some logic rules supporting deeper semantic analysis will be implemented. All these actions lead to the creation of a user-friendly system. In the nearest future we hope to apply semantic analysis so profound that the user does not feel the roughness of the system. Consequently, the achievement of all the above-mentioned stages will enable us to complete the CBIR system in its fully-fledged form.

5. References

[1] Barker R., *Entity-Relationship Modelling. Case Method*^{SM} , Addison-Wesley Pub. Co., London, 1995.

[2] Barker R., Longman C., *Function and Process Modelling. Case Method*^{SM}, Addison-Wesley Pub. Co., London, 1993.

[3] Chen Pin-Shan P., "Entity-relationships model – Toward a Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, Nr. 1, ACM, 1976, pp. 9-36.

[4] Date C.J., *An Introduction to Database Systems*. Volume I, Addison-Wesley Pub. Co., IV ed., Amsterdam, 1982.

[5] Deb S., *Multimedia Systems and Content-Based Image Retrieval*, IDEA Group Publishing, Melbourne, 2004.

[6] Fauqueur J., Boujemaa N., "Mental image search by Boolean composition of region categories", *Multimedia Tools and Applications*, Vol. 31, No. 1, Springer Netherlands, October, 2006, pp. 95-117.

[7] Flickner M., Sawhney H., et al., "Query by Image and Video Content: The QBIC System", *IEEE Computer*, Vol. 28, No. 9, 1995, pp. 23-32.

[8] Hsu W., Chua T. S., Pung H. K., "Approximation Content-based Object-Level Image Retrieval", *Multimedia Tools and Applications,* Vol. 12, Springer Netherlands, 2000, pp. 59-79.

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

[10] Mokhtarian F., Abbasi S., Kittler J., "Robust and Efficient Shape Indexing through Curvature Scale Space," *Proc. British Machine Vision Conference,* British Machine Vision Asso. 1996, pp. 53-62.

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

[12] Ogle V., Stonebraker M., "CHABOT: Retrieval from a Relational Database of Images", *IEEE Computer* **28**, No. 9, 1995, pp. 40-48.

[13] Rubner Y., Tomasi C., Guibas L., "The Earth Mover's distance as a Metric for Image Retrieval, International", *Journal of Computer Vision*, Vol. 40, No. 2, Springer Netherlands, 2000, pp. 99-121.

[14] Tan K-L., Ooi B. Ch., Yee Ch. Y., "An Evaluation of Color-Spatial Retrieval Techniques for Large Image Databases", *Multimedia Tools and Applications*, Vol. 14, Springer Netherlands, 2001, pp. 55-78.

[15] Urban J., Jose J. M., van Rijsbergen C. J., "An Adaptive Technique for Content-Based Image Retrieval", *Multimedia Tools Applications*, Springer, Vol. 31, 2006, pp. 1-28.