On-Line Images from the History of Medicine (OLI): Creating a Large Searchable Image Database for Distribution via World-Wide Web

R. P. Channing Rodgers
Suresh Srinivasan

Computer Science Branch
Lister Hill National Center for Biomedical Communications
National Library of Medicine
National Institutes of Health
Bethesda, Maryland 20894 USA

Abstract

The On-Line Images from the History of Medicine Project provides network access to a collection of nearly 60,000 images and their textual catalog descriptions. Development of this service has posed challenging technical hurdles, among them the capture of laser videodisc images, creation of a relational database for the catalog, creation of a forms-based user interface, and creation of a reliable and maximally simple state engine hiding behind the stateless HTTP protocol.

1. Introduction

The U.S. National Library of Medicine (NLM), a component of the U.S. National Institutes of Health (NIH), is the world's largest library dealing with a single scientific/professional topic, caring for over 4.5 million holdings. The NLM is actively interested in network-based means of disseminating biomedical information. The Lister Hill National Center for Biomedical Communications (LHNCBC), a research and development component of the NLM, is home to an experimental World-Wide Web server known as HyperDOC[1] which is intended as both a demonstration platform and experimental testbed for exploring and extending the technology which allows the distribution of multimedia- and hypertext-based information over the Internet.

In the 1980s, LHNCBC undertook development of a laser videodisc compendium of the NLM's History of Medicine Division (HMD) prints and photographs collection. This collection includes over 59,000 fine prints, photographs, ephemera and posters. The collection was photographed on 35 mm film, and the images were transferred to laser videodisc. The videodisc was used as part of a PC-based image retrieval system, known by various names during its development (HARPP, Picquick), but currently named Images from the History of Medicine [2].

On-Line Images from the History of Medicine (OLI) was intended to provide Internet access to this image collection and its accompanying textual catalog.

2. Capture of Image and Catalog Data

The images on the NLM Picquick laser videodisc (Version 3.93) were captured by a SPARCstation 10/41 running under SunOS 4.1.3, via a Sun VideoPix card, and saved in Sun rasterfile format using several *perl*[3] scripts and a C program. A configuration file for the controlling script defined individual videodisc frames as color or grayscale (the framegrabber could not automatically determine this). For sections of the videodisc on which color and grayscale images were mixed, all images were captured as color. The image files were post-processed by the controlling script, employing components of Poskanzer's *pbmplus* image file toolkit, and the public domain toolkit from the JPEG group; the script attempted to crop a gray border surrounding the actual image. Images were saved in both JPEG compressed format (full-size images) and in GIF format (thumbnail versions). About three percent of the images could not be processed by the script, and required manual editing using Bradley's *xv* program. The contrast of manually processed images was enhanced, and grayscale images converted to grayscale format (in the event that they were originally saved as color). Each image was assigned a unique identifier based on the side of the laser videodisc it appeared upon, and the frame number it occupied.

The entire image collection requires about 2.3 GB of disk. Typically, a JPEG file is about 30 KB in size, and the corresponding GIF thumbnail is about 6 KB in size.

An ASCII file containing the full catalog information for the collection was extracted from dBase-III files on a CD-ROM created for the Picquick project.

3. Functional Design Overview

On-Line Images makes extensive use of the forms capabilities of HTML+. When connecting to the HTTP server offering OLI, the user encounters a brief home page document with hypertext links to extensive documentation concerning the background, usage instructions, and technical limitations associated with the system. A small information icon appears here and elsewhere throughout the system to provide context-specific help. Forms-based buttons allow three modes of image retrieval:

1) Searching using text expressions

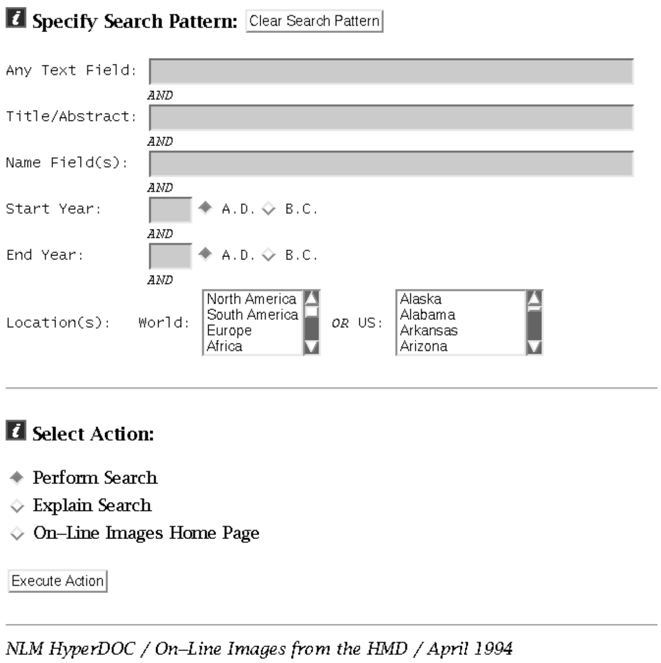
2) Searching by frame number

3) Random browsing (retrieves 10 images at random).

Selecting a search by text expression produces a form such as that shown in Figure 1.

The original catalog contains many distinct fields; for indexing, fields of a similar nature have been collapsed into broad categories to enable a simple interface. The user performs field-restricted searches by entering search patterns in any of the windows labelled: Title/Abstract, Name Fields(s), and Start and End Year (for which radiobuttons allow the selection of B.C. or A.D.). Menu boxes allow selection of specific geographical locations throughout the world, or within the U.S. A text window labelled "Any Text Field" applies to any of the title, abstract, topical heading, or name fields.

Words appearing within a text window are implicitly joined by Boolean OR operations unless separated by the word `and`, in which case they are joined by a Boolean AND. Similarly, the geographical locations from the two selection lists are joined by Boolean ORs. Text expressions appearing within a given text window are, in effect, enclosed within parentheses and joined by a Boolean AND to the contents of any other non-empty text window. Text can also take advantage of suffix truncation (for example, `nurs*` matches *nurse*, *nurses*, *nursing*, *nursery*, etc.). Enclosing multiple words within a pair of double quotes causes them to be searched for as an intact multi-word phrase. Stop words are filtered from catalog entries prior to indexing, and thus should not be employed within phrases or as an operand of a Boolean AND, as this will preclude any matches.



Specify Search Pattern: Clear Search Pattern

Any Text Field:

AND

Title/Abstract:

AND

Name Field(s):

AND

Start Year: ◆ A.D. ◇ B.C.

AND

End Year: ◆ A.D. ◇ B.C.

AND

Location(s): World: North America, South America, Europe, Africa *OR* US: Alaska, Alabama, Arkansas, Arizona

Select Action:

◆ Perform Search
◇ Explain Search
◇ On–Line Images Home Page

Execute Action

NLM HyperDOC / On–Line Images from the HMD / April 1994

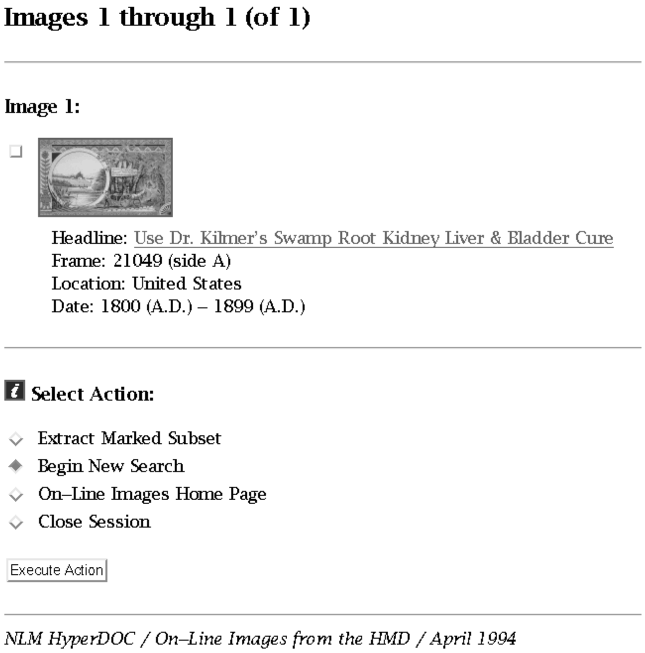
Figure 1. The form for searching by text expressions.

After a search pattern is specified, the user indicates whether the pattern is to be used to produce a simple English-language explanation of how the pattern will be interpreted, or to perform a search. The action is then triggered by a submit button.

When a search is completed, a summary page is returned, showing elapsed clock time and the number of images found. Images are returned for examination in browsing subsets; the user indicates how many images to send back in each browsing subset. Browsing subsets contain thumbnail images and brief catalog extracts (including a *headline*, a brief textual description drawn from the title, subtitle, or abstract fields; see Figure 2). An image marked in the catalog as restricted (for example, due to copyright) is accessible only at authorized sites; at other sites, an explanatory image is displayed in its place. Clicking on an abbreviated description displays the corresponding full catalog entry, and clicking on a thumbnail displays a full-sized image (Figure 3). Special characters such as foreign accent marks are correctly mapped into their HTML encodings prior to display. An image may be marked by selecting the checkbox at the upper left-hand corner of the corresponding thumbnail; marked images may be retrieved as a new retrieval set. Facilities of the WWW client may be used to print any of the display screens.

4. The Relational Database Search Engine

The catalog contains several text fields with descriptive information suitable for a word index. The retrieval mechanism is built around POSTGRES 4.1 [4], a freely available object-oriented relational database which employs the (QUEL-like) POSTQUEL query language. In the realm of object-oriented database systems, the traditional relational table becomes an *object class*. POSTGRES is limited in its usefulness as a text search engine, particularly for supporting features such as Boolean operations (AND and OR), phrase searching, and truncation. OLI uses POSTGRES to do what it does best — access the images via the frame number. A *Frame* class



Images 1 through 1 (of 1)

Image 1:

□

Headline: Use Dr. Kilmer's Swamp Root Kidney Liver & Bladder Cure
Frame: 21049 (side A)
Location: United States
Date: 1800 (A.D.) – 1899 (A.D.)

Select Action:

◇ Extract Marked Subset
◆ Begin New Search
◇ On–Line Images Home Page
◇ Close Session

NLM HyperDOC / On–Line Images from the HMD / April 1994

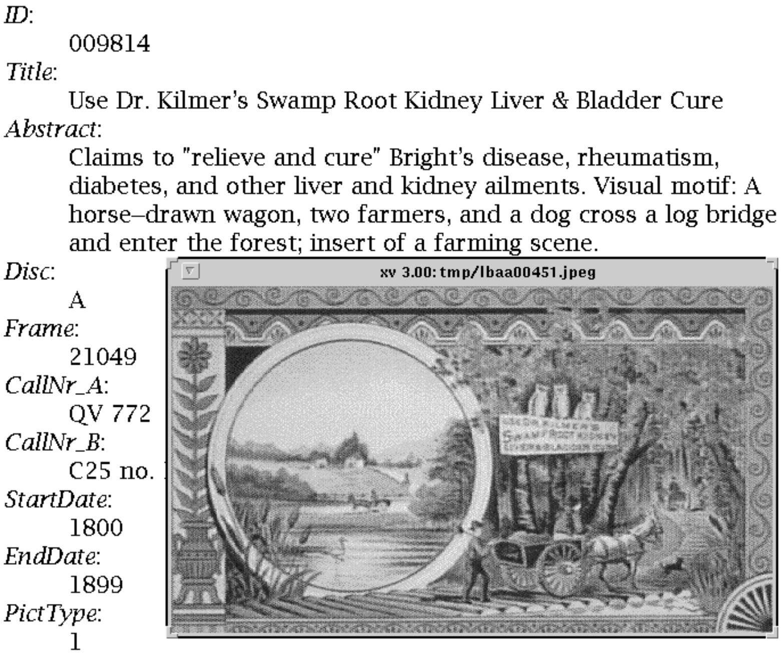
Figure 2. A sample browsing page (only one image).

contains all the information that is needed for image display, including abbreviated catalog data.

The catalog contains some 500,000 non-unique words (where a word is defined as a run of two or more alphanumeric characters). A *perl* script extracts these words from the title, subtitle, abstract, and topical heading fields, and multiple fields containing personal and corporate names. The text search mechanism is implemented (external to POSTGRES) as a simple binary search of a sorted file to identify the words in the query. Prefix matching is also done at this stage by identifying all words with the given prefix. Memory mapping facilities considerably speed up these searches. A temporary *Word* class is then created and loaded with these words. Since this class contains only the words in the query, scans of this class for boolean operations, truncation, etc., are reasonably fast. Finally, suitably qualified POSTQUEL commands are generated for the search. In the POSTQUEL, truncation is handled as a range query. For example, the query *kidney** generates: *...where w.word >= "kidney" and w.word < "kidnez"*. The upper limit on the range is automatically generated to be just greater than all the words with that prefix. Phrase searching is handled using word position information. The *Word* class has an attribute recording the position of the word in the catalog within a specific field. In a phrase, word positions must be consecutive and the words must all occur in the same field. Clearly, more sophisticated search mechanisms will be needed for a fully functional system.

The first non-empty field among the title, subtitle and abstract fields (in that order) forms the headline for the record. If all three fields are empty, the string "No title" is employed.

The field containing geographical information encodes the world region, the country and, for some countries like the U.S., state information. Not all records carry this information, however. These classes are not indexed and all searches on these classes (*World*, *State*) default to scans.



ID:
009814
Title:
Use Dr. Kilmer's Swamp Root Kidney Liver & Bladder Cure
Abstract:
Claims to "relieve and cure" Bright's disease, rheumatism, diabetes, and other liver and kidney ailments. Visual motif: A horse–drawn wagon, two farmers, and a dog cross a log bridge and enter the forest; insert of a farming scene.
Disc:
A
Frame:
21049
CallNr_A:
QV 772
CallNr_B:
C25 no.
StartDate:
1800
EndDate:
1899
PictType:
1

Figure 3. A full-sized version of the thumbnail image from Figure 2, superimposed on its complete catalog display.

The time period referenced in the time field varies, but the lowest level of granularity is a decade and the highest, a century. Matches are found if the time range requested in the search intersects the period spanned by the image.

5. The OLI State Engine

The program to access the On-Line Images from the History of Medicine, *nlmhmd*, is called from an unmodified HTTP server (such as NCSA HTTPD) that supports the CGI (Common Gateway Interface) specification, a standard for interfacing external gateways to HTTP servers. Keeping the server intact maximizes portability at the expense of slightly higher overhead. The program makes extensive use of HTML+ fill-out forms which support user interface elements such as text input fields, radio buttons, checkboxes, etc.

On the server end, *nlmhmd* stores the result set for each search. This contains the unique identifiers (UI's) of records matching a given query. Currently, each search is assigned a unique session UI. A separate file stores the mapping of session UI's to information about the client, such as the IP address, host name, last access time, etc. The program also maintains information about the subset of records that the user has marked. When the user requests the records that she has marked, the marked subset replaces the result set for that search.

The HyperText Transport Protocol (HTTP) underlying World-Wide Web is a stateless protocol, whereas a fully developed retrieval protocol such as Z39.50 maintains state to enable multiple sequential operations on a result set. OLI hides a state engine behind a stateless communications protocol. State information resides in files on the server and is also transmitted to the client in *hidden fields* within HTML+ forms. To this end, appropriate client support is critical. As of this writing, the only client supporting hidden fields within forms is NCSA Mosaic for X (version 2.2 or later). The result of a form submission is a POST query to the server. The CGI specifies how data from the client is passed to *nlmhmd*.

Hidden fields in forms encode state information that is used by *nlmhmd* to maintain context between communications with the server. They are used in the spirit of the slot/value notation where the names of the hidden fields are the slots and their contents are the values.

Field Name	Contents	Description
action	newsearch	Cleans up state and presents search page
	dosearch	Performs search and computes result set
	explainsearch	Explains the search about to be done
	getnext	Updates marked set; retrieves next set
	getmarked	Retrieves marked set
	searchframe	Presents the search-by-frame form
	getframe	Executes a search by frame
	getrandom	Retrieves a random set of images
	gethome	Retrieves the home page
	close	Cleans up state; links to home page
sessionUI	<time:number>	Contains time stamp and session number
batchstart	<integer>	Offset in result set for first in batch
batchsize	<integer>	Size of batch
framenum	<integer>	Frame number to search for
side	<char>	Side ‘A’ or ‘B’

Table 1. Hidden Field Names and Contents

The hidden fields and their permissible contents appear in Table 1. The fields (slots) used are:

action

This slot encodes the type of action that the program is to perform with the contents of the form. The program branches on the value and calls an appropriate function to perform the required action.

sessionUI

This slot encodes both the session number and a time stamp for that form. On submission, the program checks the time stamp to eliminate stale sessions, where the session number may already have been assigned to another session. The session number is used as an index into a session database to obtain information about the session.

batchstart

The content of this field specifies which record (from the result set) should be made the first of a batch (default: zero).

batchsize

This slot specifies how many records to retrieve and is set by the user selecting from a menu (default: 5).

framenum
This slot is used when searching for a specific frame.

side This slot is also only used when searching for a specific frame and encodes the side (A or B) of the original videodisc the frame was on.

Nlmhmd parses the data that it receives via the CGI and performs the necessary action. The results are then packaged into an appropriate (dynamically generated) HTML page and returned to the client. A simplified state transition matrix for the program is shown in Table 2.

EVENT	STATE	
	Stale/Non-existent Session (State 1)	Active Session (State 2)
`newsearch`	Present the search page	Clear state; present search page; set state=1
`dosearch`	Compute new result set; return results page; set state=2	Assign new session; compute new result set; return results page*
`explainsearch`	Return an explanation page	Return an explanation page
`getnext`	Error: present new search page	Store marked images, if any; return requested batch
`getmarked`	Error: present new search page	Generate the marked subset; return results page
`searchframe`	Present the search-by-frame page	Present the search-by-frame page
`getframe`	Return record(s) for requested frame(s)	Return record(s) for requested frame(s)*; set state=1
`getrandom`	Return random records	Return random records*; set state=1
`gethome`	Present home page	Clear state; present home page
`close`	Clean up state if any	Clean up state if any; state=1

* Note that these events orphan the existing session, which will eventually timeout.

Table 2. State Transition Matrix

6. Discussion

On-Line Images from the History of Medicine is a prototype system which demonstrates the speed with which previously created data can be repackaged and offered as a network service via World-Wide Web technology. The entire project took about one man-month on the part of a highly skilled programmer.

The On-Line Images Project is distinguished from earlier efforts at Internet-distributed image databases, such as the collection of solar magnetograms created by CNIDR [5], by the size of its collection, the relative richness of its catalog structure, a fuller use of HTML+ forms capability, and by the increased sophistication of interactions which its state engine allows.

There are a number of improvements and additional features under development which will enhance the utility of OLI:

1) Rescanning the images from film, at higher resolution and larger size (this would increase disk requirements about threefold). Currently, the resolution is limited by the inherently low resolution of NTSC television, compounded by further losses in the inexpensive Sun

VideoPix framegrabber and in the use of "lossy" JPEG compression. Images should be offered in multiple sizes to accommodate users with slow network connections.

2) Completion of the catalog data, which is rather sparse at present.

3) Replacement of POSTGRES with a cleaner and faster searching engine. Explore alternate search engines; for example, allow a WAIS-like search using a non-Boolean free-text search expression, with the production of matching images ranked according to some weighting scheme (currently, the sparseness of the catalog information makes this impractical).

4) Creation of a mechanism to allow interactions with other applications. A prototype link currently connects NetCoach, a UMLS[6] Metathesaurus™ browser, to OLI (NetCoach helps refine the search pattern to be used for biomedical information retrievals). This communication mechanism builds upon a set of pre-defined name/value pairs, that are conveyed between independent applications within form-based hidden fields. These name/value pairs trigger pre-defined actions, and can be used to customize forms. For example, the suggestion box mechanism for OLI and HyperDOC, called *mailform*, is a customizable form for sending electronic mail, triggered by using either a GET or POST request. The suggestion box form contains various text windows (sender address, recipient address, subject, body of message). These have built-in default values that can be supplanted by application-specific defaults. For example, to have the address text field of the suggestion box form appear with the recipient address already filled in, the URL pointing to this application can be specified as:

   `http://`*hostname*`/cgi-bin/mailform?recipient=`*emailaddr*

   (where *hostname* is the name of the server, and *emailaddr* is the intended mail address). Alternatively, the address can be embedded within a form of the invoking application, by setting the value of the hidden field named *recipient* to the desired address.

5) Mapping of all special codes employed in catalog records into appropriate human-readable descriptive text.

6) Expansion of the state engine. For example: support multiple searches within a session, allowing operations between multiple retrieval sets.

7) Enhancement of the sophistication of Boolean expressions, supporting grouping by parentheses, appropriate rules of precedence, and the NOT operator.

8) Generalization of the phrase searching mechanism to provide full proximity searching (specification of the distance within which words must occur).

9) Implementation of a more general partial pattern-matching capability, perhaps even a full implementation of UNIX-like regular expressions (unfortunately, elaborate indexing mechanisms would be needed to accomplish this).

10) Allowing ranges and lists of non-consecutive frame numbers to be used when retrieving by frame number.

11) Expansion of the random browsing mechanism to allow the user to specify the number of images to return.

12) Allowing the user to specify sorting criteria for ordering of the returned images.

13) Creation of a more flexible search page in which features are configurable by the user.

Several deficiencies result from technological shortcomings beyond the scope of OLI itself:

1) The intensity of images will vary depending upon the gamma value of the display device. There is no universally employed mechanism to automatically compensate for this effect.
2) A HTML+ form may be associated with only one submit button. Thus the forms interface requires two selections (specify action, perform action) where one should have sufficed (select and perform action).

Acknowledgements

We thank the creators of the PicQuick laser videodisc project; Dr. Daniel Masys provided the ASCII version of the dBase-III catalog data; Jim Fullton of CNIDR shared his experience with his Internet-accessible solar magnetogram database; our colleagues in the Computer Science Branch provided useful testing and suggestions.

References

[1] The Uniform Resource Locator (URL) for HyperDOC is: `http://www.nlm.nih.gov/`.

[2] W. R. Leonard and J. R. Stokes, *Proceedings, Interactive Videodisc in Education and Training, Twelfth Annual Conference, Society for Applied Learning Technology, August 22-24, 1990*, p. 40-44, Washington, D.C., 1991.

[3] L. Wall and R. L. Schwartz, *Programming Perl*, O'Reilly & Associates, Inc., Sebastopol. CA, 1990. (454 pages).

[4] S. Wensel, ed., ‘‘The POSTGRES Reference Manual,’’ *Report M88/20*, Electronics Research Laboratory, University of California, Berkeley, CA, March 1988.

[5] J. Fullton, ‘‘Distributed image archives,’’ *Proceedings, Astronomical Data Analysis Software and Systems (ADASS), Victoria, British Columbia, Oct 1993*, (in press, 1994). The Uniform Resource Locator (URL) for the CNIDR documentation is: `http://cnidr.org/cnidr_papers/archives.html/`; the service itself can be accessed at: `http://argo.tuc.noao.edu/`.

[6] B. L. Humphreys and D. A. B. Lindberg, ‘‘The UMLS project: making the conceptual connection between users and the information they need,’’ *Bulletin of the Medical Library Association*, 81(2):170-177, 1993.