Extended Query Refinement for Content-Based Access to Large Medical Image Databases

Thomas M. Lehmann[1], Bartosz Plodowski, Klaus Spitzer,
Berthold B. Wein [a], Hermann Ney [b], Thomas Seidl [c]

Department of Medical Informatics
[a] Department of Diagnostic Radiology
[b] Chair of Computer Science VI
[c] Chair of Computer Science IX
Aachen University of Technology (RWTH), Aachen, Germany

ABSTRACT

The differentiating characteristics of text versus images and their impact on large medical image databases intended to allow content-based indexing and retrieval have recently been explored. For the design of powerful user interfaces, we propose a grouping of the various mechanisms into four classes: (i) output modules, (ii) parameter modules, (iii) transaction modules, and (iv) process modules, all of which being controlled by a detailed query logging.

Relevance feedback by the physician loops the input, search, and output sequence and is commonly accepted to be most effective for query refinement. Our modular concept provides two additional loops of interaction. Based on the detailed logging of user interaction, an inner loop allows to step back to any previous answer that was given by the system during the interactive session. Boolean linkage of successive queries is provided by an outer loop. Nonetheless, the entire data flow is still controlled within a single web page by means of simple decision rules that are implemented only using push buttons, which can be handled most easily.

Our approach is exemplified by means of an application for content-based access to medical images of similar modality, orientation, and body region using global features that model gray scale, texture, structure and global shape characteristics. The three nested loops for interaction provide a maximum of flexibility within a minimum of complexity. The resulting extended query refinement has a significant impact for content-based image retrieval in medical applications (IRMA).

Keywords: Content-Based Image Retrieval (CBIR), Picture Archiving and Communication Systems (PACS), User Interface, Web-based Interface, Query Refinement, Relevance Feedback, Usability

1. INTRODUCTION

Direct digital imaging techniques are of increasing importance in medical diagnosis. Based on the digital imaging and communications in medicine (DICOM) protocol, large archives hosting medical imagery are nowadays used routinely. However, access to the increasing volume of digital image data is still based on alphanumerical attributes such as patient name, imaging modality, study descriptors, or recording date. Therefore, content-based access to medical images has strong impacts for computer-aided diagnosis, evidence-based medicine, or case-based reasoning [1,2].

For content-based image retrieval (CBIR), the images are represented by a set of numerical features, which may be extracted global, i.e. describing the entire image, or local, i.e. representing only a part of the image. At time of query processing, the user presents an example image (query by example, QBE), which also passes the feature extraction algorithm. Similar images are retrieved from the archive comparing the features by means of a distance or similarity

[1] Corresponding author: Dr. Thomas Lehmann, Department of Medical Informatics, Aachen University of Technology, Pauwelsstr. 30, D - 52057 Aachen, Germany, email: lehmann@computer.org; web: http://irma-project.org/lehmann, phone: +49 241 80-88793.

Medical Imaging 2004: PACS and Imaging Informatics,
edited by Osman M. Ratib, H. K. Huang, Proceedings of SPIE Vol. 5371
(SPIE, Bellingham, WA, 2004) · 1605-7422/04/$15 · doi: 10.1117/12.534964

measure. Although both the features and the distance measures have major impact to recall, i.e. the number of retrieved relevant images divided by the number of retrieved images, and precision, i.e. the number or retrieved relevant images divided by the number of relevant images, it is commonly accepted that relevance feedback is most important for query accuracy [3,4]. After the initial query, the user interactively marks images as "good" or "bad", and the system re-computes the query taking into account the selection of the user. Therefore, relevance feedback may close the semantic gap between the low-level feature extraction by machine and the high-level scene interpretation by humans [5]. A lot of research has been performed on the mechanisms the system evaluate the users feedback [6,7]. For instance, the feedback is used to adjust the parameters for feature extraction [8], the similarity measure [9], or the database organization [10]. Positive vs. positive and negative feedback has been analyzed [7] and short-term vs. long-term learning strategies have been compared [11].

Recent research addresses the problem of user interfacing relevance feedback and query refinement mechanisms [12,13]. Three-dimensional visualization of iconic images and features have been proposed, where relevance feedback is expressed by the user moving the icons into clusters of "expected" or "unexpected" elements [14]. Similarly, Meiers et al. propose a combination of relevance feedback and a hierarchical structure representing the image archive with a three-dimensional visualization of the image maps, which leads to an intuitive browsing environment [15]. However, the usability of such complex graphical user interfaces has not yet been evaluated. Furthermore, relevance feedback does not guarantee the improvement of accuracy. Müller et al. claim that too much negative feedback may destroy a query as good features gets negative weightings [7]. Zhu et al. point out that the impact of relevance feedback is limited. In particular, relevance feedback does not work very well when the user wants to express an OR-relationship among the queries [16].

So far, only a few implementations of CBIR-systems are available via the Internet. However, such approaches are restricted to a simple QBE, where the user interactively selects the query image, and similar images are retrieved in some way automatically from the database. Nonetheless, this basic mechanism is insufficient to support medical queries, because information in medical images is regional and geometric [1]. Therefore, advanced concepts for user interfacing content-based image access are required, especially if CBIR is applied in the medical domain [17]. In this paper, we present a static layout that is capable to host extended mechanisms for query refinement. These mechanisms enable not only relevance feedback but also UNDO and REDO functions. Beside these single steps, a complete HISTORY and BACK functionality is offered and combined with AND and OR relationships of intermediate query results, all integrated into a user friendly interface that is intuitive and easy to use.

2. METHODS

Effective access to medical imagery must allow the physician to select local patterns and their relations. In general, this is not possible with a simple query but requires loops, and, consequently, some kind of loop control of the interactive retrieval dialogue. Our approach is composed of (i) systematically collecting the required interface functionality, which a CBIR system must offer, (ii) grouping these mechanisms into the smallest number of similar classes of modules, (iii) connecting these modules in such a way that module interaction is composed into a simple session flowchart, and (iv) structuring these modules into a GUI that is intuitively guiding the user through his session.

2.1. Interface Functionality

Our content-based approach to image retrieval in medical applications (IRMA)[2] is not focused on a certain modality, body region and/or pathology [18]. Instead of, the rather general framework of IRMA is applied to various applications in case-based reasoning, evidence-based medicine, or computer-aided diagnosis. For each application, a certain GUI is composed and connected to the IRMA core hosting the database as well as the programs for feature extraction and comparison. However, several mechanisms are of major importance in every retrieval interfaces.

To initialize a query, the physician must specify a certain image category, modality, or body region examined, present a query image to the system or just draw a sketch, annotate a region of interest (ROI) within an image, select a certain distance measure or value range of parameters to describe the images, and finally submit the query. The visualization of retrieval results includes an overview by displaying thumbnails, but must also provide magnified views of selected

[2] http://irma-project.org

images. Furthermore, the system must reason why a certain image answers the query. This mechanism is referred to as relevance facts. The relevance facts are very important for the user when he attempts to specify his query more precisely. Since the principle of query refinement has proven to be most effective for CBIR, the interface must enable the user to judge the answered images as "relevant" or "irrelevant" and resubmit the query. In general, all functionality required for initialization of a query must also be offered for relevance feedback. Frequently, the interactive process of query refinement does not result in the expected improvement of recall and/or precision. Therefore, the interface must offer essential UNDO and REDO functionality. Moreover, direct access to any result obtained previously during the interactive session should be possible. This kind of history access multiples the impact of medical CBIR, in particular, if the physician can save and label intermediate results by means of general HISTORY functions. With respect to the complexity of information hosted in medical imagery, extended mechanisms are also required for combining system answers. For instance, the intersection of image sets from separated queries might significantly improve the precision while the union of sets might impact the system's recall. In other words, the GUI must provide mechanisms to compute AND and OR relationships.

In summary, a GUI for image retrieval in medical applications must integrate functionality for

- initializing a query,
- visualizing of the answered images and their relevance facts,
- evaluating the answered images and resubmitting a query,
- accessing previously obtained results, and
- combining intermediate results by means of Boolean relationships.

2.2. Interface Modules

The mechanisms described above can be grouped into classes of modules. Firstly, the so called *output modules* contain all functionality regarding the visualization of information such as images, descriptions, or numerical parameters. In particular, "text", "image", "line", "shadow", "frame", and "table" are basic output modules. These modules are used to initialize the query and to visualize the query result. Furthermore, output modules are applied to display the relevance facts.

The second class of modules are called *parameter modules*, which allow the user to interact with the system. "Input field", "radio button", "slider", or "selection box" are some prominent examples. Furthermore, tools for selecting a ROI within an image or drawing a sketch belong to this class of modules. The parameter modules are used whenever a query is initialized or (re-)submitted. In particular, they support relevance feedback.

Frequently used combinations of output and parameter modules are coupled into so called *function modules*. For instance, the output modules "image", "frame" and "shadow" are merged with the parameter module "push button" into the function module "standard thumbnail", which opens a new window displaying the image in full resolution whenever the mouse button is pressed over the image icon. These function modules are used to compose specialized GUIs for certain IRMA applications.

Query logging is used to completely track the user's interaction with the system. This mechanism requires access to a database management system. Every action that is performed by the user is stored with a corresponding session identifier (ID), a unique timestamp and the IDs of all images answered within their current order and respective relevance facts. Therefore, query logging is fundamental for further classes of modules.

Transaction modules collect all mechanisms allowing the user to step back and forward (UNDO or REDO functions) and restore any steady state the system has been before during the current and also during already closed sessions (HISTORY functions). With respect to the web-based GUIs of the IRMA system, all output, parameter, and function modules are embedded in a hypertext form sheet, and the data transfer between the browser and the web server is captured and stored in the database of the IRMA core. Previous stages are easily restored by "updating" the hypertext form with information from older stages.

To provide the ability to union or cut intermediate sets of query results, which already have been looped for query refinement, *process modules* are defined. The image lists and all corresponding relevant facts are stored in the database of the IRMA core, when process modules are activated during a session. Again, these class of modules is certainly based on the query logging and requires an individual user authorization. As a result, Boolean AND and/or OR operations offer a variety of potentials for advanced query refinement to all IRMA applications.

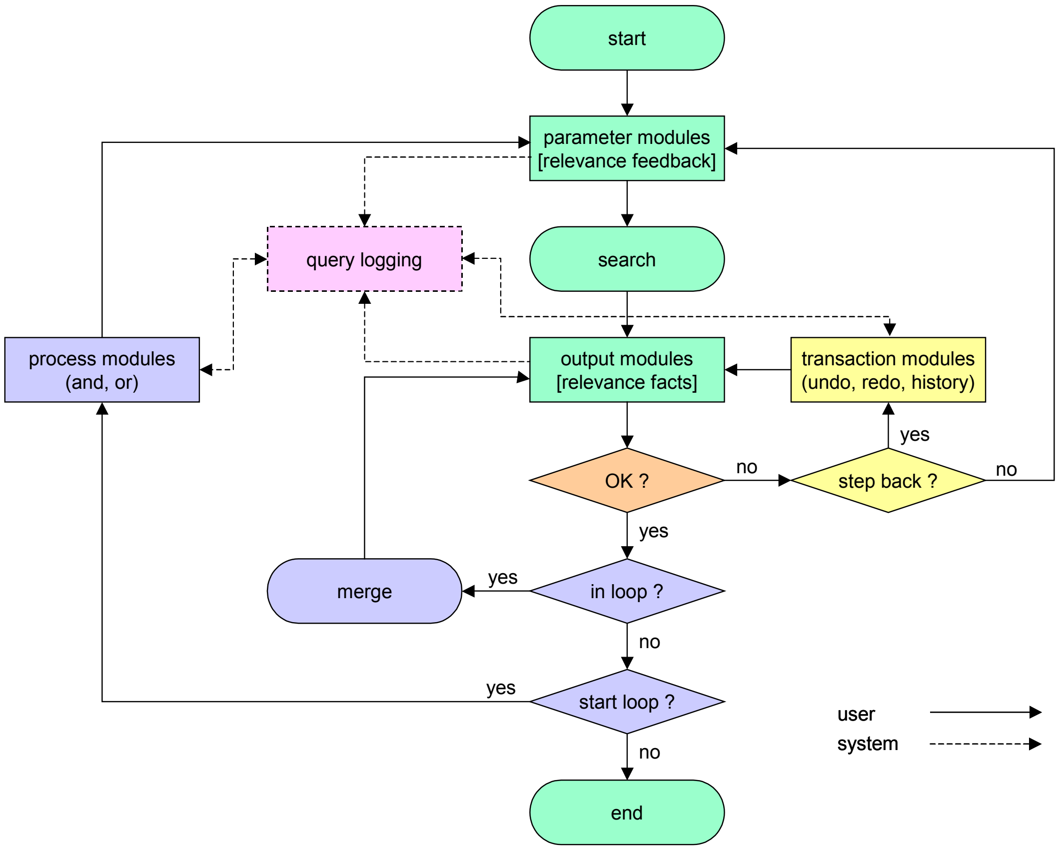

Figure 1: Flowchart of extended query refinement

2.3. Module Interaction and Session Flowchart

Based on this classes of modules, the flowchart of a retrieval session is straightforwardly developed. Figure 1 visualizes the resulting structure. At first, parameter or function modules are used for the initialization of the query. After performing the search, which compares the query with all images in the system by means of the selected abstract features, resulting images are displayed and relevance facts are given by means of the output or function modules. In Figure 1, this basic functionality of any CBIR engine is displayed in green. Not providing relevance feedback, the session flow is straight linear ("rigid" [6]) without any loops for interactive query refinement.

If the system offers relevance feedback, a central loop from output to parameter modules is added (Fig.1, displayed in brown). The sequence of performing relevance feedback, computing a revised search, and presenting relevant facts of updated results is repeated until the user accepts the current response to his query.

The transaction modules add a second loop to the flowchart intra-connecting the output models (Fig. 1, displayed in yellow). Note that this loop is supported by the system's query logging (Fig. 1, displayed in red), which also reads from the parameter and the output modules. Since the query logging is connected to the database of the IRMA core, any previous search result can be addressed, reloaded and restored from the database.

Finally, a third loop is added to provide extended query refinement by combining successive but independent queries based on the same database (Fig. 1, displayed in blue). If the system is currently not within a Boolean loop, such a loop can be opened to extend (OR relationship) or refine (AND relationship) the resulting image list, which is stored as pending in the database via the query logging mechanisms. The Boolean loop is closed automatically by merging the pending with the current response list of images if the user accepts the current result. Again, this loop can be iterated until the user agrees to the final result. If Q and i denote the resulting image list and the number of iterations counted over all loops, expressions such as

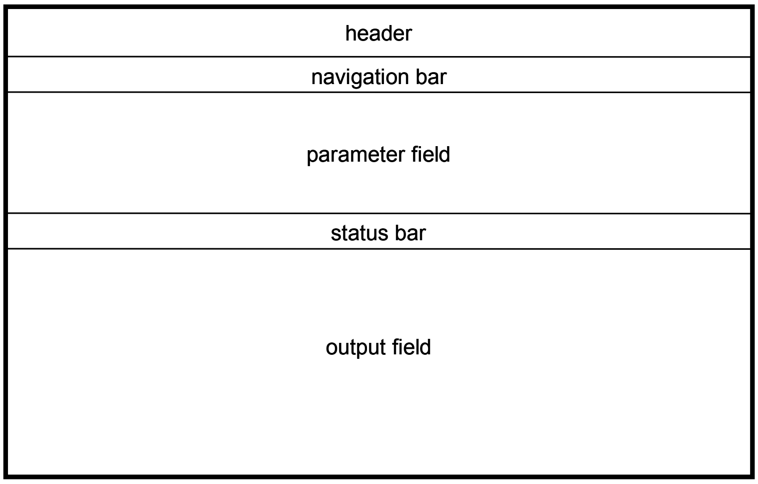

Figure 2: General GUI layout

$$Q_{i+1} = (((Q_{i-2}) \cup / \cap Q_{i-1}) \cup / \cap Q_i) \tag{1}$$

can be evaluated directly by means of the outer Boolean loop (Fig. 1, displayed in blue). Note however, that more complex expressions like

$$Q_{i+1} = (Q_{i-2} \cup / \cap Q_{i-1}) \cup / \cap (Q_{i-4} \cup / \cap Q_{i-3}) \tag{2}$$

can also be computed if the user combines process and transaction loops. For instance in equation (2), the history function of the process modules is called within the Boolean loop *i*, to restore a previous result, which itself was obtained from Boolean looping. Using this principles, arbitrary expressions can be used to combine intermediate results.

2.4. Interface Layout and Usability

In total, four loops are contained in the flowchart of the IRMA extended query refinement approach (Fig. 1). With respect to the system's usability, it is important to present the current state of the system as simple and intuitive as possible. In fact, all interaction can be integrated within a single web screen (Fig. 2). A header denotes the name of the GUI that is currently displayed. The navigation bar contains all buttons required to navigate a session. This includes push buttons to initialize and to submit a query as well as to loop the session. Within the parameter field, function modules are used to enable the user to parameterize the system and the query. The status bar announces system information and may provide push buttons to navigate the results, which are displayed in the output field.

Note that all web GUIs required for CBIR in medical applications can be composed of this five basic components, all of which being optional in each of the GUI. For instance, a full resolution view of an image, which is initiated by hitting the icon of its thumbnail module that is displayed in the output field, may open a separate GUI window that is composed only of a header and an output field. In all of the GUIs, available options are denoted depending o the current position in the work flow (Fig. 1) by appropriate shading of the push buttons for navigation. For instance, the buttons of those options being not available in the current state are shaded gray. This enables a static composition of all modules in the parameter and output fields of the GUIs. For Boolean loops, colored buttons are used to indicate the current state of the system.

3. RESULTS

The approach for extended query refinement was implemented within the IRMA framework [17,18]. The IRMA system is composed of a central relational database storing images, features, and feature extraction methods. This IRMA core can be DICOM-connected to a PACS image archive. In order to be platform independent and broadly available, IRMA applications are interfaced by common web browser technology. The browser communicates with the IRMA web server by means of the hypertext transfer protocol (HTTP). The IRMA web server operates the hypertext preprocessor (PHP)

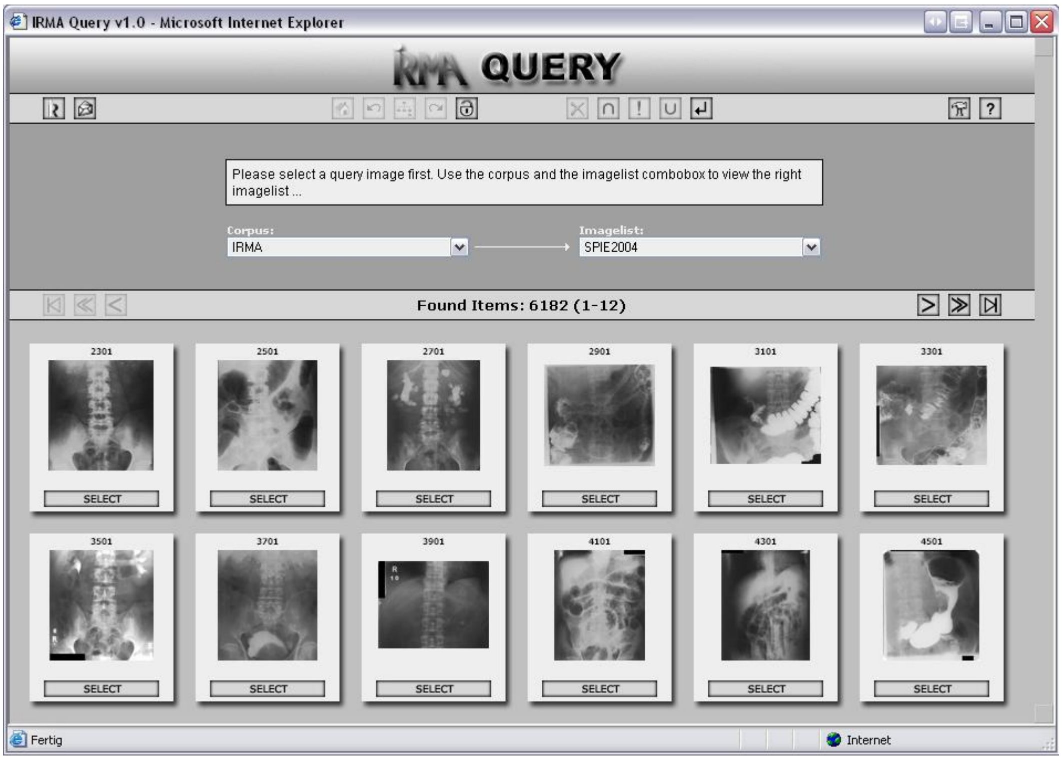

Figure 3: Initialization stage of a web-based CBIR GUI

to generate dynamic web pages. It is connected to the database (PostgreSQL) by means of the standard query language (SQL). This link is also used to store the logging information from the user's session, which is coded in the extensible markup language (XML).

Figures 3 and 4 show screenshots of an IRMA front-end for content-based access to radiographs by means of global texture, structure, and shape analysis. Output, parameter, and function modules are placed in the center fields (parameter field, status bar, output field in Fig. 2) of the browser while transaction and process modules are represented by the push buttons within the navigation bar below the top banner (header in Fig. 2). Here, the transaction commands HOME, UNDO, HISTORY, REDO, and EXIT are coupled in a first group while the process commands RESET, AND, NOT, OR, and SUBMIT form the second group. Note that three loops of interaction are nested within the static layout of a single web page merging a maximum of flexibility with a minimum of complexity. Depending on the current position in the flow chart, some buttons may marked as unavailable. For instance in Figure 3, neither transaction modules (except the EXIT button) nor process modules (except the SUBMIT button) are available.

Figure 3 displays the initialization stage of a content-based query. In the parameter field, the output module "text" is used to display user instructions and the parameter modules "selection box" are used to offer the available data corpora and image lists. After receiving the user's selection, the number of images found is displayed in the status, which also offers push buttons to scroll the list of returned images. Several function modules "select image" are positioned in the output field to display the query result. The number of icons can be preset by the user. Since a standard module is called, a separate window is always opened whenever the left mouse button is hit over the image icon.

After the user hits the SELECT button below an icon, status bar and output field are cleared and the user can select the desired experiment (features and distance measure). After submitting, the first query result is presented (Fig. 4). The function module "evaluate image" is now used in the output field. Moving the slide bar below the image, positive and negative relevance feedback can be provided by the user. Using the function module "evaluate image", not only "good" or "bad" can be indicated. The distance by which the slider is shifted from its origin to the left or right denotes the disagreement or agreement, respectively, of the user with this response. In this stage, most transactions and process modules are available. For instance, the user may step BACK to the initialization (Fig. 3) or call the HISTORY function.

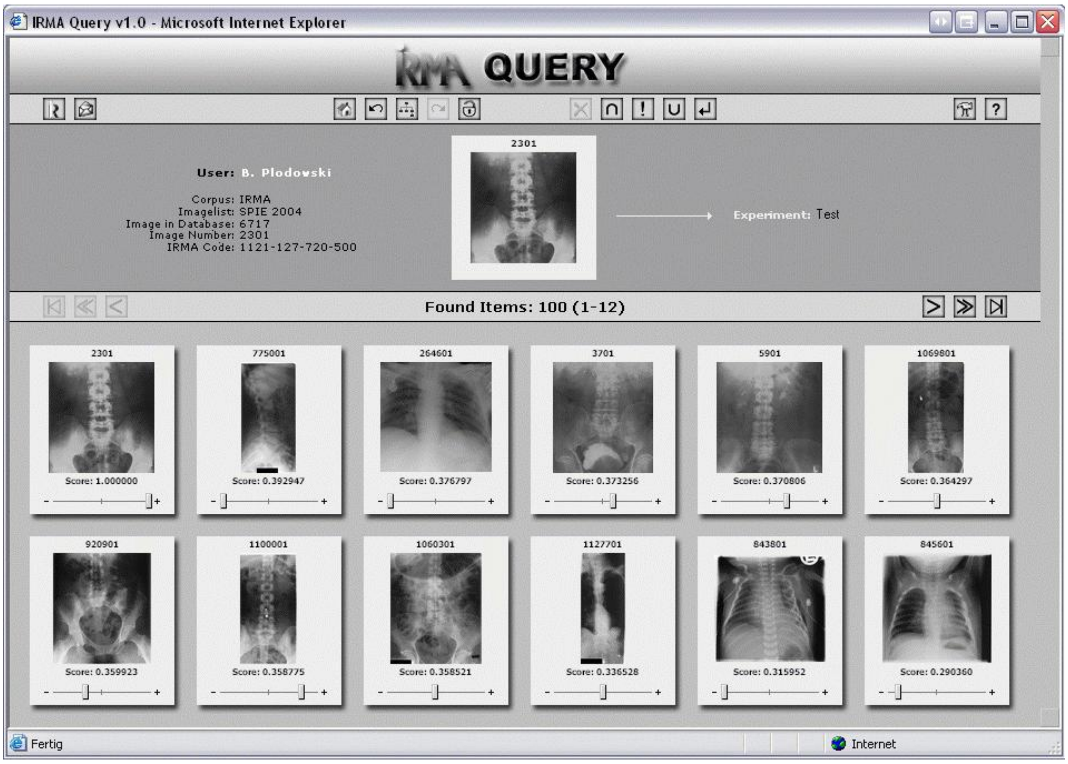

Figure 4: Query refinement stage of a web-based CBIR GUI

Figure 5 shows a screenshot of the history selection GUI, which is opened via the HISTORY button in Fig. 4. The history GUI is composed of the header, the parameter and the output field. Neither the navigation nor the status bar are displayed. The parameter field offers push buttons to resize the log tree, which can become quite large for complex sessions, and to label specific nodes. Note that this concept offers access to the complete history of user interaction. In most other programs (e.g. the Microsoft Office Package), only the direct path to the current node is maintained by the back and forward stepping. For instance, the path to node 4 will be immediately erased when stepping back to node 3 and then going to node 5. considering the current node 11 (Fig. 5, marked red), a linear path will offer only the nodes 0, 1, 2, 6, and 11, cutting most of the history information.

4. DISCUSSION

Since it provides loops during system interaction, relevance feedback is recognized as one of the most important mechanisms for CBIR user interfaces [4,5,6]. However, it is usually restricted to label some of the answered images as “relevant” or “irrelevant”. Using our parameter modules, a continuous ranking is possible.

More important, a second loop is offered by database-assisted query logging, which enables the user to step back and forward or to select any previous system response. Particularly for interactive CBIR systems, this functionality is fundamental for usability, since query refinement and relevance feedback must not necessarily improve the accuracy of the system response.

A third loop provides Boolean operations to image sets obtained from successive queries, which itself can be looped by query refinement. Zhu et al. have already stressed the impact of Boolean relationships among the queries [14], but – to the best of our knowledge – simple user interfaces supporting such kind of extended query refinement have not yet been published. Nonetheless, the data flow is still controlled by means of simple decision rules that are implemented in a static interface using only push buttons. Since our approach refrains from complex menus hosted in several web pages, it can be handled most easily and can be adopted to a plenty of applications in radiology and medicine.

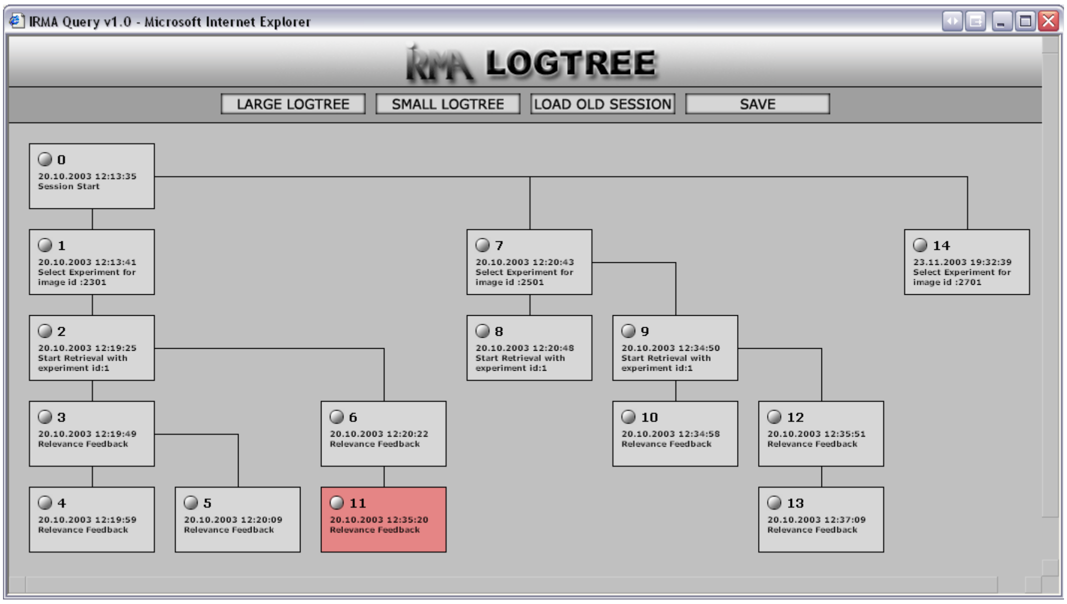

Figure 5: Screenshot of a web-based GUI to select and restore previous system stages

5. CONCLUSION

CBIR is an emerging field of research in particular for picture archiving and communication systems (PACS). However, the usefulness of existing approaches is limited since two distinct characteristics of medical CBIR systems are relatively ignored: the gap between high level concepts and low level features; and subjectivity of human perception of visual content. The extended query refinement proposed in this paper allows to capture the physician's high level query and perception subjectivity by dynamically updated image features based on relevance feedback as well as query combinations. Consequently, CBIR approaches become available for routine use in medicine.

ACKNOWLEDGEMENT

This work was performed within the image retrieval in medical applications (IRMA) project, which is supported by the German Research Community (Deutsche Forschungsgemeinschaft, DFG), grants Le 1108/4 and Le 1108/6. For further information visit http://irma-project.org.

REFERENCES

1. Tagare HD, Jaffe CC, Dungan J: Medical image databases – A content-based retrieval approach. Journal of the American Medical Informatics Association JAMIA 4: 184-198, 1997.
2. Müller H, Michoux N, Bandon D, Geissbuhler A: A review of content-based image retrieval systems in medical applications – Clinical benefits and future directions. International Journal of Medical Informatics, 2004, in press.
3. Celetyno A, Di Sciascio E: Feature integration and relevance feedback analysis in image similarity evaluation. Journal of Electronic Imaging 7(2): 308-317, 1998.
4. Rui Y, Huang TS, Ortega M, Mehrotra S: Relecance feedback – A power tool for interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 8(5): 644-655, 1998.
5. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 22(12): 1349-1380, 2000.

6. Naster C, Mitschke M, Meilhac C: Efficient query refinement for image retrieval. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 547-552, 1998.
7. Müller H, Müller W, Squire DM, Marchand-Maillet S, Pun T: Strategies for positive and negative relevance feedback in image retrieval. Proceedings 15th International Conference on Pattern Recognition 1043-1046, 2000.
8. Bartolini I, Ciaccia P, Waas F: FeedbackBypass – A new approach to interactive similarity query processing. Proceedings 27th International Conference on Very Large Data Bases, 201-210, 2001.
9. Laaksonen J, Koskela M, Laakso S, Oja E: Self-organizing maps as a relevance feedback technique in content-based image retrieval. Pattern Analysis and Applications 4(2-3): 140-152, 2001.
10. Chen JY, Bouman CA, Dalton JC: Active browsing using similarity pyramids. Proceedings SPIE 3656: 144-154, 1998.
11. He XF, King O, Ma WY, Li MJ, Zhang HJ: Learning a semantic space from relevance feedback for image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 13(1): 39-48, 2003.
12. Crestani F, De la Fuente P, Vegas J: Design of a graphical user interface for focussed retrieval of structured documents. Proceedings 8th Symposium on String Processing and Information Retrieval, 246-249, 2001.
13. Vegas J, De la Fuente P, Crestani F: A graphical user interface for structured document retrieval. Lecture Notes in Computer Science 2291: 268-283, 2002.
14. Zhu XQ, Zhang HJ, Liu WY, Hu CH, Wu L: New query refinement and semantics integrated image retrieval system with semiautomatic annotation scheme. Journal of Electronic Imaging 10(4): 850-860, 2001.
15. Meiers T, Sikora T, Keller I: Hierachical image database browsing environment with embedded relevance feedback Proceedings 2002 International Converence on Image Processing 2: 593-596, 2002.
16. Fox KL, Frieder O, Knepper MM, Snowberg EJ: SENTINEL – A multiple engine information retrieval and visualization system. Journal of the American Society for Information Science 50(7): 616-625, 1999.
17. Lehmann TM, Güld MO, Thies C, Fischer B, Keysers D, Kohnen M, Schubert H, Wein BB: Content-based image retrieval in medical applications for picture archiving and communication systems. Proceedings SPIE 5033: 109-117, 2003.
18. Lehmann TM, Wein B, Dahmen J, Bredno J, Vogelsang F, Kohnen M: Content-based image retrieval in medical applications – A novel multi-step approach. Proceedings SPIE 3972: 312-320, 2000.