Journal of Digital Imaging

POSTGRESQL-IE: An Image-handling Extension for PostgreSQL

Denise Guliato,[1] Ernani V. de Melo,[1] Rangaraj M. Rangayyan,[2] and Robson C. Soares[1]

The last decade witnessed a growing interest in research on content-based image retrieval (CBIR) and related areas. Several systems for managing and retrieving images have been proposed, each one tailored to a specific application. Functionalities commonly available in CBIR systems include: storage and management of complex data, development of feature extractors to support similarity queries, development of index structures to speed up image retrieval, and design and implementation of an intuitive graphical user interface tailored to each application. To facilitate the development of new CBIR systems, we propose an image-handling extension to the relational database management system (RDBMS) PostgreSQL. This extension, called PostgreSQL-IE, is independent of the application and provides the advantage of being open source and portable. The proposed system extends the functionalities of the structured query language SQL with new functions that are able to create new feature extraction procedures, new feature vectors as combinations of previously defined features, and new access methods, as well as to compose similarity queries. PostgreSQL-IE makes available a new image data type, which permits the association of various images with a given unique image attribute. This resource makes it possible to combine visual features of different images in the same feature vector. To validate the concepts and resources available in the proposed extended RDBMS, we propose a CBIR system applied to the analysis of mammograms using PostgreSQL-IE.

KEY WORDS: Database management systems, digital mammography, digital image management, image database, image retrieval, information storage and retrieval, information system

INTRODUCTION

Content-based image retrieval (CBIR) has emerged as an important area in computer vision and multimedia computing. A large number of new techniques to derive image features[1–7] and several CBIR systems[8–14] have been developed and used in a vast range of applications. An important subject related to the development of CBIR systems is searching a database using similarity queries with high performance. Queries of this kind search the database for objects that are similar to one given as the reference, according to a similarity measure given by a distance function.[15] The most well-known strategies to implement similarity queries are: K-nearest neighbor (KNN), range search, and similarity join.[16,17] With the aim of increasing the efficiency of similarity searches, several index structures have been proposed.[18–22] However, few works have been proposed to extend the structured query language SQL to provide support for similarity queries and are restricted to predefined access methods and a limited number of feature extraction procedures.[15,25,26]

A CBIR system developer is required to be involved with several complex tasks such as: understanding the domain of the application, organizing the data of the application and storing

[1]From the Faculdade de Computação, Universidade Federal de Uberlândia, Av. João Naves de Ávila, 2121, 38.400-902, Minas Gerais, Brazil.

[2]From the Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive N. W., Calgary, Alberta, Canada T2N 1N4.

Correspondence to: Denise Guliato, Faculdade de Computação, Universidade Federal de Uberlândia, Av. João Naves de Ávila, 2121, 38.400-902, Minas Gerais, Brazil; e-mail: guliato@ufu.br

Online publication 23 January 2008

doi: 10.1007/s10278-007-9097-5

them using a database management system (DBMS) or a repository as the folders of an operating system, designing a layout and developing an intuitive graphical user interface (GUI), developing feature extraction algorithms, access methods, and similarity operators, and finally, defining and implementing similarity query strategies. However, many of these tasks are common to the majority of CBIR systems and could be available in an extendable and configurable application-independent framework.

With the aim of providing facilities for the development of CBIR systems, we propose an extension to the architecture of PostgreSQL,[27] termed *PostgreSQL with Image-handling Extension* (PostgreSQL-IE).[13] The proposed system is independent of the application, makes use of the powerful tools available with PostgreSQL, while including more flexibility with respect to the creation of new feature descriptors and the definition of new feature vectors, without extra cost. At the current stage of its development, PostgreSQL-IE makes available two similarity search procedures: KNN and range search. Twelve feature extraction procedures are available, with one of them related to color: histogram, and the remainder related to shape factors: Fourier descriptors (FDs), area, perimeter, compactness, spiculation index (SI), concavity index,[6] index of spiculation based on the turning angle function, fractal dimension, index of the presence of concavity, index of the presence of convexity, and index of convexity.[3,4,28]

To illustrate and validate the power of the resources available in PostgreSQL-IE, we have developed a research system that supports content-based retrieval of mammograms. The research system allows the user to retrieve information from a mammographic database by combining conventional and visual data, using a friendly graphical Web interface.

The remainder of this paper is structured as follows: "Review of Related Works on CBIR" presents a review of related works. "PostgreSQL with Image-handling Extension" presents the fundamental concepts related to PostgreSQL-IE, including the data structure of the new image data type and the extension of the database catalog.[13] "A Breast Cancer Database Application" provides a description of the dataset used in the examples. "The Extended SQL: SQL-IE" gives an overview of the extended SQL, including a description of the main commands. "SISPRIM—A System Design for CBIR Applied to Mammograms" describes SISPRIM—Sistema de Pesquisa para Recuperação de Imagens Mamográficas por Conteúdo,[29] a research system designed using PostgreSQL-IE that supports CBIR applied to the analysis of mammograms. Finally, "Conclusions" gives the conclusions of this paper and suggestions for future works.

REVIEW OF RELATED WORKS ON CBIR

To support the development of CBIR systems, commercial extensions are available to relational DBMS (RDBMS) that are able to store images as a special data type and to include strategies for searching by similarity, restricted to KNN and range search.[16,17] Examples of such extended RDBMS (XRDBMS) are DB2 with Image Extender,[30] Informix with Excalibur Image DataBlade,[31] Informix with Image Foundation DataBlade (IFDB),[32] and Oracle with interMedia.[33] However, these XRDBMS have a private architecture, are expensive, and do not include facilities for extension or modification. In general, these systems possess similar functionalities but support different sets of image data type and include different feature extraction procedures, access methods (if available), and image manipulation commands.

With the aim of standardizing the resources and the image data type of XRDBMS, the International Organization for Standardization proposed *SQL/MM Part 5: Still Image*.[26,34] The proposed standardization provides a description for an image data type, termed *SI_StillImage*, that should encapsulate: the binary image, its dimension, format, width, and height, a set of five image manipulation functions that includes scaling, rotation, cropping, color reduction, and sharpening and blurring, and a set of four feature extraction procedures, of which one is based on texture and the remainder are based on color (global color average, local color average, and color histogram). Each feature, derived from the execution of a feature extraction procedure, can be associated with a feature vector data type. A special feature vector data type called *SI_FeatureList* makes it possible to combine different features, using different weights, in the same feature vector. However, *SQL/MM Part 5: Still Image* does not cover aspects related to the query image by similarity search and does not provide flexibilities to include new feature extraction procedures. Notwithstanding the efforts, none of the available commercial XRDBMS is completely aligned with the standardization proposed by *SQL/MM Part 5: Still Image*, except Oracle interMedia—version 10, which includes the *SQL/MM Part 5: Still Image* module besides its own previously defined functionalities. Oracle interMedia[33] supports two image data types, *StillImage* and *ORDImage*, and is limited to five feature extraction procedures (global color, local color, texture, structure, and facial). The image data type available in DB2 with Image Extender[30] is defined as a user-defined type (UDT), termed *DB2Image*, and is limited to four feature extraction procedures (color average, color histogram, positional color, and texture). Informix Excalibur Image DataBlade Module[31] supports a new image data type, defined as a UDT, called *IfdImageDesc*, and makes available six feature extraction procedures (color content, color structure, texture, shape, brightness structure, and aspect ratio). IFDB[32] defines another image data type, termed *IfxImage*; the image is stored as a binary large object (BLOB) type attribute, and no feature extraction function is available in this system.

The system proposed by Baroni et al.[15] does not allow execution of the feature extraction procedures as part of the system and does not include a special data type to store image data. The feature vector is comprised of features previously obtained by the user and is stored in the database as a special data type termed PARTICULATE. The system makes it possible to create new metrics associated with the PARTICULATE data type. The similarity query is carried out by taking into account the metric associated with the feature vector defined by the user and the similarity operators KNN, Range, and Join. Although this system extends the SQL to pose similarity queries, the approach is different from that proposed in the present work. Because PostgreSQL-IE is closer to the commercial RDBMS, we restrict further discussions and comparisons to the related context. The features supported for CBIR in all available commercial XRDBMS are mostly restricted to color and texture. The Informix Excalibur Image DataBlade and Oracle RDBMS go beyond of the standardized features and implement a shape-based feature. However, additional specialized features would be desirable for specific applications.[35]

All of the commercial systems described above are able to store only one image in a given image data type attribute (for short, image attribute) and make it possible to associate only one feature vector with each image attribute. The feature vector may be composed as a weighted combination of all of the features available in the system: No other user configuration is permitted. The systems do not allow the inclusion of new functionalities, new feature extraction procedures, or new access methods.

POSTGRESQL WITH IMAGE-HANDLING EXTENSION

To develop CBIR systems, it is desirable that the XRDBMS provides flexibility to store images, to define new feature extraction procedures, and to combine the features into vectors in new manners that are not predefined. To address these requirements, we propose PostgreSQL-IE, an extension designed to support similarity queries in a straightforward manner. To manage the new functionalities, PostgreSQL-IE extends the PostgreSQL database catalog and includes the definition of a new image data type, as described in the following sections.

The Extended Database Catalog

The extended database catalog for PostgreSQL-IE is a database, called *extended_template1*, created using *template1* as the template, with a set of 12 internal functions written in the C language, a set of 16 functions written in *plpgsql*,[36] which extends the SQL commands, whose details are described in "The Extended SQL: SQL-IE," and a set of six tables that manages all of the information related to image handling, whose scheme is shown in Figure 1. The tables *pge_extractor*, *pge_vector*, and *pge_accessmethod* keep track of all new feature extraction procedures, new feature vectors, and new access methods added to the system by authorized users, respectively. The table *pge_vector_extractor* relates each feature vector to a respective set of features that compose the vector. Because of this aspect, PostgreSQL-IE makes it possible to combine one or more features in the same feature vector, according to the needs of the application. To

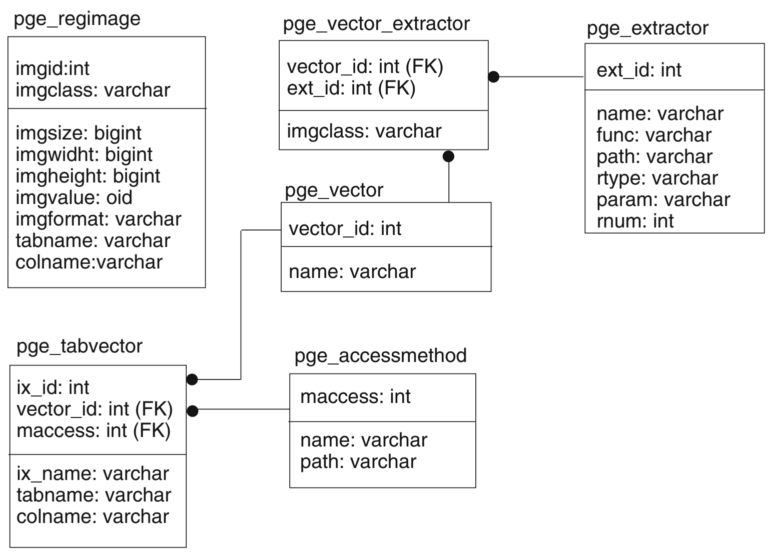

Fig 1. Scheme of the extended database catalog for PostgreSQL-IE.

speed up the answer to a query, an access method can be associated with an available feature vector: The table *pge_tabvector* manages this relationship. Finally, the table *pge_regimage* maintains information about all of the images stored in a given database, which is accessed directly from the application table, as described in the following sections. New databases created using *extended_template1* as the template inherit all the resources to manage and handle CBIR operations.

A New Image Data Type

PostgreSQL-IE implements a new image data type as a UDT, termed *PGImage*, according to the *plpgsql* command:

This is possible by attributing different classes to different images of the same attribute. This novel approach is helpful in developing applications where each image attribute is composed of a set of images such as computed tomography (CT), magnetic resonance imaging (MRI), mammography, or architectural plans.

The *PGImage* data type encapsulates an identifier of the instance of a given image set (*imgid*) and the classes associated with each image of this image set (*imgclass*). The identifier *imgid* is a key of the *pge_regimage* table, which stores information about each image stored in the database, such as image dimension, width, height, format, the referred table name, the referred column, and the binary image stored as a BLOB.

```
CREATE TYPE PGImage AS (
       imgid INTEGER,
       imgclass VARCHAR,
       classes VARCHAR   /*internal control for the CHECK clause in the CREATE TABLE command */
).
```

This new image data type includes the capability to store more than one image in the same image attribute, different from all other image data types reviewed in "Review of Related Works on CBIR."

The new data type *PGImage* allows the modeling of a database application in a simpler way, in comparison to other XRDBMS. For example, consider an application that involves a radiological electronic report that includes a short clinical history of each patient and her set of related exams obtained at different instants of time. Each exam set can be composed of X-ray, CT, and MRI images (the number of slices in CT and MRI may vary between patients). Figure 2 illustrates the database scheme using DB2 and PostgreSQL-IE XRDBMS.

Because *PGImage* allows the storage of more than one image in the same attribute, the application model using PostgreSQL-IE is simpler and can be modeled with fewer tables, as compared to DB2. By modeling the database using PostgreSQL-IE, the visual information related to all exams carried out for a patient at a given instant of time can be combined in the same similarity query. This is not possible using any commercial XRDBMS available at present.

A BREAST CANCER DATABASE APPLICATION

To illustrate the use of resources available in PostgreSQL-IE and the commands for the extended SQL, to be presented in the following sections, we will use a breast cancer database application.

The mammographic database, called *mammo_database*, is modeled by two tables: *patient* that contains information about the patients and *patient_case* that contains information about the mammographic exams carried out on the patients at a given instant of time. Each mammogram is associated with the contour of the lesion, drawn by an expert radiologist. The command

```
CREATE TABLE patient_case(
   c_id INTEGER NOT NULL,
   c_patient INTEGER NOT NULL,
   c_examdate DATE NOT NULL,
   c_tumor CHAR NOT NULL
   c_mammo PGImage,
   CHECK ((c_mammo).classes in ('mammography', 'lesion_contour'))
);
```

creates the *patient case* table. The attribute *c_mammo* is a *PGImage* data type and stores two different images: a mammographic image whose class is named *mammography* and the lesion contour whose name is *lesion_contour*, drawn by an expert radiologist. Figure 3 shows an instance for the *patient_case table* and its respective *pge_regimage table* with the aim of illustrating the management for the PGImage data type.

THE EXTENDED SQL: SQL-IE

SQL-IE is composed of a set of 16 new functions that includes commands to create new feature extraction procedures, new feature vectors as combinations of previously defined features, and new access methods. The details about the syntax of the SLQ-IE commands using the functions to be described in the following sections can be found in Appendix A.

Data Definition Functions

In this section, we describe three of the six functions available in SQL-IE for data manipulation. The others three functions termed *Delete_Extractor*, *Delete_Feature_Vector*, and *Delete_Accessmethod* are used for maintenance of the database catalog.

- *Create_Extractor* —This function makes it possible to include a new feature in PostgreSQLIE, by inserting specific information about the feature extraction procedure in the *pge_extractor* table and by creating internal control functions, which are able to access the function in the appropriate library.

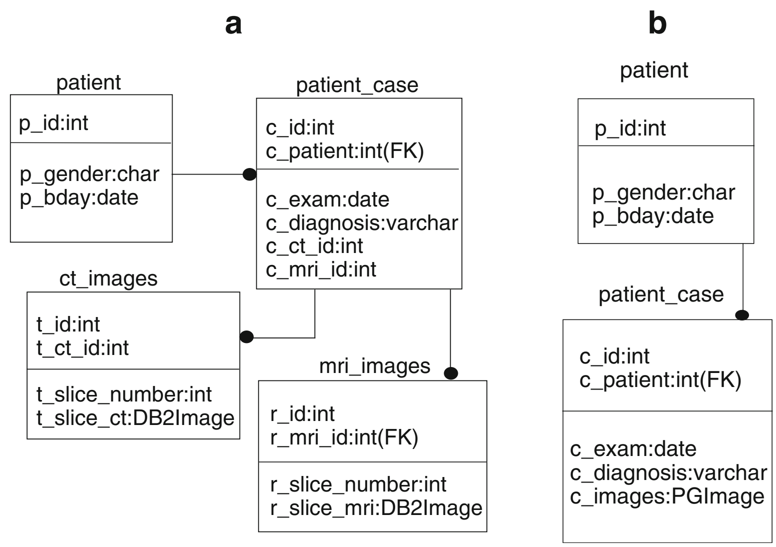

Fig 2. Two possible database schemes: a for DB2 with Image Extender, b for PostgreSQL with Image-handling Extension.

To include a new feature, it is required to convert the corresponding function's code to the library format *dll* (dynamic linked library) or *so* (shared object), depending on the operating system, and to store it in the *lib* directory of PostgreSQL. At present, PostgreSQL-IE makes available four different libraries in *dll* and *so*: *ExtShape* with six feature extraction procedures proposed by Rangayyan et al.[37] (SI, FDs, fractional

a

patient_case table				
c_id	c_patient	c_examdate	c_tumor	c_mammo
1	1	2000-04-05	B	(1, "mammography, lesion_contour")
2	2	2004-08-01	B	(2, "mammography, lesion_contour")
3	3	2004-10-10	M	(3, "mammography, lesion_contour")

b

pge_regimage table								
imgid	imagclass	imgsize	imgwidth	imgheight	imgformat	imgvalue	tabname	colname
1	lesion_contour	4194304	2048	2048	bnd	140338	patient_case	c_mammo
1	mammography	4194304	2048	2048	bmp	140337	patient_case	c_mammo
2	lesion_contour	1048576	1024	1024	bnd	140346	patient_case	c_mammo
2	mammography	1048576	1024	1024	bmp	140345	patient_case	c_mammo
3	lesion_contour	1048576	1024	1024	bnd	140352	patient_case	c_mammo
3	mammography	1048576	1024	1024	bmp	140351	patient_case	c_mammo

Fig. 3. a An instance of the patient case relation scheme. b The corresponding *pge regimage table.*

concavity, perimeter, area, and compactness), *ExtColor* with a feature extraction procedure based on the global histogram, *ExtShape-PolyModelTAF* with five feature extraction procedures,[2,28] and *ExtShapeTAFModeling* with five feature extraction procedures.[3,4] The last two libraries include the same set of feature extraction procedures, derived from the turning angle function of a given contour: index of spiculation, fractal dimension, presence of concavity, presence of convexity, and convexity index; they differ from each other with respect to the contour approximation approach used to remove noise and artifacts from the original contour. The commands given below illustrate the use of the *Create_Extractor* function to include new feature extraction procedures in PostgreSQL-IE. In the example, we use the functions *getFourierDescriptors*, *getSpiculationIndex*, and *getHistogram* available in the *ExtShape* and *ExtColor* libraries.

```
SELECT Create_Extractor ('Fourier', 'getFourierDescriptor', 'float', 'ExtShape');

SELECT Create_Extractor ('SI', 'getSpiculationIndex', 'float', 'ExtShape');

SELECT Create_Extractor (' Histogram', 'getHistogram', 'float', 256, 'ExtColor');
```

- *Define_Feature_Vector*—This function defines a new feature vector as a combination of features derived from the feature extraction procedures previously created. The tables *pge_vector* and *pge_vector_extractor* of the extended database catalog keep information about all feature vectors that have already been defined and about their elements, respectively. The following commands illustrate the definition of three different feature descriptors: *Shape_SI* that is composed of SI, *Shape_SI_FD* that is composed of SI and FDs, and finally, *Hist_SI* that is composed of SI and histogram feature extractors. Note that the last feature vector mentioned above combines features of images from different classes (lesion_contour and mammography in our example).

```
SELECT Define_Feature_Vector ( 'Shape_SI', 'SI:lesion_contour');

SELECT Define_Feature_Vector ('Shape_SI_FD', 'SI:lesion_contour; Fourier:lesion_contour');

SELECT Define_Feature_Vector ('Hist_SI', 'Histogram:mammography; SI:lesion_contour');
```

- *Create_AccessMethod*—This function allows to create a new access method. The similarity operators are functions defined within the access method, with a specific structure, as shown in Appendix A1. The table *pge_accessmethod* of the extended database catalog is updated to support the new access method. At present, PostgreSQL-IE does not include any multidimensional access method. The queries are carried out by scanning all the tuples in a specified table. This procedure is referred to in this paper as the *Standard-Access-Method* (Scan) and includes KNN search and range search. The similarity operators allow to associate weights with each feature of a feature vector being used in the query. To carry out a KNN or range search, the difference between the normalized weighted sum of the feature vector associated with the reference image and all the images in the referenced database table are computed. The images are ranked in increasing order of the resulting distances, and the similarity operations are appropriately processed.

Data Manipulation Functions

In this section, we present the main functions to manipulate data in the XRDBMS. Other data manipulation functions include *Width_IE*, *Show_Image*, *UnSet_Feature_Vector*, *Show_Extractor*, *Update_Image_Attr*, *Replace_Image_Attr*, *Height_IE*, and *Value_Extractor*.

- *Insert_Image*—This function is used as part of the conventional SQL Insert command. The

Insert_Image function allows the user to insert images into an image attribute. If there are one or more feature vectors already associated with the image attribute of the table being updated, the database system launches the execution of all of the feature extractors related to the feature vectors. Additional information such as the width, height, and format of the images are also saved. The command

```
INSERT INTO patient_case( c_id, c_patient, c_examdate, c_mammo)
VALUES (1, 1, '04-05-2000',
        Insert_Image ('patient_case', 'c_mammo', 'mammography,D:/mammo/circ_db_290.bmp;
                      lesion_contour,D:/mammo/circ_db_290.bnd'));
```

illustrates the insertion of one tuple into the table *patient_case*. The *pge_regimage* table of the extended database catalog is updated to reflect this insertion (see Fig. 3a, an instance for the patient case table).

- *Set_Feature_Vector*—This function associates an image attribute of a given database table with a previously defined feature vector and creates an index structure if an access method is explicitly specified in the command (if not, a scan of all the tuples in the database is executed). More than one feature vector can be associated with the same image attribute of a table, as shown below.

```
SELECT Set_Feature_Vector ('Index_SI', 'patient_case', 'c_mammo', 'Shape_SI');
SELECT Set_Feature_Vector( 'Index_SI_FD', 'patient_case', 'c_mammo', 'Shape_SI_FD');
```

Once the feature vectors have been associated with a given attribute with instances already inserted, all of the feature extraction functions associated with the feature vectors are executed for these instances. To avoid allocating extra storage space and to avoid repeated calls to execute an extractor function for the same attribute, PostgreSQL-IE reuses the results of the feature extractors that have already been executed before. An internal table named with the result of the concatenation of the referred database table name and the attribute nameis dynamically created, with only one column for each feature extractor. Note that although the *SI* and *Fourier* feature extractors comprise two different feature vectors (*Index SI* and *Index SI FD*), they appear only once in the internal table.

SQL-IE Commands to Search an Image Database by Similarity

The proposed extended SQL (SQL-IE) supports similarity query in a simple way. The similarity operators are implemented as functions enclosed within the access method and are used as subqueries of the WHERE clause of a conventional SQL command. The answer to the query is ranked in increasing order, according to the similarity of the objects in the database and an object given by the reference. The follow examples illustrate queries using PostgreSQL-IE.

Example 1:

```
SELECT  p1.p_id, p1.p_bday,Score_IE ('sc_1', c1.c_mammo) AS score1
FROM patient p1, patient_case c1, patient_case c2
WHERE p1.p_id = c1.c_id
      AND (c2.c_mammo).imgid = 13
      AND p1.p_bday > 1948-01-01
      AND (c1.c_mammo).imgid in (SELECT * FROM KNN ('sc_1', 5, 'Index_SI', c2.c_mammo))
ORDER BY score1;
```

Example 2:

```
SELECT  p1.p_id, p1.p_bday,Score_IE ('sc_1', c1.c_mammo) AS score1,
                           Score_IE ('sc_2', c1.c_mammo) AS score2
FROM patient p1, patient_case c1, patient_case c2
WHERE p1.p_id =c1.c_id
      AND (c2.c_mammo).imgid = 13
      AND ((c1.c_mammo).imgid in (SELECT * FROM KNN( 'sc_1', 5, 'Index_SI', c2.c_mammo))
         AND
          (c1.c_mammo).imgid in (SELECT * FROM RANGE ('sc_2', 0.1, 'Index_SI_FD:SI=0.9;
                                                      Fourier=0.7', c2.c_mammo)))
ORDER BY score1;
```

Because SQL-IE allows the searching of the database by combining similarity operators connected by logical operators, as shown in example 2 above, the explicit statement of a score name in the *Score IE* function, in the SELECT clause, makes it possible to rank the answer to the query according to the score given by the specified similarity operator.

SISPRIM—A SYSTEM DESIGN FOR CBIR APPLIED TO MAMMOGRAMS

This section presents SISPRIM,[13,29] a research system that supports CBIR, applied to the analysis of mammograms for computer-aided diagnosis of breast cancer. The system is composed of a *mammographic database* modeled using PostgreSQL-IE, a *research engine* developed in PHP, and a *Web GUI*.

The Mammographic Database

The *mammographic database* was modeled using PostgreSQL-IE and stores, for each patient, her historical clinical information, some relevant aspects of her life style, and the mammographic exams performed at different instants of time.[38] For each mammographic exam, the following are stored: the four standard views (two views of each breast: cranio-caudal and medio-lateral oblique or

MLO), complementary exams, information about the presence of abnormalities, such as architectural distortion, asymmetric density, palpable lump, and calcifications, and the diagnosis, as applicable, according to the Breast Imaging—Reporting and Data System (BI-RADS) classification system.[39] The mammographic database associates, with each mammographic exam, the report given by the radiologist and links each mammogram of each exam with the contour of the breast, the boundary of the pectoral muscle (MLO views only), the contours of masses (if present), the regions of clusters of calcifications (if present), the number of calcifications (if present), and the locations and details of any other features of interest. The contours of masses and regions of clusters of calcifications may be drawn and entered into the system by an authorized expert radiologist. In the current stage of development, the mammographic database of SISPRIM is composed of contours and regions of interest (ROIs) of breast masses obtained in two preceding studies; one set of images was derived from mammograms of 20 cases obtained from Screen Test: the Alberta Program for the Early Detection of Breast Cancer.[14,40,41] The mammograms were digitized using the Lumiscan 85 scanner at a resolution of 50μm with 12b/pixel. The set includes 57 ROIs, of which 37 are related to benign masses and 20 are related to malignant tumors. Another set of images was obtained from the Mammographic Image Analysis Society (MIAS, UK) database[42] and the teaching library of the Foothills Hospital (Calgary).[43] The MIAS images were digitized at a resolution of 50μm; the Foothills Hospital images were digitized at a resolution of 62μm. This set includes smooth, lobulated, and spiculated contours in both the benign (28) and malignant (26) categories. The contour of each lesion was manually drawn by an expert radiologist specialized in mammography. The combined dataset has 111 contours, including both typical and atypical shapes of benign masses (65) and malignant tumors (46). The development of an additional mammographic database is in progress: The new database is modeled to contain cases from Digital Database for Screening Mammography[44] with images in a well-supported format. The radiological findings of each case stored in the database are being extended by expert radiologists at the Clinical Hospital of Federal University of Uberlândia, Brazil.

The Research Engine

The *research engine* is the heart of the research system: It manages the actions of the user and accesses the mammographic database as required to answer queries. The proposed research system can answer queries such as *return five images that are similar to the given reference image and the patient takes antidepressive medication* or *return the identification of the patient and the diagnosis associated with each one of ten images similar to the reference image, and the density of parenchyma is heterogeneous, and the age at menarche is 12 years*.

To handle queries as above, the research engine requires that each mammogram in the database be associated with a set of features combined into one or more feature vectors and that the data manipulation language possesses resources to combine visual and conventional information in the condition clause of the query. As the mammographic database used by SISPRIM was modeled using PostgreSQL-IE, all requirements to support CBIR are available. To facilitate easy access to the information stored in the mammographic database and to access the resources available in the XRDBMS, the research engine manages a Web GUI that incorporates facilities to support two kinds of users: administrators and researchers. The interface for an administrator guides the user in the creation of a new feature extractor and/or a new feature vector and in the association of a given feature vector with an image attribute, including an access method (optional), as shown in Figure 4. The interface for a researcher, shown in Figure 5, guides the user through the resources previously configured by an administrator. A researcher can interactively configure a query, combining conventional and visual data, as desired. Figure 6 shows the results of the query as posed in Figure 5. The results of the queries indicate the capabilities of the proposed system in CBIR: The specific query illustrated shows CBIR based upon the shape of masses with a condition on the age of the patient. Further work is in progress to include into SISPRIM strategies to improve the retrieval performance using relevance feedback.[45–47]

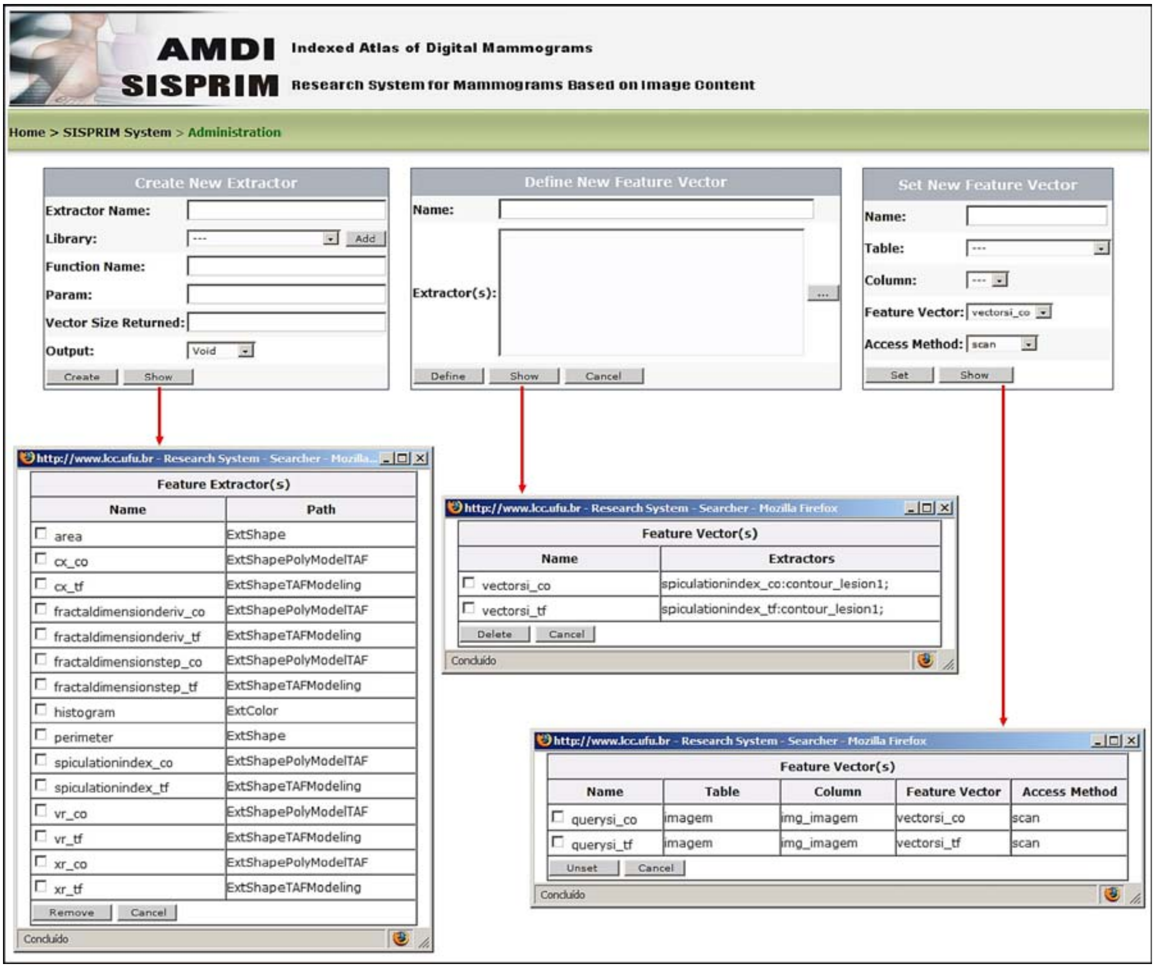

Fig 4. The administrator–user interface to configure the research system.

CONCLUSIONS

We have presented two important contributions. The first one is PostgreSQL with Image-handling Extension, an XRDBMS that supports the development of CBIR systems in a flexible manner. The system is open source, portable, extendable, easy to be installed, and available via the Web for Windows and Linux operating systems. At present, the system includes 12 different feature extraction procedures organized in four libraries. Detailed discussions on the accuracy of the features used in the present paper can be found in our previous works.[2–4,6,28] PostgreSQL-IE supports a new image data type attribute termed PGImage, which permits the user to model a relational scheme by storing various images of different classes in the same attribute. Because of this image attribute, the proposed system makes it possible to design the conceptual model for a given database application with fewer relation schemes than those required in currently available commercial RDBMS. The new image data type also makes it possible to combine features derived from different images, stored in the same image attribute, in the same feature vector. To date, PostgreSQL-IE makes available two conventional similarity operators (KNN and range); further studies are being conducted to extend PostgreSQL-IE with two new similarity search procedures based on fuzzy sets to take into

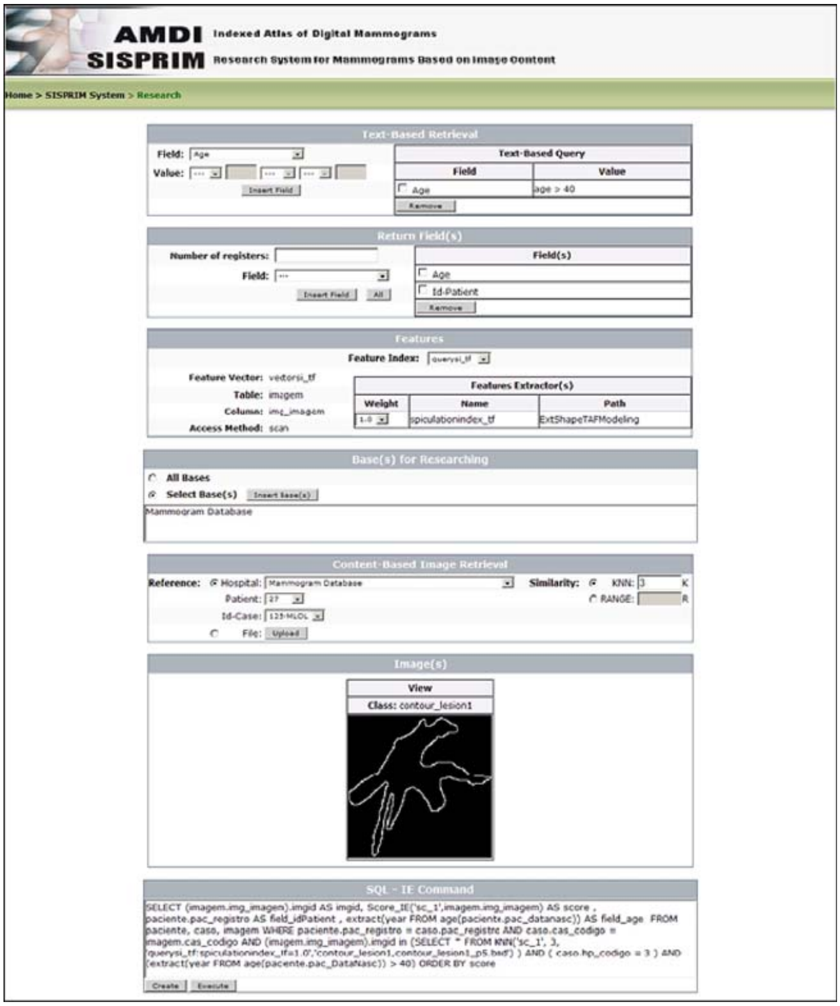

Fig 5. The researcher graphical Web interface to define a query.

account the uncertainty present in medical applications of CBIR. New feature extraction procedures and new access methods can be easily included in the system.

The second contribution is SISPRIM, a research system designed by using PostgreSQL-IE. The research system accesses a mammographic database that includes complete information about the patient (clinical history and life style), her mammographic exams, complementary exams, and the report of each exam associated with the radiological findings of each mammogram, according to the BI-RADS classification system. The information about the clinical history and lifestyle includes details such as the use of alcohol, tobacco, antidepressive medication, and hormone replacement medication, age at menarche, age at first pregnancy, and menopausal status. The radiological findings include asymmetric density, architectural distortion, parenchymal type, and descriptions of the characteristics of masses and calcifications (as applicable). With information as above, it is possible to compute statistical measures and correlation coefficients to investigate and derive relationships between the incidence of breast cancer and the lifestyle of the patient. The system also permits temporal analysis of the evolution of the breast, which is useful in un-

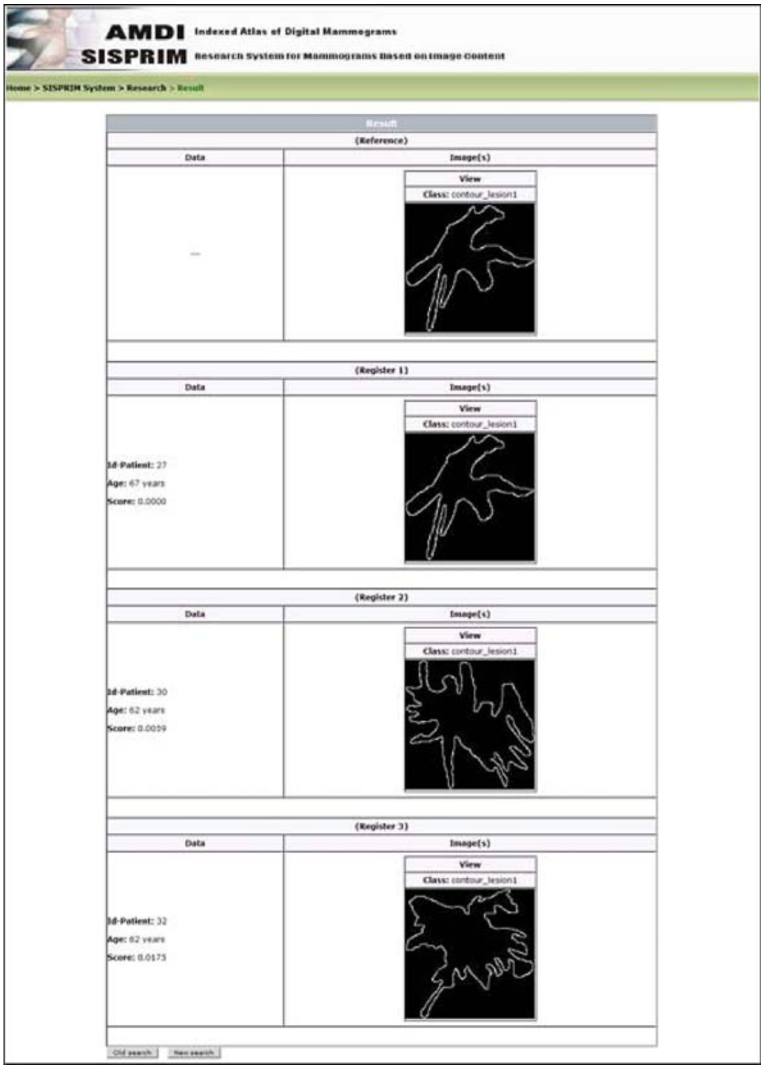

Fig 6. The results of the search as defined in Figure 5.

derstanding the natural changes of the breast and interval cancer. SISPRIM makes available a graphical Web interface that helps the user to configure and execute queries. To illustrate the power and facilities of SISPRIM and PostgreSQL-IE, we have modeled a mammographic database with 111 images and carried out queries using the KNN operator. The development of an additional mammographic database is in progress.

The relevance of the results obtained by the proposed system depends on the feature extractors and the feature vectors used. To improve the performance of the CBIR system, strategies of relevance feedback that take into account the perception of the user are being developed.

SISPRIM is part of An Indexed Atlas of Digital Mammograms[38,48,49] and can be accessed at www.lcc.ufu.br/amdi.

The script to install PostgreSQL-IE is available at www.lcc.ufu.br/pdi/downloads.

ACKNOWLEDGMENT

This work was supported by the Conselho Nacional Desenvolvimento Científico e Tecnológico, Brazil, Universidade Federal de Uberlândia, Brazil, and Research Services, University of Calgary, Canada.

APPENDIX A

A1—Syntax for SQL-IE Data Definition Functions

In this Appendix, we present the detailed syntax of the definition functions used in "Data Definition Functions."

- The *Create_Extractor* function

```
Create_Extractor (Extractor-Name, Extractor-Function-Name,
                  Return-Data-Type, [Number-of-Output-Parameter],
                  Library-Path[,<Input-Option>])
```

- The *Define_Feature_Vector* function

```
Define_Feature_Vector (Feature-Vector-Name, <List-of-Extractor>)
where:

 <List-of-Extractor>::= Extractor-Name:<Image-Class-List> |
                        Extractor-Name; <List-of-Extractor>
<Image-Class-List>::= Image-Class, <Image-Class-List>
```

- The *Create_AccessMethod* function—The access method must be developed in the C programming language, converted to library format (*dll* or *so*), and has to include the following functions:

Create_Index_Structure (), *Drop_Index_Structure ()*, *Insert_Element_Index ()*, *KNN ()*, *RANGE ()*, *Delete_Element_Index ()*, and *Update_Element_Index ()*.

The input parameters for the similarity operators are: the score name, the value of the neighborhood for KNN and the value of ratio for the RANGE operators, the name of the index structure, the image class, and the file path of the reference image for the similarity query.

```
Create_AccessMethod (Access-Method, Access-Method-Path)
```

A2—Syntax for SQL-IE Manipulation Commands

In this Appendix, we present the detailed syntax of the manipulation functions used in "Data Manipulation Functions."

- The *Insert_Image* function

```
Insert_Image (Table-Name, Image-Attribute-Name, <Image-List>)

where:


<Image-List>::= Image-Class, File-Image-Path |

                Image-Class, Image-File-Path; < Image-List >
```

- The *Set_Feature_Vector* function

```
Set_Feature_Vector (Index-Structure-Name, Table-Name,

                    Image-Attribute-Name, Feature-Vector-Name

                    [, Access-Method-Name)]
```

REFERENCES

1. Datta R, Li J, Wang JZ: Content-based image retrieval—approaches and trends of the new age. In: Proceedings of the ACM International Workshop on Multimedia Information Retrieval, ACM Multimedia, Singapore, November, 2005, pp 253–261
2. Guliato D, Rangayyan RM, Carvalho JD, Santiago SA: Spiculation-preserving polygonal modeling of contours of breast tumors. In: Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New York City, NY, September, 2006, pp 2791–2794
3. Rangayyan RM, Guliato D, Carvalho JD, Santiago SA: Feature extraction from the turning angle function for the classification of breast tumors. In Proceedings of the International Special Topics Conference on Information Technology in Biomedicine—IEEE ITAB2006, Ioannina, Greece, October, 2006 (6 pages on CDROM)
4. Carvalho JD, Rangayyan RM, Guliato D, Santiago SA: Polygonal modeling of contours using the turning angle function. In 20th IEEE Canadian Conference on Electrical and Computer Engineering, Vancouver, BC, April, 2007, pp 1090–1267
5. Chen Y, Wang JZ: A region-based fuzzy feature matching approach to content-based image retrieval. IEEE Trans Pattern Anal Mach Intell 24(9):1252–1267, 2002
6. Mudigonda NR, Rangayyan RM, Desautels JEL: Detection of breast masses in mammograms by density slicing and texture flow-field analysis. IEEE Trans Med Imag 20(12):1215–1227, 2001
7. Rangayyan RM, Nguyen TM: Fractal analysis of contours of breast masses in mammograms. J Digit Imaging 20(3):223–237, 2007 (September)
8. Veltkamp RC, Tanase M: Content-based Image and Video Retrieval, Norwell, MA: Kluwer, 2002
9. Csillaghy A, Hinterberger H, Benz AO: Content-based image retrieval in astronomy. Inf Retr 3(3):229–241, 2000
10. Painter TH, Dozier J, Roberts DA, Davis RE, Green RO: Retrieval of subpixel snowcovered area and grain size from imaging spectrometer data. Remote Sens Environ 85(1):64–77, 2003
11. Schroder M, Rehrauer H, Seidel K, Datcu M: Interactive learning and probabilistic retrieval in remote sensing images archives. IEEE Trans Geosci Remote Sens 38(5):2288–2298, 2000
12. Wang JZ, Li J, Wiederhold J: SIMPLIcity: semantics sensitive integrated matching for picture libraries. IEEE Trans Pattern Anal Mach Intell 23(9):947–963, 2001
13. Guliato D, Rangayyan RM, Melo EV, Soares RC: A system for content-based image retrieval and analysis of mammograms using PostgreSQL with image-handling extension. In: Proceedings of the Fifth IASTED International Conference on Biomedical Engineering, Innsbruck, Austria, February, 2007, pp 402–408

14. Alto H, Rangayyan RM, Paranjape RB, Desautels JEL, Bryant H: An indexed atlas of digital mammograms for computer-aided diagnosis of breast cancer. Ann Telecommun 58(5–6):820–835, 2003

15. Baroni MCN, Rezende HL, Traina-Jr C, and Traina AJM: Queryng complex objects by similarity in SQL*. In: Proceedings of Brazilian Symposium on Databases—SBBD2005, Uberlândia, MG, Brazil, October–November, 2005, pp 1–5

16. Böhm C, Berchtold S, Keim DA: Searching in high-dimensional spaces—index structures for improving the performance of multimedia databases. ACM Comput Surv 33(3):322–373, 2001

17. Chávez E, Navarro G, Baeza-Yates R, Marroquím J: Searching in metric space. ACM Comput Surv 33(3):273–321, 2001

18. Ciaccia P, Patella M: M-tree: an efficient access method for similarity search in metric spaces. In: Proceedings of International Conference on Very Large Data Bases (VLDB), Athens, Greece, 1997, pp 426–435

19. Traina Jr, C, Traina AJM, Faloutsos C, Seeger B: Fast indexing and visualization of metric datasets using slim-trees. IEEE Trans Knowl Data Eng 14(2):244–260, 2002

20. Digout C, Nascimento M, Coman A: Similarity search and dimensionality reduction: not all *dimensions are equally useful*. In: Proceedings of Database Systems for Advanced Applications—9th International Conference, DASFAA 2004, Jeju Island, Korea, March 17–19. Springer, Berlin, Germany, 2004 (also in Lect Notes Comput Sci 2973:831–842)

21. Kailing K, Kriegel HP, Schönauer S, Seidl T: Efficient similarity search for hierarquical data in large databases. In: Proceedings of Advances in Database Technology—EDBT 2004—9th International Conference on Extending Database Technology, Heraklion, Crete, Greece, March. Springer, Berlin, Germany, 2004 (also in Lect Notes Comput Sci 2992:676–693)

22. Vieira MR, Chino F, Traina-Jr, C, Traina AJM: Dbm-tree: a metric access method sensitive to local density data. In: Proceedings of Brazilian Symposium on Databases—SBBD2004, Brasilia, DF, Brazil, October, 2004, pp 163–177

23. Carey MJ, Kossmann D: On saying enough already in SQL. In: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, Tucson, AZ, May, 1997, pp 219–230

24. Carey MJ, Kossmann D: Reducing the braking distance of an SQL query engine. In: Proceedings of the Conf. on Very Large Data Bases (VLDB), New York City, NY. VLDB Endowment, Saratoga, CA, 1998, pp 158–169

25. Gao L, Wang M, Wang XS, Padmanabhan S: Expressing and optimizing similarity queries in SQL. In: Proceedings of Conceptual Modeling—ER—23rd International Conference on Conceptual Modeling, Shanghai, China, November. Springer, Berlin, Germany, 2004 (also in Lect Notes Comput Sci 3288:464–478)

26. Melton J, Eisenberg A: SQL multimedia and application packages (SQL/MM). ACM SIGMOD Record 30(4):97–102, 2001 (December)

27. The POSTGRES Group: The POSTGRES Reference Manual, Berkeley, CA: Computer Science Division, University of California, 1993 (January)

28. Guliato D, Rangayyan RM, Carvalho JD, Santiago SA: Polygonal modeling of contours of breast tumors with the preservation of spicules. IEEE Trans Biomed Eng 55:14–20, 2008

29. Melo EV, Guliato D, Rangayyan RM, Soares RS: SISPRIM—Sistema De Pesquisa Com Suporte Para Recupera, cão de imagens por conte'udo. In: Proceedings of WIM2006 - VI Workshop de Inform'atica M'edica, Vila Velha, ES, Brazil, June, 2006.

30. IBM: DB2 Universal Database Image, Audio, and Video Extenders Administration and Programming. 2000. http://www-306.ibm.com/software/data/db2/extenders/index.html

31. Informix: Excalibur Image Datablade Module, Users Guide. 2000. http://informix.com.ua/answers/english/alpha.htm

32. Informix: Informix Image Foundation DataBlade Module, Users Guide, Version 2.0. 2000 (December)

33. Oracle: Oracle8i interMedia Audio, Image, and Video—Users Guide and Reference. 2005. http://download.oracle.com/docs/pdf/A67296_01.pdf

34. ISO: ISO/IEC IS 13249-5:2001 SQL/MM, Information Technology Database Languages SQL Multimedia and Application Packages Part 5: Still Image. 2001.

35. Stolze K: Still image extensions in database systems—a product overview. In: Datenbank-Spektrum, 2002, pp 40–47 (February)

36. The POSTGRES Group. PostgreSQL 8.0.0 Documentation. 2005

37. Rangayyan RM, El-Faramawy NM, Desautels JEL, Alim OA: Measures of acutance and shape for classification of breast tumors. IEEE Trans Med Imag 16(6):799–810, 1997

38. Guliato D, Bôaventura RS, Melo EV, Rangayyan RM: AMDI: an indexed atlas of digital mammograms that integrates case studies, e-learning, and research systems via the web. In: Suri JS, Rangayyan RM Eds. Recent Advances in Breast Imaging, Mammography, and Computer-aided Diagnosis of Breast Cancer. Bellingham, WA: SPIE, 2006, pp. 529–555

39. American College of Radiology: Breast Imaging Reporting and Data System BI-RADS, 4th edition. Reston, VA: American College of Radiology, 2004

40. Screen Test: Alberta Program for the Early Detection of Breast Cancer—2001/03 Biennial Report. 2004. http://www.cancerboard.ab.ca/screentest

41. Alto H, Rangayyan RM, Desautels JEL: Content-based retrieval and analysis of mammographic masses. J Electron Imaging 14(2):023016, 2005

42. The Mammographic Image Analysis Society digital mammogram database. http://peipa.essex.ac.uk/info/mias.html, accessed October, 2006

43. Rangayyan RM, Mudigonda NR, Desautels JEL: Boundary modelling and shape analysis methods for classification of mammographic masses. Med Biol Eng Comput 38:487–496, 2000

44. Digital Database for Screening Mammography. http://marathon.csee.usf.edu/Mammography/Database.html, accessed June, 2007.

45. Rui Y, Yang TS, Mehrotra S: Content-based Image Retrieval with Relevance Feedback in Mars. In: IEEE International Conference in Image Processing, volume 2, Santa Barbara, CA, 1997, pp 815–818

46. Rui Y, Yang TS, Ortega M, Mehrotra S: Relevance feedback: a powerful tool in interactivecontent-based image retrieval. IEEE Trans Circuits Syst Video Technol 8(5):644–655, 1998 (September)

47. Triana AJM, Marques J, Traina Jr C. Fighting the semantic gap on CBIR system through new relevance feedback techniques. In: Proceedings of 19th IEEE Symposium on Computer-based Medical Systems (CBMS'06), 2006, pp 881–886

48. Guliato D, de Melo EV, Bôaventura RS, Janones FR, de Deus V, Rangayyan RM: AMDI: an atlas to integrate case studies, e-learning, and research systems via the Web. In: Proceedings of the IASTED International Conference on Telehealth. Banff, AB, Canada, 2005, pp 69–74

49. Guliato D, Caetano M, Janones FR, de Deus V, Lima SC, Rangayyan RM, Bôaventura RS, and Marques PMA. AMDI: An indexed atlas of digital mammograms availablevia the Web. In: III Latin American Congress on Biomedical Engineering, IFMBE Proceedings, 5, 2004 (4 pages on CDROM)