Content Based Image Retrieval CBIR Confocal Laser Endomicroscopy

Ruben Tous, Jaime Delgado and Olga Ferrer-Roca

***Abstract*— Content-based Image Retrieval (CBIR) is the problem of searching for digital images by analyzing the actual contents of the image rather than the structured (metadata) or unstructured text annotations associated with the image. In medical imaging, CBIR is being applied to support educational activities, research studies and even the clinical decision making process itself. In one hand this article gives an introduction to generic content-based image retrieval, its related technologies, standards and its future directions (e.g. Cloud computing). On the other hand, the paper describes an example CBIR application in the medical domain. The design of a CBIR-based computer-aided diagnosis system for Confocal Laser Endomicroscopy (CLE) images is described.**

***Index Terms*— Medical image search, CBIR, multimedia, images, metadata, search and retrieval, standard, MPEG, JPEG, MPQF, JPSearch, query format, cloud computing.**

I. INTRODUCTION

Content-based Image Retrieval (CBIR) is the problem of searching for digital images in large databases by considering the actual contents of the image rather than the metadata such as keywords, tags, and/or descriptions associated with the image [1]. CBIR extends traditional Information Retrieval (IR) approaches to digital image search by incorporating into the process information which can only be obtained by directly analyzing the image's pixel values. Processing this information requires extracting and representing the visual features of the image in a structured way, a step which is accomplished through the application of Computer Vision techniques. This paper gives an introduction to CBIR, its related technologies and standards (e.g. MPEG Query Format) and its future directions. Regarding future trends, the paper explores the opportunities that Cloud computing offers to CBIR processing. CBIR systems pose a computational challenge in terms of scalability and real-time performance. Large scale CBIR processing falls within the boundaries of what is currently known as *Big Data*, thus making traditional methods useless. Solutions to this problem could benefit from the usage of a large number of computers (a cluster) and using frameworks such as MapReduce [7]. On the other hand this paper describes an example CBIR system in the medical domain. The design of a case-based computer-aided diagnosis system that assists physicians and other medical personnel in the interpretation of optical biopsies obtained through Confocal Laser Endomicroscopy is described. An optical biopsy (OB) [2][3] is a non-intrusive optic diagnosis method, capable to analyze the tissue in surface and in deepness using laser, OCT, infrared, fluorescence, spectroscopy or other methods. This means that it is not necessary to extract the tissue from the body. The example system focuses on OB obtained through confocal laser endomicroscopy (CLE). CLE is a confocal microscopy that obtains histological images closer to the field of pathologists than endoscopists. It is therefore reasonable the lack of confidence on their interpretation. To solve the problem without the need of teleconsultation with a pathologist, the endoscopists could consult a medical images search system such as the one described in this paper. The system enables retrieving information about precedent diagnostics by providing an example OB image for CBIR, by using keywords, or by filtering different fields for structured retrieval.

Manuscript received January 11, 2011. This work was supported in part by the Spanish government (TEC2008-06692-C02-01).

Ruben Tous and Jaime Delgado are with the Universitat Politècnica de Catalunya. BarcelonaTech (UPC), Barcelona, Spain. (E-mail: rtous@ac.upc.edu).

Olga Ferrer is with the University of La Laguna. UNESCO Chair of Telemedicine. Full Professor of Pathology. La Laguna, Canary Islands, Spain.

II. CBIR PROCESS

A. Image processing and analysis

CBIR systems involve many different techniques at several stages in the indexing and retrieval workflow. The first and probably the more challenging task consists of extracting the vector of discriminant visual features (e.g. color histogram, shape boundaries, particles count, etc.) which will be indexed for every image in the database. Early works on CBIR were traditionally based on computing color, shape, and texture based features from images, allowing only evaluating the visual similarity of images in a very generic way. Later approaches for CBIR extract more sophisticated descriptors such as point based features (e.g. the Scale Invariant Feature Transform, SIFT [19]). The set of selected visual features to be extracted depends on the application scenario, as the concept of "visual similarity" is not absolute but relative to the goals of the application. In medical applications, the selected set of visual features tend to be very specific, in order to capture the particular visual treats which are relevant from a diagnosis point of view. The visual feature extraction task is commonly performed through three different steps:

1) Image preprocessing: Facilitates the analysis step by removing expendable information (e.g. noise) and

ISBN: ?????

reducing computation costs (e.g. downscaling). Usual preprocessing techniques are various forms of histogram manipulation (e.g. normalization and equalization), filtering (e.g. gaussian blur, Sobel operator for edge detection, etc.) and downscaling.

2) Image segmentation: Partition the image into two or more sets of disjoint pixels, usually represented by filling them with a different color, thus creating homogeneous (and disjoint) areas. An important particular case is binary segmentation, where one splits pixels only into foreground (the objects of interest) and background (which surrounds these objects). In binary segmentation, the detected objects are sometimes called particles. Popular segmentation techniques are *thresholding*, which assigns pixels to classes depending solely on their grey value, not on their spatial position, *histogram-based* (e.g. Otsu's thresholding), *region growing methods* (e.g. *watershedding*).

3) Feature extraction: The results of the segmentation step must be further analyzed to obtain numerical values which capture the particular morphology of the different particles or regions and their boundaries. Example features could be the number of particles, their solidity, roundness, etc. and the mean and deviation of the different values.

A popular software for image processing and analysis tasks is ImageJ. ImageJ is written in Java and open source. It provides an API and a GUI.

B. *Indexing and retrieval*

Once we have the feature vectors of all the images in the database we need to decide 1) how to store them efficiently, and 2) how to, given the feature vector of an example image, retrieve the most "similar" images in the database. "Similarity" in this context is formalized as a *distance function*. A *distance function* or *metric* is a function which, according to a specific set of rules, provides a particular way of measuring how "close to" or "far away from" two elements of some space are. In a CBIR system, a distance function has to be selected to compute the dissimilarity between the feature vector of the query image and the feature vectors of the images in the database. Once a distance function has been selected, we need to solve the problem of finding the images with a minimum distance to a given example image. This is a particular case of the *nearest neighbor* (NN) problem. A naive way for this problem is as follows: given a query q, compute the distance from q to each feature vector in the database, and report the point with the minimum distance. This *linear scan* approach has linear complexity time ($\Theta(dn)$). This approach is infeasible, as is not scalable in practical settings (e.g. Flickr has over 4 billion images [20]). Instead of doing exact *nearest neighbor* search through linear scan, a faster indexing method with sublinear ($o(n)$), logarithmic ($O(\log n|)$) or even constant ($O(1)$) time is required. Over the past decade, a new type of techniques known as the *approximate nearest neighbor* (ANN) have been developed for large scale CBIR applications. Techniques for solving the approximate nearest neighbor problem include many tree-based methods and also the recently popular hashing based approach, which can reach constant query time. Locality Sensitive Hashing (LSH) [21] is arguably the most popular unsupervised hashing methods in CBIR. Another effective method called Spectral Hashing (SH) was proposed recently by Weiss et al. [22].

C. *Image classification*

A CBIR system does not require an image classification step. However, sometimes the extracted low-level features can be automatically interpreted to infer a high level classification of the images. This classification can serve to annotate the images with semantic metadata, thus enabling advanced querying capabilities. When classification is the only goal, for instance for automatic diagnosis in medicine, then we do not talk about CBIR but only about Computer Vision.

Classification can be performed through many different ways (e.g. by manually designing a decision tree), but it's normally systematized through the usage of *supervised learning* methods. *Supervised learning* is the *machine learning task* of inferring a function from supervised (labeled) training data. A wide range of supervised learning algorithms is available, each with its strengths and weaknesses (bias-variance tradeoff, complexity, etc.). Popular learning algorithms are *Support Vector Machines* (SVM), *linear regression*, *logistic regression*, *naive Bayes*, *linear discriminant analysis*, *decision trees*, *k-nearest neighbor* algorithm, and *Neural Networks*. A popular software for classification tasks is Weka (Waikato Environment for Knowledge Analysis). Weka is written in Java and available under the GNU General Public License.

Figure 1. Weka - Main GUI appearance

D. *Performance evaluation*

The final result of a CBIR system is a collection of result items (normally but not necessarily images) for a given user query. Queries are formal statements capturing user's information needs, and several objects may match the query with different degrees of relevancy. As relevancy is subjective, all common performance measures assume a ground truth notion of relevancy: every document in the evaluation data set is known to be either relevant or non-relevant to a particular query. When evaluating an Information Retrieval (IR) system one must notice that it is in fact a particular case of a classification system in which there are only two classes, the objects that match the query

ISBN: ?????

and the objects that do not match the query. In a classification system evaluation we call *positive* of a class *C* those retrieved objects which belong to that class and *negative* of a class *C* those retrieved objects which belong to other classes. In order to evaluate a classification system we compare the classification results (the predicted classes) against the actual classes of the test set. This comparison is visualized through a *confusion matrix* (a.k.a contingency table or error matrix). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The table reports, for a given class, the number of *false positives* (objects which the algorithm has erroneously assigned to the class), *false negatives* (objects which belong to the class but the algorithm has erroneously assigned to other classes), *true positives* (objects that the algorithm has correctly assigned to the class), and *true negatives* (objects which do not belong to the class and the algorithm has correctly assigned to other classes). There are several statistical measures which operate over these values to express how well the classification has performed. Intuitively, *accuracy* should be enough:

$$accuracy_C = \frac{truePositives + trueNegatives}{truePositives + falsePositives + trueNegatives + falseNeg}.$$

Accuracy is the proportion of correct results (both true positives and true negatives) for a given class. However, if the size of the classes is unbalanced (e.g. one class containing almost all the objects) then the *accuracy paradox* states that this measure becomes useless, as we would probably be more interested in the classification success for the smaller class without considering the overall results. For instance, when classify images of a certain tissue into “normal” and “malignant”, if we classify all as “normal” we will obtain a good *accuracy*. So, in general is better to avoid the accuracy metric in favor of other metrics such as *precision* and *recall*, or the compact *F-measure* among others:

$$precision_C = \frac{truePositives}{truePositives + falsePositives}$$

$$recall_C = \frac{truePositives}{truePositives + falseNegatives}$$

In a IR system, in which we only have two classes (“matching” and “not matching”) we are normally interested in evaluating only the “matching” class, thus when we talk about *positives* we refer to the matching or retrieved objects and *negatives* to the non retrieved objects.

E. Standards supporting CBIR

The need for any standard which is basically an agreement between interested parties comes from an essential requirement: interoperability. Standardization in the area of content retrieval architectures is essential for decoupling the different system components and for facilitating the deployment of distributed repositories. There are two main areas related with CBIR in which standardization has a relevant role, *visual descriptors* and *query languages*. Regarding visual descriptors, the more relevant initiatives have been lead by the ISO/IEC JTC1 SC29/WG11 group (Moving Picture Experts Group, MPEG). The MPEG-7 standard (ISO/IEC 15938-x) specifies several visual descriptors in XML format such as the Color Layout Descriptor (CLD) or the Scalable Color Descriptor (SCD). Recently, MPEG has initiated an standardization activity to specify Compact Descriptors for Visual Search (CDVS).

Regarding the standardization of an unified query language, two different ISO working groups have developed standards which extend traditional query languages with CBIR functionalities. In one hand, ISO JTC 1/SC 32/WG4 (SQL/Multimedia and application packages group) has developed the ISO/IEC 13249 (SQL Multimedia) standard, which defines a number of packages of generic data types common to various kinds of data used in multimedia and application areas, to enable that data to be stored and manipulated in an SQL database. On the other hand, MPEG has developed the ISO/IEC 15938-12:2008 [15][16][17] (MPEG Query Format or MPQF). MPQF is an XML-based query language that defines the format of queries and replies to be interchanged between clients and servers in a distributed multimedia information search-and-retrieval context. MPQF is an XML-based in the sense that all MPQF instances (queries and responses) must be XML documents. One of the key features of MPQF is that it is designed for expressing queries combining the expressive style of Information Retrieval (IR) systems (e.g. query-by-example and query-by-keywords) with the expressive style of XML Data Retrieval (DR) systems (e.g. XQuery), embracing a broad range of ways of expressing user information needs.

In ISO/IEC 15938-12:2008, the CBIR criteria can be formulated in several different ways. The CBIR’s query-by-example technique relies in expressing user information with one or more example digital objects (e.g. an image file). Low-level features description instead of the example object bit stream is also considered query-by-example, in MPQF these two situations are differentiated, naming *QueryByMedia* to the first case (the digital media itself) and *QueryByDescription* the second one. In the first case is the query processor who decides which features to extract and use, and in the second case is the requester who perform the feature extraction and selection. The MPQF’s *QueryByMedia* type offers multiple possibilities to refer to the example media, as just including the media identifier (a locator such as an URL pointing to an external or internal resource) or directly embedding the image bit stream in Base64 encoding within the XML Query (see example in Code 1).

```
<MpegQuery>
 <Query>
  <Input>
   <QueryCondition>
     <TargetMediaType>image/jpeg</TargetMediaType>
     <Condition xsi:type="QueryByMedia">
      <MediaResource xsi:type="MediaResourceType">
       <MediaResource>
        <InlineMedia type="image/jpeg">
         <MediaData64>R0lGODlhDw…</MediaData64>
        </InlineMedia>
       </MediaResource>
      </MediaResource>
     </Condition>
```

```
    </QueryCondition>
   </Input>
  </Query>
</MpegQuery>
```

Code 1: *QueryByMedia* example

When the *QueryByMedia* type is used, it is up to the query processor to extract the proper low-level features to perform a similarity search over the index. MPQF does not specify which parameters or algorithms must be applied. In our case image analysis automatic extraction is done whenever possible. The other way to express a CBIR condition in MPQF is the *QueryByDescription* type. While the *QueryByMedia* query type uses a media sample such as image as a key for search, *QueryByDescription* allows querying on the basis of an XML-based description.

III. CBIR ON CLOUD COMPUTING

A. CBIR's increased computational demands

CBIR technologies offer many advantages over purely text-based image search. However, one of the problems associated with CBIR is the increased computational cost arising from tasks such as image processing and analysis or pattern recognition. Consequently CBIR systems pose a challenge both in terms of scalability and real-time performance. In one hand, real-time performance is usually approached with algorithmic optimizations and hardware support. The use of hardware platforms with parallel processing is now generally accepted as necessary to support real-time image analysis applications [4]. On the other hand, scalability of CBIR systems has become a popular topic nowadays as the amount of community-contributed digital images such as photos and videos has drastically increased due to the proliferation of digital cameras first, and smartphones later, along with the success of Web 2.0 applications. For instance, according to [5], more than 500 billion digital photos are taken each year, which, on average, cover the equivalent of the total surface area of all cities on the planet at least once per year [6]. A significant part of these visual data becomes publicly available through social networking sites, blogs, microblogging sites, wikis, photo and video sharing sites, and folksonomies, often accompanied by explicit structured (e.g. geo-location, timestamps and other EXIF fields) and unstructured text annotations. These data is continuously uploaded, thus constituting a mighty multimedia stream which requires capabilities that go beyond the traditional 'compute and storage' paradigm and fall within the "Big-Data" territory. Hence, a distributed stream processing system is necessary to capture, filter and analyze the image and metadata stream to enable effective, incremental classification, and knowledge analysis. The topic of dealing with images with Big-Data constraints is known as "large scale image processing".

B. Cloud computing supporting CBIR

Cloud computing is a general term for encompassing technologies that provide computation, software and data storage without requiring end-user knowledge of the component devices or infrastructure required to provide the service. These technologies typically involve provisioning of dynamically scalable and often virtualized resources. Besides some popular end-user applications mainly related with storage (e.g. Dropbox) or web desktops (e.g. eyeOS), cloud computing can be a solution for problems with high computational demands. These problems could benefit from being processed across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes use the same hardware) or a grid (if the nodes use different hardware) using frameworks such as MapReduce [7] and its popular open source implementation, Hadoop [8]. Some recent works state that Cloud computing can be good alternative to the usage of some popular CBIR libraries such as LIRe [9], which are not designed to deal with Big-Data. In [10] authors take a the problem of finding approximate nearest neighbors for a repository of over one billion images, and perform clustering based on these results. Authors introduce a parallel version of a state of art approximate nearest neighbor search algorithm, known as spill trees. In [11] authors present NIR, an open source cloud computing enabled content based image retrieval system. NIR is based in Nutch, a Cloud-based text search engine framework which in turn based on Hadoop. Another Cloud-based API for CBIR (in this case only for image processing tasks) is HIPI [12], the open-source Hadoop Image Processing Interface (HIPI). This API aims to create an interface for computer vision with MapReduce technology, abstracting the highly technical details of Hadoop. A direct usage of Hadoop for CBIR development is inconvenient, as, for instance, it would involve significant overhead to obtain standard float image representation (assumed to have values in the range [0..1]). Besides, distributing the image dataset to a set of Map nodes would require passing the images as a String, then decode each image in each map task before being able to do access pixel information. HIPI provides its Image Bundle data type as input, and distributes images in the HIPI Image Bundle across all map nodes. The distribution attempts maximize locality between the mapper machines and the machine where the image resides. The HIPI Image Bundle data type stores many images in one large file so that MapReduce jobs can be performed more efficiently. A HIPI Image Bundle consists of two files: a data file containing concatenated images and an index file containing information about the offsets of images in the data file.

IV. AN EXAMPLE MEDICAL CBIR SYSTEM. OPTICAL BIOPSY RETRIEVAL

A. System's architecture overview

We have designed an optical biopsy retrieval system which will allow endoscopists navigating and searching over a CLE image database. Endoscopists will be able to refine their search results by selecting a representative image (the example) and using it to submit a new query based on the selected example. This way an endoscopist can gain diagnostic confidence for situations in which the teleconsultation with a pathologist is not feasible. The overall functionality of the system is divided in the following independent tasks (see Figure 2):

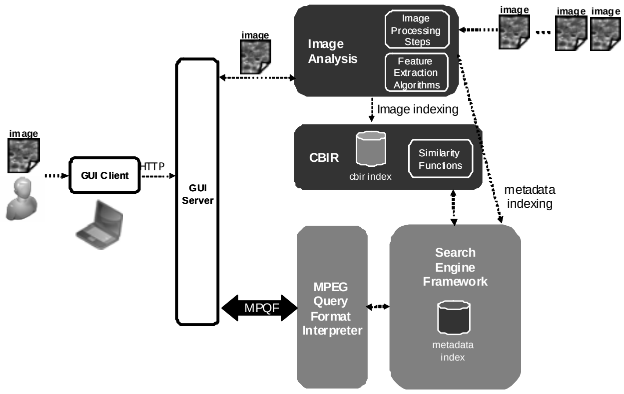

Figure 2. Overall architecture of the Optical Biopsy Retrieval System

- Module I: Image processing and analysis: Applied to the offline extraction of low-level metadata from the images in the database, and also to the on-the-fly extraction of the same metadata from an example image submitted by a user as a query.
- Module II: CBIR index construction: Generation of an index for query-by-example search. Implies the design of feature vectors and also the selection of a similarity function.
- Module III: MPEG Query Format Interpreter: In order to effectively ensure interoperability with potential third-party applications. The system provides a standard interface based on ISO/IEC 15938-12:2008 (MPEG Query Format, MPQF).
- Module IV: Search Engine Framework: General query processor capable of solving text-based queries, CBIR queries and also combinations of both.

B. *Image processing*

In order to extract the necessary low-level features, the OB image must be simplified (preprocessed) and then analyzed. Preprocessing basically consists on image normalization (to minimize light in homogeneities caused by laser light source) that included several steps (enhanced contrast, equalization, etc.) and grey level reduction as seen in Figure 3.

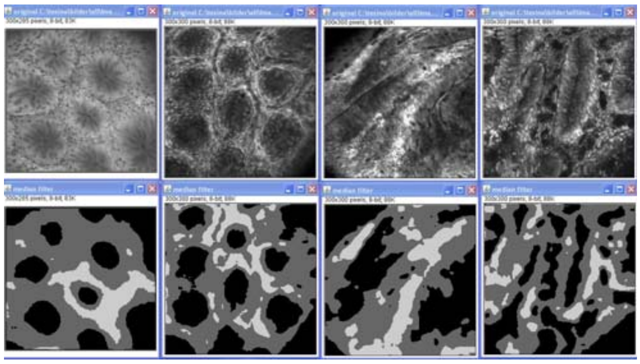

Figure 3. Preprocessing of original images (top). Results in the bottom line

C. *Image segmentation*

Once the image has been simplified, it should be segmented into sets of disjoint pixels. There are many ways of doing this, but here we have applied the following algorithms:

1) The Local Binary Pattern (LBP) [13] operator (A gray-scale invariant texture measure derived from a general definition of texture in a local neighborhood). The process included (a) Integration: On each pixel, we calculated an array of bits of 0 and 1 comparing the original pixel value and its neighbors in a certain radius. (b) Decision maker: The array values are summed up. The higher lbpSum for a pixel indicated more likely to be the center of one of the big black areas (Figure 3).

2) The modified density-based DBSCAN algorithm, highlighted the various crypts and their boundaries.

With the lbpSum value for every pixel, we apply a clustering algorithm to cluster to a certain crypt. In the clustering process we used a modified density-based DBSCAN algorithm originally proposed in [14]. See Figure 4.

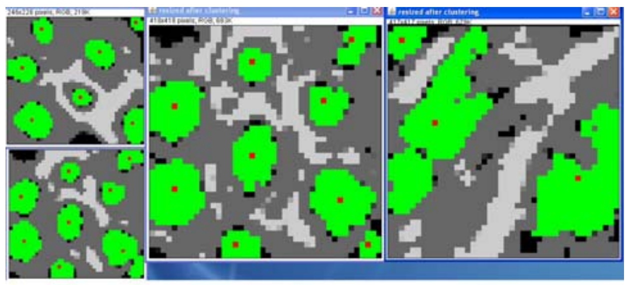

Figure 4. Clustering for gland identification. Normal (left) and hiperplastic glands (right).

Once the image has been segmented we can start the next step, feature extraction. This stem consists on extracting and measure certain features such as the silhouette coefficient, the crypt compactness, the crypt roundness or the inter-crypt distance. Figure 5 shows an example feature vector extracted from a CLE image from a normal tissue.

$$F(image) = \begin{pmatrix} silhouettecoefficient \\ silhouettecoefficientvariance \\ compactness \\ compactnessvariance \\ roundness \\ roundnessvariance \\ bordersizedeviation \\ bordersizedeviationvariance \\ interclusterdistance \\ interclusterdistancevariance \\ chisquare \end{pmatrix}$$

$$= \begin{pmatrix} 0.7571 \\ 0.0048 \\ 1.0861 \\ 0.000272 \\ 0.2690 \\ 0.0142 \\ 0.0462 \\ 0.1718 \\ 4.5227 \\ 0.3272 \\ 0.1002 \end{pmatrix}$$

Figure 5. Example feature vector

D. *Image indexing and retrieval*

Apart from the automatic extraction of low-level metadata, we still need to design a similarity function in order to retrieve similar images to a given one. This similarity function operates over a vector of selected features, whose composition determines which is the nature of the similarity being considered (similarity is relative in a multidimensional space). Before a similarity measure is computed over the feature vector, the vector should be normalized. We have applied linear scaling unit range normalization. We have tested several similarity measures, such as the *euclidian distance*, the *manhattan distance* or the *quadratic-form distance*. The results of the tests showed that the manhattan and the euclidian distance, in combination with the linear scaling unit range normalization, provide the better performance.

E. *MPEG Query Format Interpreter*

In order to effectively ensure the interoperability of the system, or any of its modules, with potential third-party applications, we rely on the usage of a standard query interface based on ISO/IEC 15938-12:2008 (MPEG Query Format or MPQF).

E. *Search Engine Framework*

Once the user information needs have been formalized as an MPQF input query, it is received at the system and processed by the MPQF Interpreter. The MPQF Interpreter translates the query into calls to another pluggable module, the Search Engine Framework, which is responsible of processing the different types of conditions present in a query, i.e. text-based queries, CBIR queries or combinations of both. This design pursues decoupling from the system a functionality which can be implemented by existing information retrieval libraries and databases. The goal is to delegate the evaluation of the query condition tree and the text-based conditions to third-party software which should be also capable to interoperate with our CBIR modules.

In our implementation of the proposed architecture, we have selected Apache Lucene [18] to play the role of the Search Engine Framework. Lucene is a high-performance, full-featured text search engine library developed by The Apache Software Foundation. Since it uses its own optimized index of documents, every CLE image has to be transformed into a Lucene Document and indexed before any search can be conducted. Lucene is essentially a text search engine, it does not natively accept CBIR Queries, but we have been able to extend it with our own CBIR modules while preserving its capability to orchestrate boolean condition trees.

F. *Graphical user interface*

Figure 6 shows a screenshot of the basic view of the system's GUI. The provided interface allows users navigating and searching over the optical biopsy image database. Users are able to retrieve information about precedent diagnostics by providing an example OB image, by using keywords, or by filtering different fields for structured retrieval.

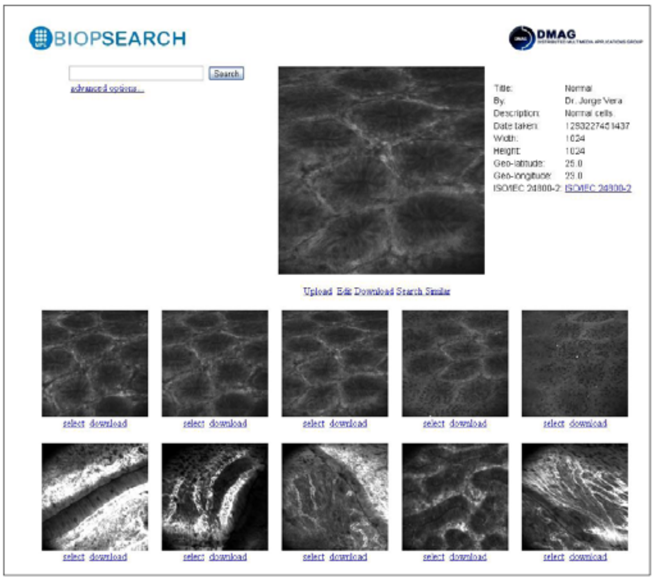

Figure 6. Web interface of the Optical Biopsy Retrieval System

The GUI also allows users performing certain operations over selected OBs, such as cropping, zooming/scrolling, annotating ROIs, highlighting crypts, etc. Figure 7 shows a screenshot of the highlighting crypts view.

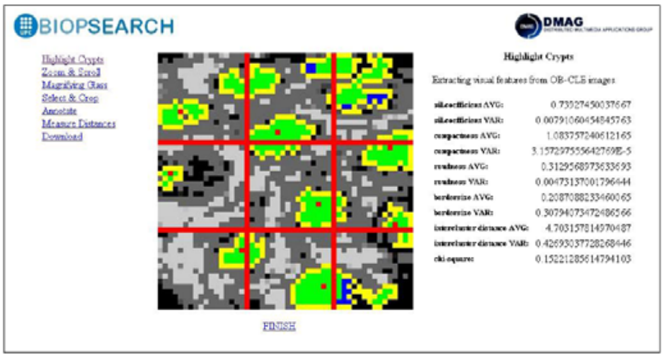

Figure 7. Web interface of the Optical Biopsy Retrieval System

V. CONCLUSION

The present paper provides an introduction to CBIR, its related technologies, standards and its future directions (e.g. Cloud computing). As an example, the paper describes an example CBIR application in the medical domain. An optical biopsy retrieval system based on the query-by-example paradigm and the multimedia standard ISO-15938-12:2008 is described. The system allows users retrieving information about precedent diagnostics by providing an example OB image for content based image retrieval (CBIR), by using keywords, or by filtering different fields for structured retrieval. The usage of the system could speed up medical diagnostic knowledge regarding novel technologies. This is the case of OB that is carried out by clinicians (endoscopists) while is based on microscopic morphology of the tissue, a domain specific of the surgical pathology.

REFERENCES

[1] Content-based Multimedia Information Retrieval: State of the Art and Challenges, Michael Lew, et al., ACM Transactions on Multimedia Computing, Communications, and Applications, pp. 1-19, 2006.

[2] Wang TD, VanDam J.(2004) Optical Biopsy: A New Frontier in Endoscopic Detection and Diagnosis. Clin Gastroenterol Hepatol 2(9): 744–753.

[3] Kiesslich R., Galle P.R., Neurath M.F., 2008. Atlas of endomicroscopy. Springer-Verlag Heidelberg. ISBN 978-3-540-34757-6.

[4] Charles C. Weems (1991) Architectural requirements of image understanding with respect to parallel processing, Proc. IEEE 79:537-547.

[5] ‘‘The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010,’’ white paper, IDC, March 2007.

[6] John R. Smith, “History Made Every Day,” IEEE Multimedia, July-September 2011 (vol. 18 no. 3)

[7] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters”, Communications of the ACM, 51(1): 107-113, Jan. 2008

[8] Hadoop project, http://hadoop.apache.org/

[9] M. Lux and S. A. Chatzichristofis. “LIRe: Lucene Image Retrieval – An Extensible Java CBIR Library”, Proceeding of the 16th ACM International conference on Multimedia, pages 1085-1088 ,2008

[10] Ting Liu, Charles Rosenberg, Henry Rowley, Clustering Billions of Images with Large Scale Nearest Neighbor Search 2007 IEEE Workshop on Applications of Computer Vision WAC V 07 (2007)

[11] Zhuo Yang; Kamata, S.; Ahrary, A.; NIR: Content based image retrieval on cloud computing. IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009. ICIS 2009.

[12] HIPI - Hadoop Image Processing Interface. http://hipi.cs.virginia.edu/

[13] Pietikainen M., Ojala T. and Maenpaa T., 2000. Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In Proceedings of the Sixth European Conference on Computer Vision (ECCV2000), pages 404–420.

[14] Sander, et al., 2000. Density-based Clustering In Spatial Databases: The Algorithm Gdbscan and its Applications. Data Min. Knowl. Discov., 2(2):169–194.

[15] Matthias Gruhne, Ruben Tous, Mario Doeller, Jaime Delgado, and Harald Kosch. MP7QF: An MPEG-7 Query Format. In Proceedings of the 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS 2007), pages 15–18, Barcelona, Spain, November 2007.

[16] ISO/IEC 15938-12:2008. Information Technology - Multimedia Content Description Interface - Part 12: Query Format ISO/IEC 15938-12:2008 MPEG Query Format. Editors: Kyoungro Yoon, Mario Doeller, Matthias Gruhne, Ruben Tous, Masanori Sano, Miran Choi, Tae-Beom Lim, Jongseol James Lee, Hee-Cheol Seo.

[17] Mario Doeller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian S. Burnett (2008). The MPEG Query Format: On the way to unify the access to Multimedia Retrieval Systems. IEEE Multimedia, ISSN: 1070-986X, volume 15, issue 4, 2008.

[18] Apache lucene project, 2010. http://lucene.apache.org (Accessed January 5, 2011).

[19] David G. Lowe, Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91–110. 2004

[20] J. Wang, S. Kumar and S.-F. Chang, Semi-Supervised Hashing for Scalable Image Retrieval, in CVPR, San Francisco, USA, 2010

[21] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In Proc. of 25th VLDB, 1999.

[22] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In Proc. of NIPS, volume 21, pages 1753–1760. 2008.

Ruben Tous received his his Ph.D in Computer Science and Digital Communication from UPF (Universitat Pompeu Fabra, Barcelona, Spain) in 2006. From 2000 to 2001 he worked as a consultant at CapGemini Ernst&Young in Barcelona, From 2001 to 2005 he worked at the Department of Techology of UPF. Since 2006 he is a researcher at DMAG (Distributed Multimedia Applications Group) of the Department of Computer Architecture of UPC (Universitat Politecnica de Catalunya, Barcelona, Spain) and an Associate Professor. He is an expert for the Asociación Española de Normalización y Certificación (AENOR) and has been participating as spanish delegate in ISO/MPEG and ISO/JPEG. His research interests include algorithms and data structures, knowledge representation and reasoning for multimedia understanding, multimedia databases and query languages, multimedia information retrieval and image retrieval in medical applications.