

- ORIGINAL ARTICLE -

Efficient Iris Recognition Management in Object-Related Databases

Gestión Eficiente de Reconocimiento del Iris en Bases de Datos Objetos-Relacionales

Carlos Alvez, Ernesto Miranda, Graciela Etchart, Silvia Ruiz

*Facultad de Ciencias de la Administración, Universidad Nacional de Entre Ríos Concordia, 3200, Argentina
{caralv, emiranda, getchart, sruiz}@fcad.uner.edu.ar*

Abstract

Biometric applications have grown significantly in recent years, particularly iris-based systems. In the present work, an extension of an Object Relational Database Management System for the integral management of a biometric system based on the human iris was presented. Although at present, there are many database extensions for different domains, in no case for biometric applications. The proposed extension includes both the extension of the type system and the definition of domain indexes for performance improvement. The aim of this work is to provide a tool that facilitates the development of biometric applications based on the iris feature. Its development is based on a reference architecture that includes both the management of images of the iris trait, its associated metadata and the necessary methods for both manipulation and queries. An implementation of the extension is performed for PostgreSQL DBMS, and SP-GiST framework is used in the implementation of a domain index. Experiments were carried out to evaluate the performance of the proposed index, which shows improvements in query execution times.

Keywords: IrisCode, index, database, object relational, extension.

Resumen

Las aplicaciones biométricas han crecido significativamente en los últimos años, en particular los sistemas basados en el iris. En el presente trabajo se presenta una ampliación de un Sistema de Gestión

Citation: C. Alvez, E. Miranda, G. Etchart and S. Ruiz. "Efficient iris recognition management in object-related databases", Journal of Computer Science & Technology, vol. 18, no. 2, pp. 105 - 113, 2018.

DOI: 10.24215/16666038.18.e12

Received: February 15, 2018 **Revised:** June 18, 2018
Accepted: September 4, 2018

Copyright: This article is distributed under the terms of the Creative Commons License CC-BY-NC.

de Base de Datos Objeto-Relacional para la gestión integral de un sistema biométrico basado en el iris humano. Aunque en la actualidad existen muchas extensiones de bases de datos para diferentes dominios, en ningún caso existen para aplicaciones biométricas. La extensión propuesta incluye tanto la extensión del sistema de tipos como la definición de índices de dominio para la mejora del rendimiento. El objetivo de este trabajo es proporcionar una herramienta que facilite el desarrollo de aplicaciones biométricas basadas en el iris. Su desarrollo se basa en una arquitectura de referencia que incluye tanto la gestión de las imágenes del rasgo del iris, sus metadatos asociados y los métodos necesarios, tanto para la manipulación como para las consultas. Se realiza una implementación de la extensión para PostgreSQL DBMS, y se utiliza el framework SP-GiST en la implementación de un índice de dominio. Se realizaron experimentos para evaluar el desempeño del índice propuesto, que muestra mejoras en los tiempos de ejecución de las consultas.

Palabras claves: IrisCode, índice, bases de datos, objeto relacional, extensión.

1. Introduction

In recent years, Object-Relational Databases (ORDB) has been widely used for data management in more complex applications. The main advantage of using the Object-Relational (OR) model is the possibility of extending the type system to suit the application domain. As a result, there are now many Object-Relational Database Management System (ORDBMS) extensions for specific domains in the marketplace supplied by the same database providers or specific projects, such as Oracle Multimedia, which allows video, audio and image management in Oracle [1]. Besides, Oracle Semantic Technology allows the management of semantic models in Oracle [2], and PostGIS, which allows the support of spatial and geographical objects in PostgreSQL [3], among others, attempting to support applications that use data from these domains. Domain support refers to data types and

their behavior, access methods (indexes use), specific operators, among other aspects.

Beyond the extensions provided by ORDBMS providers, the most important thing is that they enable the creation for new extensions for specific domains not covered by them. For example, Data Cartridge provided by Oracle [4], PostgreSQL extensions [5], among others. This allows the creation of database extensions in domains that are not covered by the providers, such as applications using biometric data.

In this sense, the present work expands the extension for the integral management of a biometric system based on the human iris proposed in [6], involving a domain index to achieve greater efficiency in the identification process. The extension includes the management of images of the iris trait and all its associated metadata; it also includes the necessary methods for manipulating them, as well as the methods for searching (verification/identification).

In biometrics, the size of databases is increasing rapidly. Therefore, efficient management of these databases is an increasing challenge to optimize the response time [7], particularly in the identification process. Since biometrics-based identification systems, especially iris, work with high dimensional characteristics; extensive searching in a large database increases response time. One strategy to improve this aspect is to use indexing techniques [8], as this reduces the search space for an identification system by quickly choosing a subset of iris images from the database in order to determine a possible match. In this sense, it is important for the extension to provide domain indexes associated with identification methods.

The use of indexes for biometric features is essential for the efficiency of biometric identification operations, where the captured feature should be compared with the features stored in the database to establish a person's identity.

For the verification process, the use of indexes is not critical, since the captured trait is only compared with the stored trait of the person whose identity is being verified.

While the use of indexes has a negative effect on the performance of database insertion operations, this cost is acceptable, since the enrolment process is not a frequent activity.

It should be noted that from the Multimedia perspective the term indexation takes on a different meaning from that used in the field of Database Management Systems (DBMS). In the first case, it refers to the process of assigning terms, phrases or values that represent the content of multimedia data information (in this case the iris image). This information is used for image recovery. In contrast, the notion of an index from the point of view of a

DBMS refers to access structures built to accelerate access to data. In the case of the iris image, an indexing scheme will assign an index value (scalar or vector) to each iris thereby allowing the query image to be compared against only those iris in the database that have comparable index values.

From a multimedia point of view, there are several techniques for indexing the iris [9], such as IrisCode (postcoding) or iris texture analysis (precoding). The IrisCode is a template obtained from the processing of the iris image. First, mathematical algorithms are used to locate the inner and outer edges of the iris and then the patterns are extracted and subjected to mathematical transformations until a sufficient amount of information is obtained for authentication purposes. The first technique is based on the main component analysis (PCA) of IrisCode, which is a binary representation of information extracted from the iris texture. Among the techniques that examine the texture content of the image, the Local Binary Pattern (LBP) technique analyzes the local binary pattern of iris texture; whereas the Signed Pixel Level Difference Histogram (SPLDH) technique is based on statistical analysis of pixel intensities and positions [7, 9].

Most commercially used iris recognition systems use IrisCode. This is one of the reasons why IrisCode has been chosen here to index the iris.

In summary, the present work incorporates in the extension for the integral management of a biometric system based on the human iris [6] the use of domain indexes in order to obtain acceptable response times, even for large volumes of data.

In the rest of the document, the topics are organized as follows: in Section 2, a brief description of the reference architecture is presented; in Section 3, the implementation proposal is detailed; in Section 4, the index structure and experimental results are presented; and finally in Section 5, conclusions and future work.

2. Preparation of manuscript

A software architecture for a system is the structure or structures of the system, which comprise elements, their externally-visible behavior, and the relationships among them [13].

The proposed extension is based on a reference architecture that contemplates the management of iris images and all their associated metadata (Fig. 1) [6,14]; it also includes all the necessary modules for manipulating them.

It was also considered that the architecture is in accordance with internationally recognized standards, so that it is as generic and adaptable as possible to the needs of different systems. In relation

to the latter, the ANSI / NIST ITL 1-2011 [15] standard is a biometric standard published in November 2011, which defines how to ensure the interoperability of biometric data between different systems. This standard defines the content, format, and units of measurement for the electronic exchange of fingerprint, palmprint, plantar, face recognition, iris, deoxyribonucleic acid (DNA), and other biometric samples and forensic information that may be used in the identification or verification process of a subject.

This standard defines the composition of the records comprising a transaction that might be transmitted to another organization. A transaction is made up of records. Transactions should consist of one Type-1 record and one or more of the Type-2 to Type-99 records. The Type-1 record is used to describe the transaction. The Type-17 record specifies interchange formats for biometric authentication systems that use iris recognition. In this work the transaction should contain at least one record Type-1 and Type-17 records.

Among its requirements, the standard ANSI / NIST ITL 1-2011 also has the requirements for representation and iris image compression [16,17].

In the proposed architecture, the iris image exchange system is responsible for generating, storage, transmitting and receiving standard records.

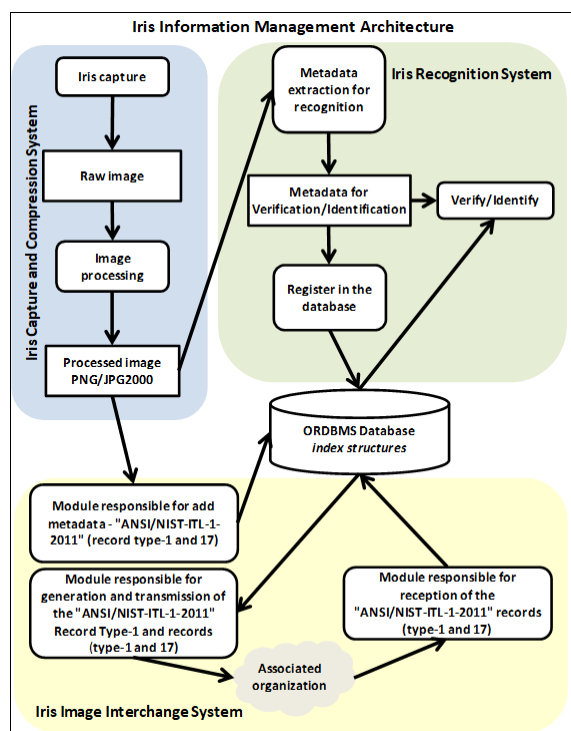


Fig. 1: Iris Information Management Architecture.

The Type-17 record of the aforementioned standard is used only for information exchange. However, generation and storage of metadata required for recognition should also be considered,

i.e. for the processes of identification and / or verification of people by means of the iris (IrisCode) [10,11].

The images that manage the systems mentioned can be obtained either from transactions with other agencies or from the system captures itself. For this, the capture system and iris image compression is used.

Both the structure Type-17 record and the metadata for iris recognition are complex structures. This can cause some problems when working with relational data model, because of the limitations imposed by it. Therefore, here we have opted for object-relational (OR) technology [18]. This technology can provide solutions to the limitations of relational databases since they provide the following abilities [19]:

1. To define data types. These types may involve complex structures such as collections, large objects, etc., without the limitation of the first normal form.
2. To define and implement the data behavior. Methods can be created to manage the data of defined types, thus facilitating safe access from applications that use, for example, the verification method that compares two IrisCodes.
3. To define and implement domain access methods. This improves the access time in recovering data from a specific domain, as in the case of biometric data.

The aforementioned facilities allow the creation of infrastructures that extend the services of the ORDBMS services [20]. This allows the extension of the ORDBMS to handle data from a specific domain, in this case biometric data.

As regards this work, this extension should include:

- The creation of specific types for biometric data. This involves types for raw data (iris images in our case), types for encoding them (IrisCode), and all the metadata required to generate ANSI/NIST ITL1-2011 transaction records.
- Defining and implementing the methods managing the types previously defined. This includes, among others, the generation of specific encodings, matching methods and methods of generation and import of ANSI / NIST ITL1-2011 records.
- The creation of domain indexes to improve the response time in the identification processes: in biometric identification, you need to access multiple records to identify a person. Improving the efficiency in this access is not simple to be achieved with traditional indexes like B-trees, hash, etc. This is because the codes (templates) generated for comparison are multidimensional data, and methods for specific access domain are needed.

Related to the last point, the implementation of the extension is intended to include indexing methods based on IrisCode [7,8,21], in order to allow simple and efficient data management, even for large volumes of data.

The following section presents the implementation of the extension based on the reference architecture presented in this section.

3. Extension implementation

One of the main decisions for implementing the extension is the choice of the ORDBMS to be used. The DBMS PostgreSQL has been chosen because it is open source and offers many alternatives to generate new access methods and operators. This last point is important, because one of the main challenges in biometric databases is to improve the response time in identification.

On the other hand, different libraries and frameworks, most of them open source, were studied both for image processing and compression as well as for obtaining the IrisCode, which could be useful for the creation of the extension.

One of the libraries used is OpenCV [22] which allows image processing to support some of the more generic functions.

To obtain the IrisCode, the following systems were analyzed: Masek and Kovsesi [23] at the University of Western Australia for open source and modular iris recognition (in MATLAB); OSIRIS (Open Source for IRIS) [24] is an iris recognition system developed as a framework for the BioSecure project (2007); the VASIR (Video-based Automatic System for Iris Recognition) framework [25] with iris recognition algorithms implemented by NIST to be used in video capture. The code of the VASIR framework is used in this work. The choice of this framework regarding the other open source options is mainly motivated by the following aspects: on the one hand, it enables the acquisition for the IrisCode, which is the chosen indexing technique. On the other hand, its ability to analyze videos as well as static images in iris recognition. It also has effective algorithms for verification and identification of subjects under a wide range of images and different environmental conditions (NIST implemented algorithms).

The implementation is based on the three systems of the reference architecture discussed in the previous section and will be described below.

3.1. Capture and processing system

In this system, the OpenCV library was used to obtain raw images. This includes face detection in the image, left eye and right eye as presented in Fig.

2. From this, images of both eyes are obtained. OpenJPEG and LIBPNG libraries were then used for image processing. These libraries allow obtaining JPEG2000 and PNG formats respectively, which are supported by the ANSI/NIST-ITL1-2011 standard. Captured images are stored in instances of IMGJPG2000 and IMGPNG object types defined in the proposed extension (Fig. 3 [14]). These types inherit from the object type ProcessedImage related to the OpenJPG and LIBPNG libraries.

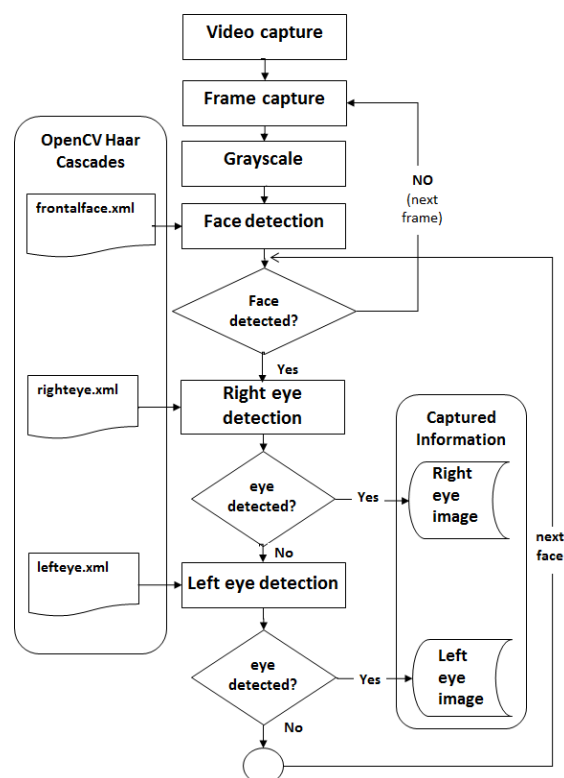


Fig. 2. Iris capture process.

3.2. Iris Image Exchange System

This system contains the import and export operations of Type-17 records established in the ANSI/NIST-ITL1-2011 standard. The format used is NIEM [26] (National Information Exchange Model) which specifies a namespace for the exchange of biometric data using XML format documents. An XML schema document (XSD) is defined, which allows validation of XML documents generated in the export process or before the import process to ensure document consistency.

The import process, implemented in PL/PgSQL, consists of taking the XML document generated in another organism. The XML Parser then validates that the document is well formed and verifies its validity against the NIEM schema. Once the validity has been checked, the fields are processed and a connection is generated with BDOR in order to store the data in it.

The export process consists of taking the data

from the database, processing the fields and building the XML document for export. Once this process is completed, it can be verified, with the use of the Parser, that the generated document is correct before sending it to another agency for import.

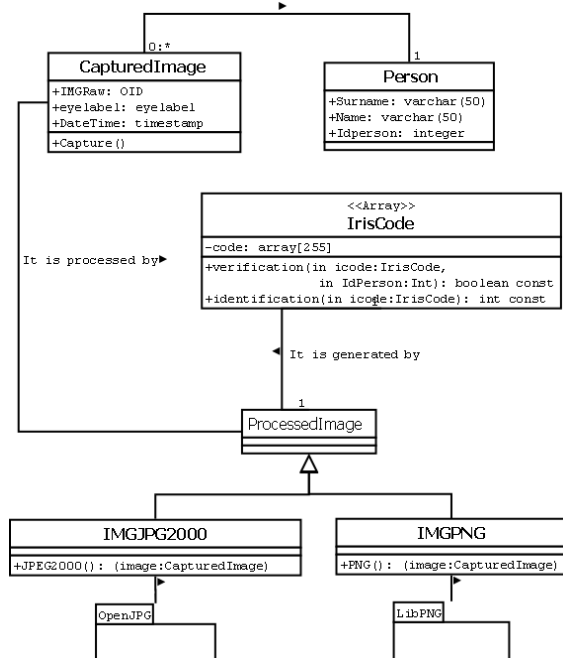


Fig. 3. Class diagram with object types required for eye capture, image processing and iris recognition.

3.3. Iris recognition system

This system is responsible for IrisCode encoding (Fig. 4 A). First, the iris is located and normalized as pre-processing tasks for a captured eye image. To locate the iris part, scale reduction and color level transformation are applied. For the processing and management of both images and iris code, different object types were created. These types contain methods for their behavior, for example, an instance of the IrisCode object type (Fig. 3), has verification and identification member methods. The verification method receives as an input an IrisCode instance, and the identity of the person who claims to be; the method returns true if the Hamming distance of this instance from the instance stored in the database for this person is within the set threshold. The identification method receives as input an IrisCode instance and returns the identification of the person, whose Hamming distance from his IrisCode instance stored in the database, is the smallest and within the established threshold. If no one is below the threshold, it means that no one corresponding to that IrisCode is found.

With these methods, comparisons (either verification or identification), can be made with simple SQL queries (Fig. 4 B), using the methods of the instances stored in a table. For example, the

Algorithm 1 shows the use of the verification method in PL/PgSQL:

Algorithm 1 Use of the verification method

```

1: CREATE FUNCTION chkIris(irisC IrisCode,
   threshold numeric)
2: RETURNS TABLE(Idperson int, Surname
   varchar(50), Name varchar(50)) AS $
3: BEGIN
4: RETURN QUERY
5: SELECT p.Idperson, p.Surname, p.Name
6: FROM Persons AS p
7: WHERE p.IrisCod.identification(irisC) <
   threshold
8: END;
9: $ LANGUAGE plpgsql;
    
```

The use of domain indexes, as presented in Section 4, allows the response time of these queries to be optimized by reducing the number of candidates that the iris matching algorithm should consider

4. Index structure

This section examines the implementation of the domain index for iris biometrics, and presents the results of the identification process with and without indexation.

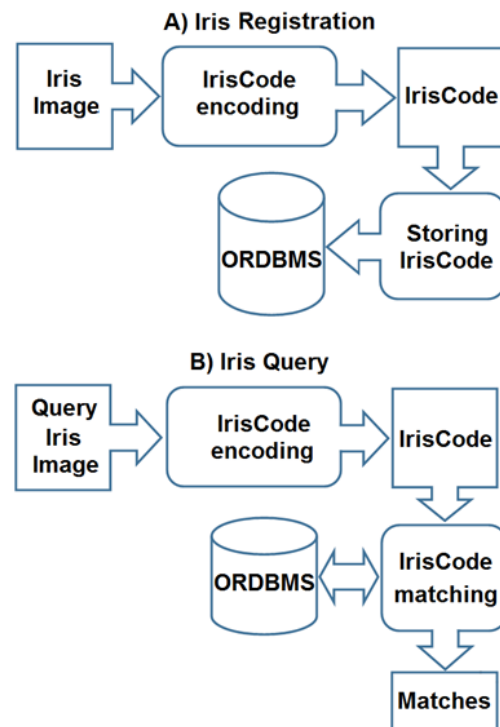


Fig. 4: A) Iris registration pipeline; B) Iris Query pipeline.

4.1. Domain index

In biometrics, as in other domains, complex data are

manipulated. To efficiently recover this type of data, it is possible to use an indexing method in a metric space developed in the area of similarity search [27]. The objective of an indexing method in a metric space is to find a set of objects (such as images or videos) in the database whose distance (or similarity) with the object of query is less than or equal to a threshold.

Essentially, a metric space is a set of objects equipped with a distance function $d(\mathbf{a}, \mathbf{b})$ for each pair of elements (\mathbf{a}, \mathbf{b}) . This distance function must satisfy a set of axioms to ensure a good behavior. These axioms are the following.

For all elements $\mathbf{a}, \mathbf{b}, \mathbf{c}$ it must satisfy:

- non-negativity: $d(\mathbf{a}, \mathbf{b}) \geq 0$
- $d(\mathbf{a}, \mathbf{b}) = 0$ implies $\mathbf{a} = \mathbf{b}$ (and vice versa)
- symmetry: $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$
- triangle inequality: $d(\mathbf{a}, \mathbf{b}) \leq d(\mathbf{a}, \mathbf{c}) + d(\mathbf{c}, \mathbf{b})$

The indexing method in a metric space reduces the number of scoring calculations between the query object and the database objects using an index that is built on the basis of scores. The main feature of this method is that it can be applied as long as a scoring measure (i. e. a measure of distance or similarity between objects) is defined. In this work, the measure of similarity used is Hamming distance.

As explained in the preceding paragraphs, with regard to DBMS-level indexing, the use of BK-Tree [28] is proposed for efficient retrieval of the IrisCode, because this tree-shaped data indexing method serves to index information in a metric space using a distance measurement, in this particular case, the Hamming distance.

4.2. BK-Tree preliminaries

For a better understanding of the implementation of the index, some BK-Tree concepts are briefly presented in this subsection.

In BK-Tree an element \mathbf{a} is arbitrarily selected as the tree root. The root may have zero or more subtrees. The n -th sub-tree is recursively built of all elements \mathbf{b} , whose distance from the root \mathbf{a} is equal to n ($d(\mathbf{a}, \mathbf{b}) = n$).

For the purpose of providing an example, IrisCode with a reduced length of 8 bits will be used.

Assuming that the following IrisCodes should be inserted in the database: 10001110 - 11101110 - 10011110 - 10001101 - 01101110 - 11011110

Any IrisCode can be selected as a root node, in this case the first one is chosen (10001110). Then, each IrisCode is added under the arc labelled with the distance d to the root node. In Fig. 5A, the tree is shown after inserting the following two elements (11101110 - 10011110).

To insert the next element (10001101) the Hamming distance from the tree root is calculated. Since this distance equals 2 ($d = 2$), analyze the

subtree with the arc labeled 2. Then, it calculates the distance d with the root node of that subtree (11101110). This distance equals 4, so a new arc labelled with 4 is added and the new node is inserted under this new arc. At the end of the rest of the insertions, the tree shown in Fig. 5B is obtained.

The search is intended to find those IrisCodes stored in the database, closer to an IrisCode belonging to a recently captured image. For this purpose, a threshold N is defined which represents the maximum allowed distance. The algorithm proceeds as follows:

1. Creates a candidates list and adds the root node to it.
2. Takes a candidate and calculates the distance D with the IrisCode of the query, and compares it with the threshold.
3. Adds to the list of candidates, all children of the current node such that, from their parent node, have a distance of $D - N$ and $D + N$ inclusive.

As an example, we will look for all the IrisCodes that are within a Hamming distance N of no more than 1 of the IrisCode 10101110. The root node is then analyzed and the distance is calculated.

$$D = \text{hamming_distance}(10101110, 10001110) = 1$$

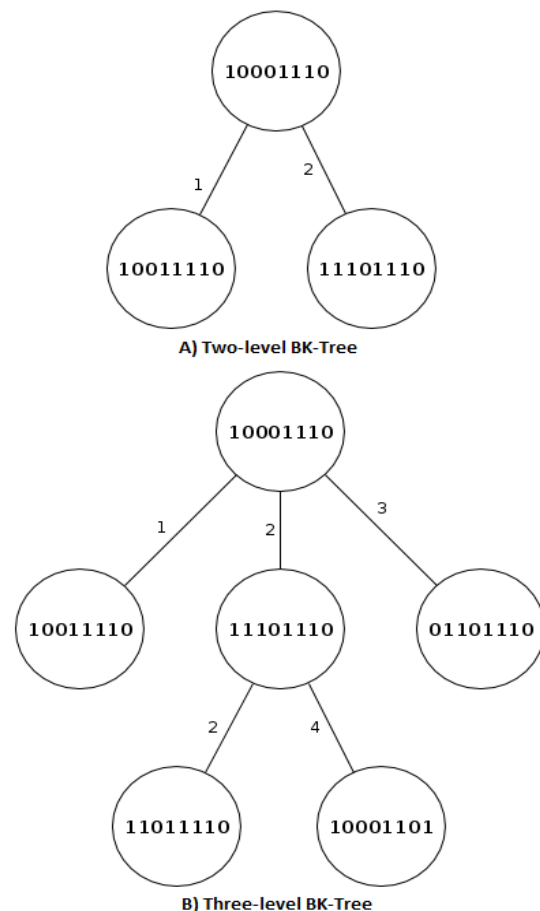


Fig. 5: A) Two-level BK-Tree (three elements); B) Three-level BK-Tree (six elements).

Since threshold $N = 1$, the root node is added to the list of results. The entire list of candidates is then extended to all child nodes that have a distance from the root node between $D - N = 0$ and $D + N = 2$.

D is computed for each of these nodes:

$$D = \text{hamming_distance}(10101110, 10011110) = 2$$

Since $D > N$, this is not a result and the next node continues to be analyzed:

$$D = \text{hamming_distance}(10101110, 11101110) = 1$$

Since $D = N$, this node is added to the list of outcomes and children with a distance between $D - N = 0$ and $D + N = 2$ are analyzed:

Since $D > N$ this node is not acceptable.

Finally, the results list is formed by nodes 10001110 and 11101110 having analyzed only 4 of the 6 nodes in the tree.

The implementation of the BK-Tree index in PostgreSQL uses the SP-GiST framework [29] as it allows the creation of non-traditional indexes belonging to the spatial partitioning tree class. In this way, with the generalized search tree provided by SP-GiST it was possible to develop specific data types to index the IrisCode with the appropriate access methods.

Generally, the Hamming distance is represented by an integer that indicates the amount of bits that differ between the objects compared. In this work an alternative method was developed to show this value as a percentage, independent of the amount of bits used for the templates. This allows the thresholds to be expressed in meaningful values for those who set them. It also allows changes in the number of bits representing the IrisCode without altering the thresholds.

4.3. Experimental evaluation

This subsection shows experimental results to evaluate the performance of the proposed index in the identification process.

The experimental evaluation used a table containing the IrisCode (2048 bits) and an associated person identifier, generated from the CASIA-Iris-Thousand subset of the CASIA Iris Image Database V4.0 database [30]. The total execution time required by the query was compared using the implemented BK-Tree index on the one hand, and without index (exhaustive search in the database) on the other hand. Several database sizes and different percentages of Hamming distance were used for this comparison. The test was carried out on a laptop with an I5 350M processor and 4 GB memory,

making five runs for each consultation scheme.

Tables 1 and 2 shows the results of the average execution time of the query, expressed in seconds.

Table 1 AVG execution time (in sec) for a query with different database sizes, without indexing.

Hamming Distance (%)	Without indexing				
	DB Size (records)				
	1000	5000	10000	50000	100000
1%	5.0955	21.2598	42.2663	210.1412	421.5418
3%	4.2864	21.2455	42.2727	209.8602	421.4073
5%	4.2372	21.4796	42.1594	211.4335	422.4264
7%	4.2304	21.0598	42.0525	211.3277	420.7906
9%	4.2307	21.0554	42.3097	211.3646	420.5675

Table 2 AVG execution time (in sec) for a query with different database sizes, with BK-Tree indexing.

Hamming Distance (%)	With BK-Tree indexing				
	DB Size (records)				
	1000	5000	10000	50000	100000
1%	0.0078	0.0109	0.0179	0.027	0.0359
3%	0.0044	0.0147	0.0199	0.0505	0.1062
5%	0.0045	0.0128	0.0178	0.0737	0.1858
7%	0.0071	0.0143	0.0342	0.1029	0.2006
9%	0.0064	0.0178	0.0242	0.1213	0.2409

From the experimental results obtained, it appears that the proposed index requires less runtime in the identification process regardless of the percentage of Hamming distance used, compared to non-indexed query. The latter, for different Hamming distance values, the execution time is practically constant for the same database size. This is because this query scheme requires a sequential scan that simply calculates the scores for all objects in the database. Based on the results, it can be seen that the index implemented significantly improves the average time of execution of the consultation in the identification process with respect to the non-indexed query.

5. Conclusions and future work

In the present work an extension of an ORDBMS for the integral management of a biometric system based on the human iris was presented. Currently, there are many database extensions for different domains, however, in no case does it do it for biometric applications. The extension includes the definition of both object types and domain indexes.

The purpose of this extension is that the management of the data corresponding to this domain be simple and efficient, that is, that SQL queries can be made using the search methods defined in the extension of the type system (even by users who are not experts in the domain) and obtain answers in reasonable time. This facilitates the

development of biometric applications based on the iris feature.

Its development is based on a reference architecture that includes both the management of images of the iris feature, its associated metadata, the necessary methods for manipulating them, as well as searching methods (verification/identification).

The implementation of the proposed extension expands the PostgreSQL DBMS-like system using open source frameworks and libraries available in the industry.

An important point sought in this implementation is efficiency in the identification process. In this sense, the SP-GiST framework was used in the implementation of the BK-Tree index to achieve greater efficiency in the identification process. The use of BK-Tree is proposed because it serves to index information in a metric space using a measure of distance, in this particular case, the Hamming distance. To evaluate the performance of the proposed index, experiments were conducted. The obtained results show that the performance of the implemented index outperforms the exhaustive database search in all datasets for query runtime.

As future work, experiments with larger iris databases and comparisons with other indexing alternatives are proposed.

Competing interests

The authors have declared that no competing interests exist.

References

- [1] S. Pelski, *Oracle Multimedia Reference*, 11g Release 2 (11.2). Part. E10776-03, 2010.
- [2] C. Murray, *Oracle Database Semantic Technologies Developer's Guide*, 11g Release 2 (11.2). Part. E25609-06, 2014.
- [3] "PostGIS 2.3.4 Developer Manual - SVN Revision (15474)". Available at: <http://postgis.net/stuff/postgis-2.3.pdf>. Accessed on 2016-12-15.
- [4] E. Belden, C. Timothy and D. Dinesh, *Data Cartridge Developer's Guide - 11g Release 2 (11.2) - Part. E10765-02*, 2010.
- [5] "Extensiones en PostgreSQL. Chapter 35. Extending SQL". Available at: <https://www.postgresql.org/docs/9.5/static/extend.html>. Accessed on 2016-12-15.
- [6] C. Alvez, G. Etchart, S. Ruiz, E. Miranda and J. Aguirre, *Extensión de una base de datos Objeto-Relacional para el soporte de datos de iris*. XXIII Congreso Argentino de Ciencias de la Computación (CACIC 2017), pp. 578-587, 2017.
- [7] R. Mukherjee and A. Ross, *Indexing iris images*. 19th International Conference on Pattern Recognition, Tampa, FL, pp. 1-4, 2008.
- [8] N. Poonguzhali and M. Ezhilarasan, *Iris Indexing Techniques: A Review*. International Journal of Computer Applications. no. 73. pp 23-29, 2013.
- [9] R. Mukherjee, *Indexing Techniques for Fingerprint and Iris Databases*. Master thesis, College of Engineering and Mineral Resources at West Virginia University, 2007.
- [10] J. Daugman, *High con dence visual recognition of people by a test of statistical independence*. IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 15(11), pp. 1148-1161, 1993.
- [11] J. Daugman and C. Downing, *Effect of severe image compression on iris recognition performance*, IEEE Transactions on Information Forensics and Security, vol. 3, no. 1, p. 52-61, 2008.
- [12] A. W. K. Kong, D. Zhang and M. S. Kamel, *An Analysis of IrisCode*, in IEEE Transactions on Image Processing, vol. 19, no. 2, pp. 522-532, 2010.
- [13] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord and J. Stafford, *Documenting Software Architectures: Views and Beyond*. 2nd Edition. Addison-Wesley Professional. 2010.
- [14] S. Ruíz, G. Etchart, C. Alvez, E. Miranda, M. Benedetto and J. Aguirre, *Iris Information Management in Object-Relational Databases*. XXI Congreso Argentino de Ciencias de la Computación (CACIC 2015), pp. 741-750, 2015.
- [15] B. Wing, ANSI/NIST-ITL 1-2011. *Update: 2015. Information Technology: American National Standard for Information Systems Data Format for the Interchange of Fingerprint, Facial; Other Biometric Information*, 2015.
- [16] ISO/IEC 15444-1:2004 *Information Tecnology-JPEG 2000 Image Coding System: Core Coding System*, Second Edition, 2004.
- [17] ISO/IEC 15948:2004 *Information Tecnology-Computer Graphics and Image Processing Portable Netware Graphics (PNG): Functional Specification*, First Edition, 2004.
- [18] J. Melton, *(ISO-ANSI Working Draft) Foundation (SQL/Foundation)*, ISO/IEC 9075-2:2003 (E), United States of America (ANSI), 2003.
- [19] C. Alvez and A. Vecchietti, *Combining Semantic and Content Based Image Retrieval in ORDBMS*. Knowledge-Based and Intelligent Information and Engineering Systems Lecture Notes in Computer Science, Springer-Verlag

- Berlin Heidelberg, vol. 6277/2010, pp. 44-53, 2010.
- [20] C. Alvez, *Models for the recovery of similarity-based images in Object-Relational Databases*. PhD thesis. Universidad Tecnológica Nacional, Regional Santa Fe, Argentina, 2012.
- [21] H. Proença, *Iris Biometrics: Indexing and Retrieving Heavily Degraded Data*. IEEE Transactions on Information Forensics and Security, vol. 8, no. 12, pp. 1975-1985, 2013.
- [22] R. Laganiere, *OpenCV 3 - Transforming and Filtering Images [Video]*. Packt, 2017.
- [23] L. Masek, *Recognition of human iris patterns for biometric identification*. Master thesis. School of Computer Science and Software Engineering, University of Western Australia, 2003.
- [24] N. Othman, B. Dorizzi and S. Garcia-Salicetti, *Osiris*. Journal Pattern Recognition Letters archive. vol. 82 Issue P2, pp. 124-131. Elsevier Science Inc. New York, NY, USA, 2016.
- [25] Y. Lee, R. J. Micheals, P. J. Phillips and J. J. Filliben, *VASIR: An Open-Source Research Platform for Advanced Iris Recognition Technologies*, Journal of Research of NIST, vol. 118, pp. 218-259, 2013.
- [26] "NIEM National Information Exchange Model: Biometric Schema Version 1.0". Available at: <http://release.niem.gov/niem/domains/biometrics/3.2/>. Accessed on 2017-03-30.
- [27] E. Chavez, G. Navarro, R. Baeza-Yates, and J. L. Marroquin, *Searching in metric spaces*. ACM Computing Surveys, vol. 33(3), pp. 273-321, 2001.
- [28] W. Burkhard and R. Keller, *Some approaches to best-match file searching*. Communications of the ACM. ACM New York, NY, USA, vol. 16(4), pp. 230-236, 1973.
- [29] W. Aref and I. Ilyas, *SP-GiST: An Extensible Database Index for Supporting Space Partitioning Trees*, Journal of Intelligent Information Systems (JIIS), (2/3), pp. 215 -240, 2001.
- [30] "CASIA Iris Database, CASIA-IrisV4- Thousand". Available at: <http://http://biometrics.idealtest.org/#>. Accessed on 2018-01-03.