

Extended Query Refinement for Medical Image Retrieval

Thomas M. Deserno,¹ Mark O. Gild,¹ Bartosz Plodowski,¹ Klaus Spitzer,¹ Berthold B. Wein,²
Henning Schubert,² Hermann Ney,³ and Thomas Seidl⁴

The impact of image pattern recognition on accessing large databases of medical images has recently been explored, and content-based image retrieval (CBIR) in medical applications (IRMA) is researched. At the present, however, the impact of image retrieval on diagnosis is limited, and practical applications are scarce. One reason is the lack of suitable mechanisms for query refinement, in particular, the ability to (1) restore previous session states, (2) combine individual queries by Boolean operators, and (3) provide continuous-valued query refinement. This paper presents a powerful user interface for CBIR that provides all three mechanisms for extended query refinement. The various mechanisms of man-machine interaction during a retrieval session are grouped into four classes: (1) output modules, (2) parameter modules, (3) transaction modules, and (4) process modules, all of which are controlled by a detailed query logging. The query logging is linked to a relational database. Nested loops for interaction provide a maximum of flexibility within a minimum of complexity, as the entire data flow is still controlled within a single Web page. Our approach is implemented to support various modalities, orientations, and body regions using global features that model gray scale, texture, structure, and global shape characteristics. The resulting extended query refinement has a significant impact for medical CBIR applications.

KEY WORDS: Graphical user interface (GUI), web-based interface, query refinement, relevance feedback, usability

BACKGROUND

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 alphanumeric attributes such as patient name, imaging modality, study descriptors, or the re-

coding date. Therefore, content-based access to medical images has strong impact on 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,³⁻⁵ which are either extracted globally, i. e., describing the entire image, or locally, 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 is also processed with the feature extraction algorithms. Similar images are retrieved from the archive by comparing the query and database features by means of a distance or similarity measure. Although both the features and the distance measures have major impact on recall, i. e., the number of retrieved relevant images divided by the total number of relevant images,⁶ and precision, i. e., the number of retrieved relevant images divided by the total number of retrieved images,⁶ it is commonly accepted that relevance feedback

¹From the Department of Medical Informatics, Aachen University of Technology (RWTH), Pauwelsstr. 30, 52057, Aachen, Germany.

²From the Department of Diagnostic Radiology, Aachen University of Technology (RWTH), 52057, Aachen, Germany.

³From the Department of Computer Science 6, Aachen University of Technology (RWTH), 52056, Aachen, Germany.

⁴From the Department of Computer Science 9, Aachen University of Technology (RWTH), 52056, Aachen, Germany.

Correspondence to: Thomas M. Deserno, Ph. D., Department of Medical Informatics, Aachen University of Technology (RWTH), Pauwels str. 30, 52057, Aachen, Germany, tel: +49-241-8088793; fax: +49-241-803388793; e-mail: deserno@ieee.org

Copyright © 2007 by Society for Imaging Informatics in Medicine

Online Publication 12 May 2007

doi: 10.1007/s10278-007-9037-4

is most important for query accuracy.^{7,8} After the initial query, the user interactively marks images as “good” or “bad”, and the system recomputes the query taking into account these selections of the user. Therefore, relevance feedback may narrow the semantic gap between the low-level feature extraction by machine and the high-level scene interpretation by humans.⁹

A lot of research has been performed on how to evaluate user feedback.^{10,11} For instance, the feedback may be used to adjust the parameters for feature extraction,¹² the similarity measure,¹³ or the database organization.¹⁴ Positive vs positive and negative feedback has been analyzed,¹¹ and short-term vs long-term learning strategies have been compared.¹⁵ Recent research addresses the problem of designing user interfaces for relevance feedback and query refinement mechanisms.^{16,17} 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.¹⁸ 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.¹⁹ However, the usability of such complex graphical user interfaces (GUI) 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 get negative weightings.¹¹ Zhu et al. pointed out that the impact of relevance feedback is limited in general. In particular, relevance feedback does not work very well when the user wants to express an OR-relationship among the queries.²⁰

We will emphasize these limitations within a medical use-case scenario: suppose a radiologist is reading a chest X-ray. As he detects irregular nodules in the upper part of the left and the lower part of the right pulmonary lobes, he aims at consulting other cases with similar patterns in the radiograph. Therefore, he marks a region of interest (ROI) and submits the request to the picture archiving and communication system (PACS). Using continuous-valued relevance feedback, he refines the query until the returned pattern matches the request. The same iteration is performed for the other ROI. Thereafter, the

two resulting sets of chest X-rays are merged by Boolean AND. The result is displayed to the physician, who selects the best matching images and accesses the corresponding patient records.

This use-case scenario emphasizes that advanced GUI concepts for content-based image access are required, especially if CBIR is applied in the medical domain.²¹ More precisely, the user must be able

- To revisit previous session states, as this is a remarkable improvement to simple UNDO and REDO
- To provide continuous-valued relevance feedback, rather than just binary judgments of “relevant” and “not relevant”
- To combine queries with complex assemblies of Boolean operators

In this paper, we present a user-friendly, static GUI layout that is capable of supporting such extended mechanisms for query refinement. It is part of our content-based approach to image retrieval in medical applications (IRMA, <http://irma-project.org>). The IRMA approach is not focused on a certain modality, body region and/or pathology.²¹ Instead, various applications in case-based reasoning, evidence-based medicine, or computer-aided diagnosis can be instantiated within the general framework of IRMA. The IRMA system is composed of a central relational database storing images, features, and feature extraction methods.²² This IRMA core can be DICOM-connected to a PACS image archive. IRMA applications are interfaced by common Web browser technology, and the particular IRMA GUI that is required for a certain application is composed of standardized GUI modules. 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) 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). Whereas the concept of extended query refinement has been proposed before,²³ this paper is based on an actual implementation. Currently, a demo system is available at http://www.irma-project.org/onlinedemos_en.php.

METHODS

Effective access to medical imagery must allow the physician to select local patterns and their relations. In general, this is not supported by a simple query but requires “loops” of interaction, i. e., a repeated interactive performance of the same sequence of manual action, and, consequently, some kind of loop control of the retrieval dialogue. Our approach is composed of (1) systematically identifying the required interface functionality that a CBIR system must offer, (2) grouping these mechanisms into the smallest number of similar classes of modules, (3) connecting these modules in such a way that module interaction is composed into a simple session flowchart, and (4) structuring these modules into a GUI that intuitively guides the user through his session.

Interface Functionality

To initialize a query, the physician must specify a certain image category, modality, or body region examined, and present a query image to the system or just draw a sketch, annotate a 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 not only includes an overview by displaying thumbnails, but also permits viewing of selected images in their original full size. Furthermore, the system provides a rationale that attempts to justify why particular images satisfy the query. This mechanism is described as providing relevance facts. The relevance facts are important for the user in attempting to refine the query, i. e., to specify his query more precisely.

The paradigm of query refinement and relevance feedback has proven to be most effective for CBIR. All functionality required for initialization of a query is also offered for relevance feedback. As the interactive process of query refinement often does not result in the expected improvement of recall and/or precision, our interface provides UNDO and REDO functionality. Moreover, direct access to any result obtained previously during the interactive session is provided. This kind of history access significantly increases the impact of medical CBIR, in particular,

because the physician can save and label intermediate results by means of general HISTORY functions.^{24,25} 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.²⁰ To satisfy this potential need, our GUI provides mechanisms to compute AND and OR operations.

In summary, our GUI for image retrieval in medical applications integrates functionality for

- Initializing a query
- Visualizing of the answered images and their relevance facts
- Evaluating the responded images and resubmitting a query
- Accessing previously obtained results
- Combining intermediate results by means of Boolean operators

Interface Modules

Query logging is used to completely track the user’s interaction with the system. It requires individual user authorization as well as access to a database management system. Every action that is performed by the user is stored with a corresponding session identifier (ID), a timestamp, and the IDs of all images returned as query results, along with their current relevance rankings and respective relevance facts. As presented in our previous work, all input and output mechanisms are grouped into four classes of modules:²³

1. *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, to visualize the query result, and to display the relevance facts.
2. *Parameter Modules* 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.

3. *Transaction Modules* include all functions that allow the user to step back and forward (UNDO or REDO functions) and to restore any “steady state” that the system has had within the current session, and also within already-closed sessions (HISTORY functions). With respect to the Web-based GUIs of the IRMA system, all output and parameter 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.
4. *Process Modules* are defined as union or intersection of intermediate sets of query results, which have already been refined by relevance feedback loops. 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. Boolean AND/OR operations are available to operate on these lists for advanced query refinement.

Module Interaction and Session Flowchart

Figure 1 diagrams the flow of a retrieval session, based on these module classes. 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 some CBIR engines is displayed in green. If there were no relevance feedback, the session flow would be linear (“rigid”¹⁰), without any loops for interactive query refinement. If the system offers relevance feedback, a central loop from output to

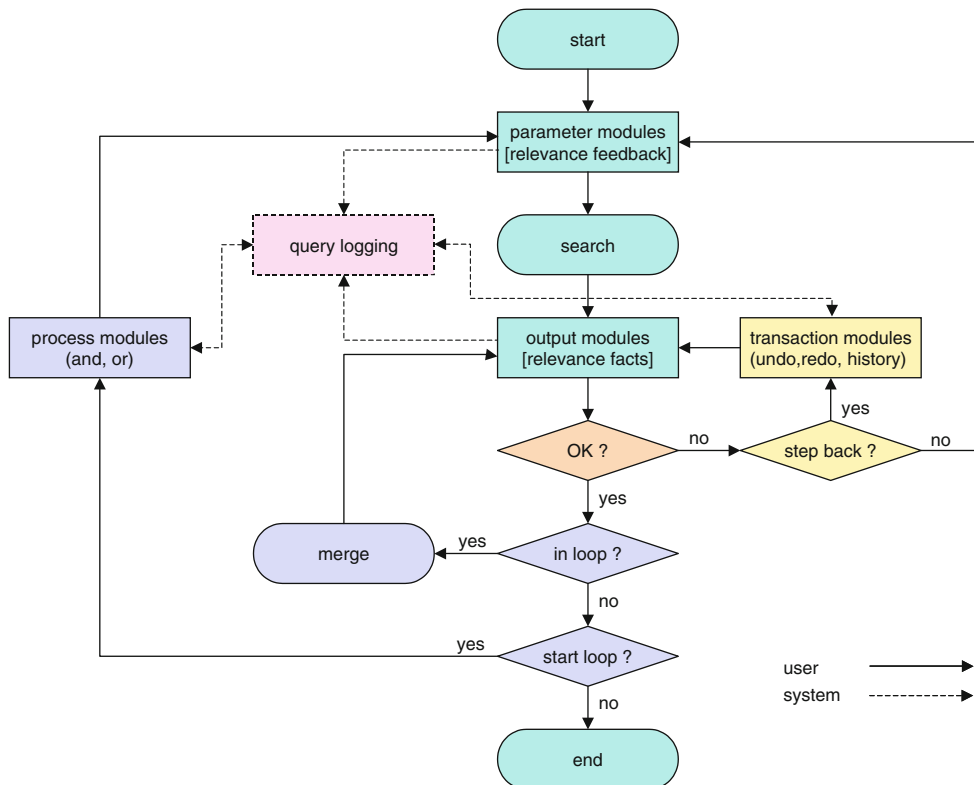


Fig. 1. Flowchart of the extended query refinement.

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 inner loop to the flowchart intraconnecting the output modules (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. As the query logging is connected to the database of the IRMA core, any previous search result can be addressed and restored from the database.

Finally, a third loop (outer loop) is added to provide extended query refinement by combining successive but independent queries over 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, respectively, expressions such as

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

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

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

can also be computed if the user combines process and transaction loops. For instance, in Eq. 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 these principles, arbitrary expressions can be evaluated by combining intermediate results.

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 that is composed of five elements (Fig. 2):

1. A header denotes the name of the GUI that is currently displayed.
2. The navigation bar contains all buttons required to navigate the session. This includes

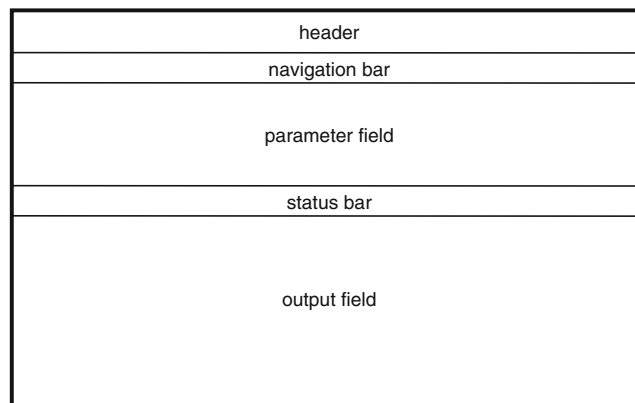


Fig. 2. General GUI layout.

- buttons to initialize and to submit a query as well as to provide relevance feedback.
3. Within the parameter field, function modules are used to enable the user to parameterize the system and the query.
 4. The status bar provides system information and may provide buttons to navigate the results, which are displayed in the output field.
 5. The output field is used to display the query results.

All Web GUIs required for CBIR in medical applications can be composed of one or more of these basic components. For instance, a full resolution view of an image, which is initiated by left-clicking on 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 on the current position in the workflow (Fig. 1), by appropriate shading of the push buttons for navigation. In particular, the buttons of those options that are 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.

RESULTS

The approach for extended query refinement was implemented within the IRMA framework.²¹ An online demonstration for medical image retrieval based on this framework is provided on the Internet (http://www.irma-project.org/online_demos_en.php). Medical CBIR can be performed in a demonstration mode using the images within the IRMA system, or in a local mode where the user can upload a query image. Extended query refinement is provided in both modes.

This demo is based on a set of 10,000 radiographs from clinical routine. Each image is represented by a down-scaled icon of 16×16 pixels (feature vectors of 256 components), and a correlation-based similarity measure is used. Without any indexing, a dual Xenon 2.8 GHz

machine with 2 GB of RAM takes about 9 s and 1.5 s response time before and after caching, respectively. The processing time grows linearly with the number of query images contained in the feedback, as the similarities are computed for each query image.

A screenshot of the GUI after submitting a query image is shown in Figure 3. Below the header, the navigation bar provides the transaction commands HOME, UNDO, HISTORY, and REDO in a first group and the process commands RESET, AND, NOT, OR, and SUBMIT in a second group. Note that three loops of interaction are nested within the static layout of a single Web page, combining a maximum of flexibility with a minimum of complexity. Depending on the current position in the flow chart, some buttons may appear as unavailable, e.g., the REDO button in Figure 3. The query image is shown in the parameter field. It was located using Google image search for “X-ray skull”, and obtained from the University of Illinois at Chicago, College of Medicine, Department of Radiology (http://www.uic.edu/com/uhrd/images/Simpson_head_Xray.jpg). Below the status bar, positive and negative relevance feedback can be provided by the user. 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. The result of a query refinement is shown in Figure 4.

Figure 5 shows a screenshot of the history selection GUI, which is opened via the HISTORY button. The history GUI is composed of the header, the parameter, and the output field. Neither the navigation nor the status bar is 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 providing back and forward stepping. For instance, the path to node 4 will be erased immediately when stepping back to node 3 and then going to node 5. If the “current node” were node 11, (Fig. 5, marked red), a linear path would offer only the nodes 0, 1, 2, 6, and 11, leaving most of the history information inaccessible.

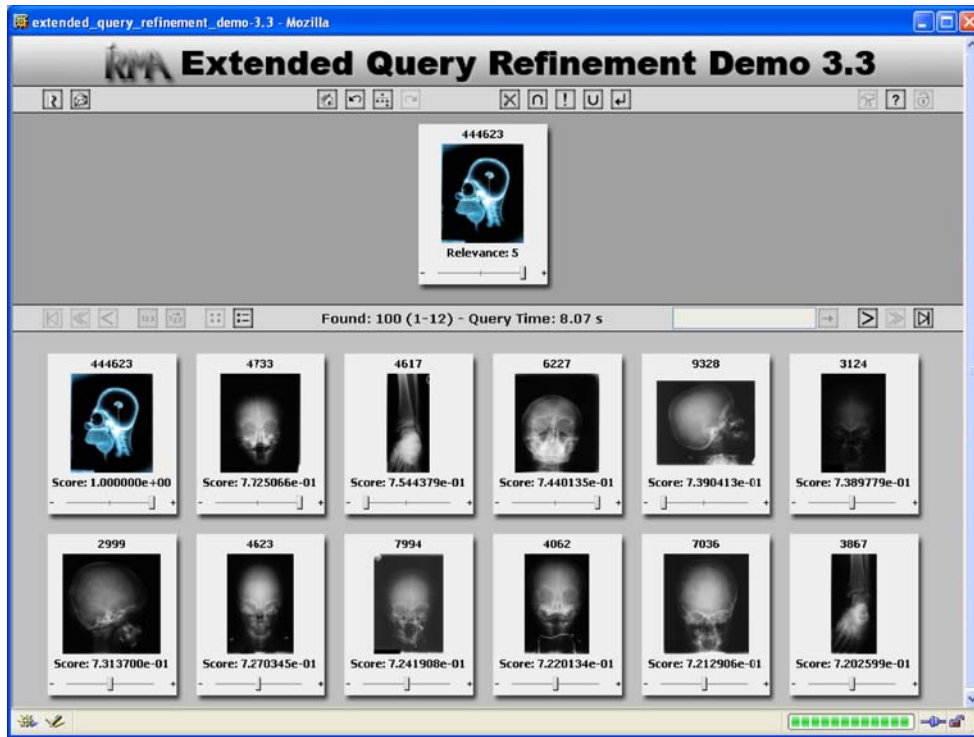


Fig. 3. IRMA GUI for content-based image retrieval.

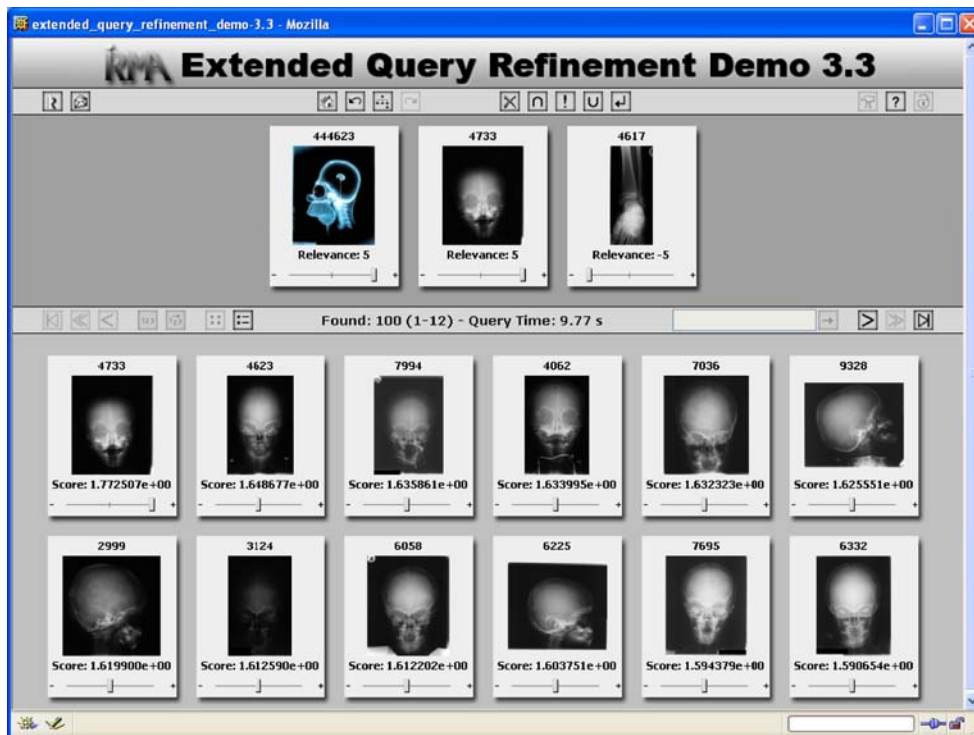


Fig. 4. IRMA GUI for content-based image retrieval after query refinement.

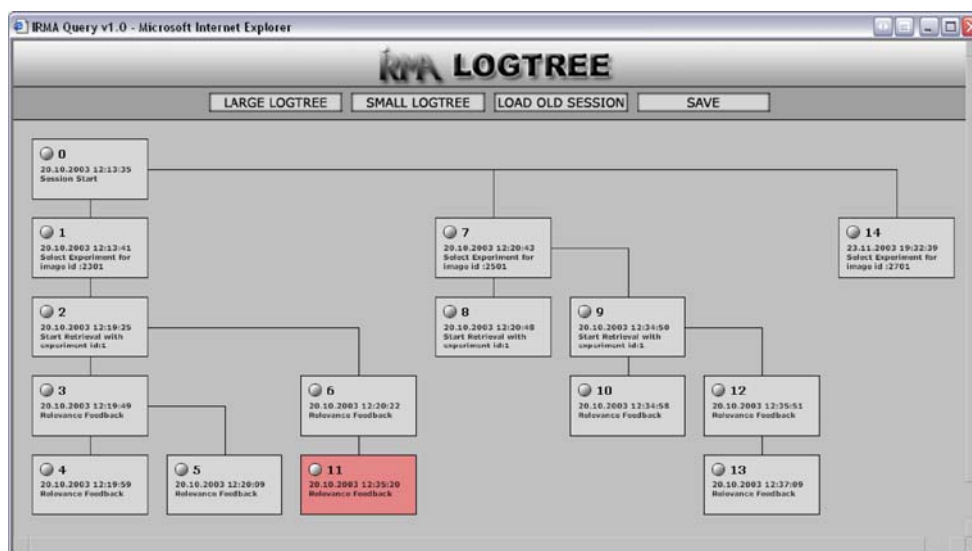


Fig. 5. IRMA GUI to select and restore previous system stages (history logging).

DISCUSSION

Relevance feedback (the “first loop”) is recognized as one of the most important mechanisms for CBIR user interaction.^{8–10} Although old systems such as the multimedia analysis and retrieval system (MARS) provided scores between -3 and 3 about 10 years ago,²⁶ relevance feedback in medical CBIR applications is usually restricted to labeling some of the returned images as “relevant” or “irrelevant”. Using our parameter modules, a continuous-value relevancy ranking is possible.

More important, a second loop of query refinement is offered by database-assisted query logging, which enables the user to step back and forward or to select any previous results state. This history functionality is critical for interactive CBIR system usability,^{24,25} as query refinement and relevance feedback, although expected to progressively improve system response in many cases, are not guaranteed to do so infallibly. According to Berlage, there are four possibilities to recall system states, which are based on storage of all²⁴

1. *Actions*: The recall is done by returning to the starting state of the system and recalling the actions one by one. Particularly for sessions with many user interactions, high processing times are required.

2. *Inverse actions*: Additionally, the inverse actions are stored to allow direct undo. However, inverse actions may be not trivial to be calculated.
3. *System states*: The recall is done by exchanging the state of all system operators according to the stored history. Here, the drawback results from the relatively high storage costs.
4. *State changes*: All the internal data structures that have been changed by an action are stored. This option is not as storage intensive, but, again, not trivial to implement.

Kreuseler et al. have presented a history mechanism for visual data mining that also results in a tree-like history structure, where the nodes of the tree can be labeled to identify system states.²⁵ However, the approach of Kreuseler et al. is based on the first variant, and, consequently, performs rather slowly. In contrast, our approach of query logging is based on the system states. In particular, the third variant is used in the retrieval interfaces avoiding any delay in execution. All system states are stored in the IRMA core database as XML-encoded text string. Without compression, a single stage is described by approximately 4 kb. Furthermore, our concept of query logging also supports variant four, which, for instance, is used

in the IRMA framework for manual reference labeling.

A third loop provides Boolean operations for image sets obtained from successive queries. Zhu et al. have already stressed the impact of Boolean operations among the queries,²⁰ 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 GUI using only push buttons. As our approach refrains from complex menus hosted in several Web pages, it can be handled most easily and can be adapted to a plenty of applications in radiology and medicine. The complete query logging provides all information that is required for the evaluation of the user's feedback.

Nevertheless, this paper is rather theoretical. Following ISO 9241, any software respective its GUI must be

1. *Effective*—all functions of clinical workflow for which the system is designed are in fact supported, and unexpected errors or system failures do not hinder the successful use of the application
2. *Efficient*—all tasks are performed with a minimum of user efforts, such as key-strokes or mouse clicks
3. *Satisfying*—all users are satisfied from the software and get the feeling of being supported by using the application.

Accordingly, the usability of our approach needs further evaluation, e.g., by a study showing impact of complete history provided to experienced users, or by performing a cognitive walkthrough, heuristic evaluation, or formal inspection.²⁷ This will be planned in the future.

CONCLUSION

CBIR is an emerging field of research, in particular for PACS. This paper contributes a methodology for extended query refinement including

1. A striking and unique feature of allowing a user to revisit previous session states and use those states as new start points in queries. This

can be done in arbitrary fashion, and is a great improvement over simple undo–redo

2. A capability for the user to provide continuous-valued relevancy feedback, rather than just judgments of “relevant” and “not relevant”
3. A capability to combine queries with complex combinations of Boolean operators
4. A technical description of the internal structures (Output Modules, Parameter Modules, Transaction Modules, Process Modules) that implement the extended query refinement and query logging
5. A GUI as a single Web page that may offer easy incorporation on other systems
6. A demo system on the Web and its interface to the IRMA core.

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>. The authors would also like to acknowledge the helpful comments of the reviewers that improved this manuscript.

REFERENCES

1. Tagare HD, Jaffe CC, Dungan J: Medical image databases—A content-based retrieval approach. *J Am Med Inform Assoc* 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. *Int J Med Inform* 73(1):1–23, 2004
3. Deselaers T, Müller H, Clough P, Ney H, Deserno TM: The CLEF 2005 automatic medical image annotation task. *International Journal of Computer Vision* 2007 (in press) DOI 10.1007/s11263-006-0007-y
4. Müller H, Deselaers T, Deserno TM, Clough P, Kim E, Hersh W: Overview of the ImageCLEFmed 2006 medical retrieval and annotation tasks. *Lect Notes Comput Sci* (in press)
5. Deserno TM, Güld MO, Deselaers T, Keysers D, Schubert H, Spitzer K, Ney H, Wein BB: Automatic categorization of medical images for content-based retrieval and data mining. *Comput Med Imaging Graph* 29(2):143–155, 2005
6. Zhou W, Smalheiser NR, Yu C: A tutorial on information retrieval: Basic terms and concepts. *J Biomed Discov Collab* 1:2, 2006
7. Celetyno A, Di Sciascio E: Feature integration and relevance feedback analysis in image similarity evaluation. *J Electron Imaging* 7(2):308–317, 1998

8. Rui Y, Huang TS, Ortega M, Mehrotra S: Relevance feedback—A power tool for interactive content-based image retrieval. *IEEE Trans Circuits Syst Video Technol* 8(5):644–655, 1998
9. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R: Content-based image retrieval at the end of the early years. *IEEE Trans Pattern Anal Mach Intell* 22(12):1349–1380, 2000
10. Naster C, Mitschke M, Meilhac C: Efficient query refinement for image retrieval. *Proceedings of the IEEE Computer Society Conference on Comput Vis Pattern Recognit* 547–552, 1998
11. 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
12. 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
13. Laaksonen J, Koskela M, Laakso S, Oja E: Self-organizing maps as a relevance feedback technique in content-based image retrieval. *Pattern Anal Appl* 4(2–3):140–152, 2001
14. Chen JY, Bouman CA, Dalton JC: Active browsing using similarity pyramids. *Proceedings SPIE* 3656:144–154, 1998
15. He XF, King O, Ma WY, Li MJ, Zhang HJ: Learning a semantic space from relevance feedback for image retrieval. *IEEE Trans Circuits Syst Video Technol* 13(1):39–48, 2003
16. 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
17. Vegas J, De la Fuente P, Crestani F: A graphical user interface for structured document retrieval. *Lect Notes Comput Sci* 2291:268–283, 2002
18. Fox KL, Frieder O, Knepper MM, Snowberg EJ: SENTINEL—A multiple engine information retrieval and visualization system. *J Am Soc Inf Sci* 50(7):616–625, 1999
19. Meiers T, Sikora T, Keller I: Hierarchical image database browsing environment with embedded relevance feedback. *Proceedings 2002 International Conference on Image Processing* 2:593–596, 2002
20. Zhu XQ, Zhang HJ, Liu WY, Hu CH, Wu L: New query refinement and semantics integrated image retrieval system with semiautomatic annotation scheme. *J Electron Imaging* 10(4):850–860, 2001
21. Deserno TM, Güld MO, Thies C, Fischer B, Spitzer K, Keysers D, Ney H, Kohlen M, Schubert H, Wein BB: Content-based image retrieval in medical applications. *Methods Inf Med* 43(4):354–361, 2004
22. Güld MO, Thies C, Fischer B, Deserno TM: A generic concept for the implementation of medical image retrieval systems. *Int J Med Inf* 76(2–3):252–259, 2007
23. Deserno TM, Plodowski B, Spitzer K, Wein BB, Ney H, Seidl T: Extended query refinement for content-based access to large medical image databases. *Proceedings SPIE* 5371:90–98, 2004
24. Berlage T: A selective undo mechanism for graphical user interfaces based on command objects. *ACM Trans Comput Hum Interact* 1(3):269–294, 1994
25. Kreuzeler M, Nocke T, Schumann H: A history mechanism for visual data mining. *Proceedings IEEE symposium on information visualization (INFOVIS)* 49–56, 2004
26. Rui Y, Huang TS, Ortega M, Mehrotra S: Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans Circuits Syst Video Technol* 8(5):644–655, 1998
27. Nielsen J, Mack RL Eds.: *Usability Inspections Methods*. Wiley, Chichester, UK, 1994