Lost in Hyperspace?
Free Text Searches in the Web

Christian Neuss,
Stefanie Höfling
Fraunhofer Institute for Computer Graphics
Wilhelminenstr.7, 64283 Darmstadt, Germany
e-mail: neuss@igd.fhg.de, hoefling@igd.fhg.de

Abstract

The World Wide Web (WWW) [LCG92] is a distributed hypermedia system for information discovery, retrieval, and collaboration. The hypertext paradigm has proven its usefulness for browsing large, distributed document structures. The ease of use provided by this paradigm is one of the reasons for the great popularity which the World Wide Web has gained through the last months.

However, as the amount of information available through the World Wide Web grows, it becomes more and more important to provide additional tools and techniques for finding servers or documents which contain relevant information on a given topic. Bibliographic and literature text searching is a difficult task, the provision of a free text search mechanism can greatly facilitate this task. By supplying a pre-computed index of keywords, a fully indexed server eliminates the need for automatic indexers (such as web robots or spiders) to walk the entire server tree, which is an unnecessary waste of resources.

Some World Wide Web servers already implement keyword searches via an interface to WAISINDEX. However, this approach lacks many important features that free text search engines provide, and does not support remapping of physical directory structures to virtual paths. In this document, we will present the ICE indexing server extension which has recently been developed at the Fraunhofer Institute for Computer Graphics. This freely available software package provides a set of routines which allow for sophisticated free text searches on a World Wide Web archive.

1. Introduction

The main difference between a hypertext network and conventional linear text is that in a hypertext system, navigation is up to the user. There is more than just one way of traversing a hypertext graph, in fact, there are often so many alternatives to chose from that the user easily gets lost. Offering features like backtracking or a graphical representation of the access "history" can make things a bit easier. However, when there is a well defined goal, a more efficient means of identifying relevant information than just trying out different hypertext links has to be found. One possible way of finding relevant documents is performing a keyword search. By providing an inverted index of words, a text database system allows for fast and efficient evaluation of a keyword search. This approach has been taken in WAIS (See [Kah91]). Keyword searching can be done in the World Wide Web too, usually by using a gateway to WAISINDEX or other indexing/retrieval software. The problem is that these indexers are not aware of HTML syntax and the logical hierarchy of virtual document paths.

In the Image Communication Information Board (ICIB) ([GeKr93], accessible via the URL http://icib.igd.fhg.de/icib/icib-home.html) information on standards, activities and organizations in the area of image communication standardization is being collected and made available to the internet community. Access to these documents is not only possible via hyperlinks, but also through sophisticated free text search mechanisms.

In order to serve the specific needs of such a text database, some requirements have been identified:

- Abbreviations are common in technical documents. They need to be treated in a special way, since otherwise a search for e.g. "IS" (which stands for "international standard") would find all documents containing the word "is", which is probably unwanted, at least if the archive serves documents in the english language.
- A large archive is usually structured. Documents are not all put in one directory, but are organized into a hierarchical structure. The indexer should provide a mechanism to define sub-views on the document hierarchy.
- The ability to perform conceptual searches is an important feature. Using a technical thesaurus allows for topic searches that are not restricted to the literal string, but can be extended to all synonyms of a given term.

In order to establish a WWW server which provides this text database functionality, the ICE indexing server extension has been developed. Although the ICIB archive deals with documents from the area of image communication, the above criteria apply as well for other areas like e.g. medicine or law archives. The ICE indexer is freely available and easy to set up and maintain. Among its features are:

- relevance feedback for documents: documents are presented with their title as a hyperlink to the file itself, together with a number indicating how many "hits" the search has found
- search terms can be combined with 'and' and 'or'
- pattern matching through full regular expressions syntax
- fault tolerant retrieval: using the Levenshtein algorithm for approximate matching of terms, documents that contain words which are 'similar' to a given term can be retrieved
- use of a thesaurus to make conceptual searches: by providing a server side thesaurus, it is possible to e.g. extend a search for "picture" to files containing semantically related terms like "image" etc.
- easy to use forms interface as an alternative to complex query language statements

The following section contains a more detailed description of these features. Section 3 contains some screenshots which demonstrate how ICE searches are perfomed. After that, section 4 gives an outlook of how ICE can be combined with an archie like search mechanism in order to perform world wide topic searches on Web archives.

2. The ICE search engine

Doing a topic search on a document archive is a difficult task. Use of mechanisms like subject specific thesauri, fault tolerant searches and wildcards help finding relevant documents by extending a search to similar words or terms with a related meaning. Also, search queries can be refined by using a boolean combination of search terms. This section describes the mechanisms implemented in the ICE search engine.

2.1 Keywords and boolean expressions

The simplest form of a query statement is by specifying a keyword: A query with the keyword "picture" will simply retrieve a list of documents containing the string "picture". To speed up the process of searching through documents, the ICE search engine uses an inverted index of words. It contains a list of every word in a document and it's frequency. The search algorithm only matches full words, so a query for "dim" will not match "dimension". However, searches for substrings are still possible through the use of wildcards: wildcards allow for selecting all strings matching a specific pattern. The ICE search engine implements regular expressions using the same syntax as the UNIX line editor "ed". Many programs and utilities use "ed"-style regular expressions. The regular expression syntax is the following:

- Any character except a special character matches itself. Special characters are the regular expression delimiter plus \[. and sometimes ^*$.
- A . matches any character.
- A \ followed by any character except a digit or () matches that character.
- A nonempty string s bracketed [s] (or [^s]) matches any character in (or not in) s. In s, \ has no special meaning, and] may only appear as the first letter. A substring a-b, with a and b in ascending ASCII order, stands for the inclusive range of ASCII characters.
- A regular expression of form 1-3 followed by * matches a sequence of 0 or more matches of the regular expression.

Searches are case insensitive, so it does not matter if e.g. a word at the beginning of a sentence starts with a capital letter. However, in order to allow for special treatment of abbreviations, words with more then one capital letter are considered special terms and added separately to the index file. The algorithm employed for the search access is as follows:

- A search term which is all lowercase, or has only the first letter capitalized will match all case insensitive occurrences of the word.
  Example: Searching for "is" will find both "is and "IS".
- If any other letters are capitalized, the search engine treats the word as a special expression and only finds exact matches.
  Example: "ICE" only matches "ICE", but not "ice".

The combination of two or more keywords can be used to create more complex queries. Combining query terms with the operator "OR" retrieves files containing either term. Connecting the keywords using the "AND" operator will limit the search to those documents containing both strings: By issuing the query "computer AND picture", only documents containing both words will be retrieved. Such a query will retrieve documents that deal with computer generated or manipulated images (although it will also find text fragments like "The book gives a detailed picture of what’s going on inside a computer"). Queries can be built up using multiple keywords connected with AND and OR, the AND operator has precedence.

2.2 Thesaurus

Another important feature is the use of a built-in thesaurus. Text processing systems often offer thesauri in order to help find synonyms, but a thesaurus is much more than a synonym list. It is a semantic network containing concepts that are related to one another in various ways. Since thesauri allow for dealing with a concept instead of just a single keyword, they can be very useful in free text queries. An example from the computer graphics area is that the words "image" and "picture" are often used synonymously, so a query for "picture" should also retrieve files containing the string "image".

Use of the thesaurus can be switched on and off by using the set notation in the query string. For example, typing " *{picture}* " will retrieve all documents containing either the word picture or related concepts (it could be read as *"retrieve the set of all concepts related to picture"*), while searching for *"picture"* retrieves only those documents which contain the literal string.

The ICE search engine uses the ANSI standard **Thesaurus Image Format (TIF)**. Thesaurus information can be compiled "by hand", but is available from other sources as well. For instance, the Library of Congress has maintained a hierarchy of subject headings which is commercially available on CD-ROM. Depending on the content of the World Wide Web archive, it probably makes sense to compile a specific technical thesaurus.

2.3 Document hierarchies

Search queries can also be refined by using a document relative context. Documents are accessed by using so called *Uniform Resource Locators* (URL, see [Lee93]). Although they do not necessarily reflect a file system hierarchy, they are used to indicate document hierarchies. Hence, specific subset of documents can be addressed through their URL. The context can be defined for the whole archive or in a way that is depending on the file from where the query has been issued (this requires a search capable server like the CERN httpd).

This is being achieved by having an optional file *.context* in every directory, which consists of a context name and the corresponding URL path. Here's an example:

```
video /icib/tv
ccir  /icib/tv/ccir
```

The syntax of the context query is as follows:

```
"{picture} AND digital CONTEXT video"
```

Note: In the current release, the indexing software retrieves files local to the directory resp. subdirectories of the file where the query has been issued. This behavior can be modified using the CONTEXT modifier. Please note that future implementations may change this to rather have a search include all directories, and have a special context identifier "LOCAL" in order to make the search local to the current directory subhierarchy.

2.4 Fault tolerant searches

In order to widen a search to related or similar terms, a fault tolerant search mechanism using an algorithm developed by V.I. Levenshtein [Lev65] can be employed. The Levenshtein algorithm defines a distance between two words as the weighted sum over the number of character deletions, insertions and changes needed to transform one word into the other. By defining a word to be a match when the Levenshtein distance of the word and the given search term lie within a certain limit, searches are tolerant towards typographical errors. Such errors are not only introduced by typing in texts through a keyboard, but also by using optical character recognition software for scanning in printed documents.

Fault tolerant searches also extend to variants of the given term: For example, by setting the Levenshtein distance limit to three, word variants like "rendering" and "renderer" are recognized as matches.

3. Examples

In this section, we will use some screenshots to show how the ICE search extension works, and how it interfaces with a World Wide Web user. Access to the extension can be provided through two mechanisms:

- a script interface, compliant with the CGI Specification, and
- a server modification which allows for entering search queries from within other HTML documents.

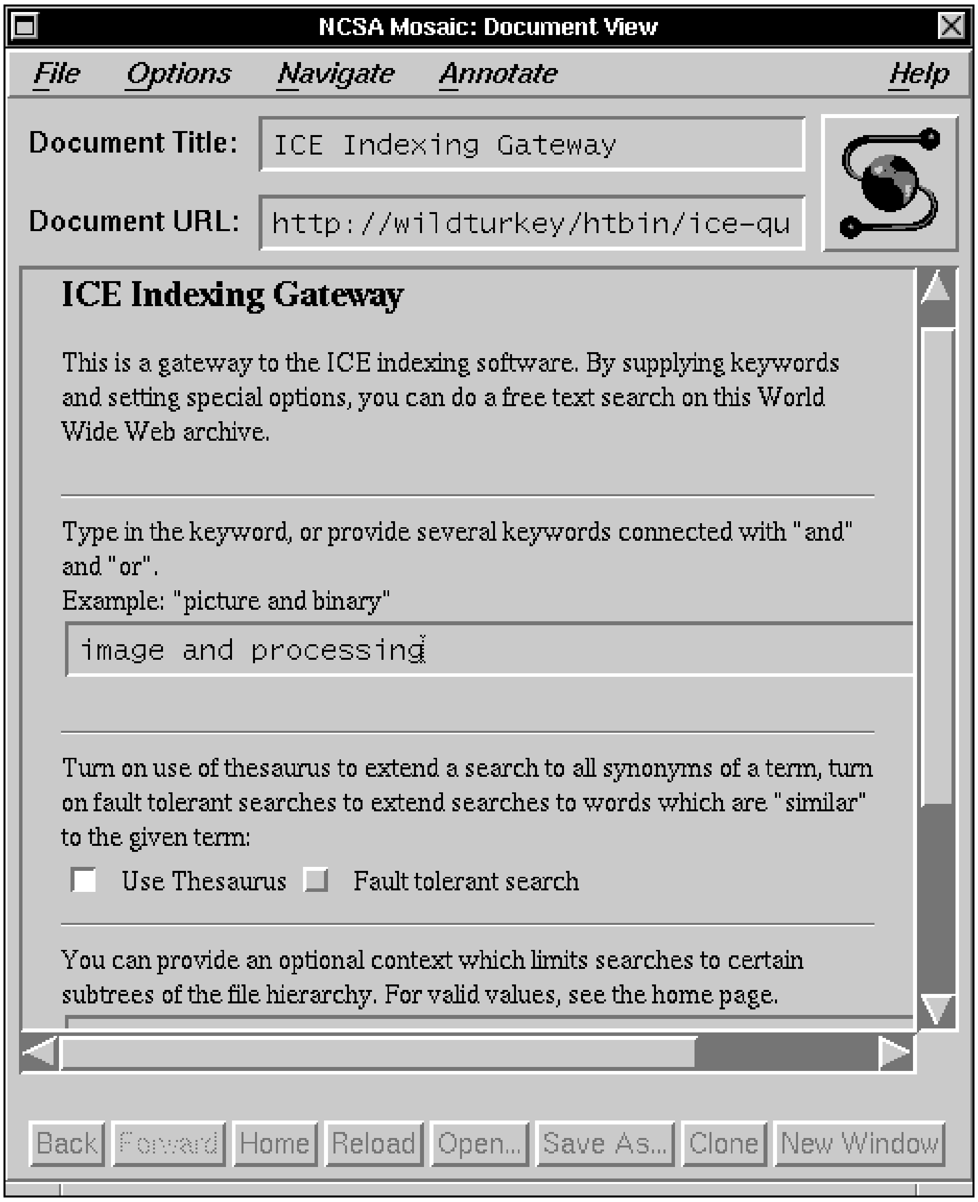

Figure 1: Forms interface to the ICE search engine

Figure 1 shows a CGI server backend that serves as a gateway to the ICE engine. With input forms as implemented in NCSA's Mosaic browser [And93], an easy to use interface has been provided. Instead of dealing with complex switches and special characters, features like use of the thesaurus can be toggled by simply clicking a checkbox.

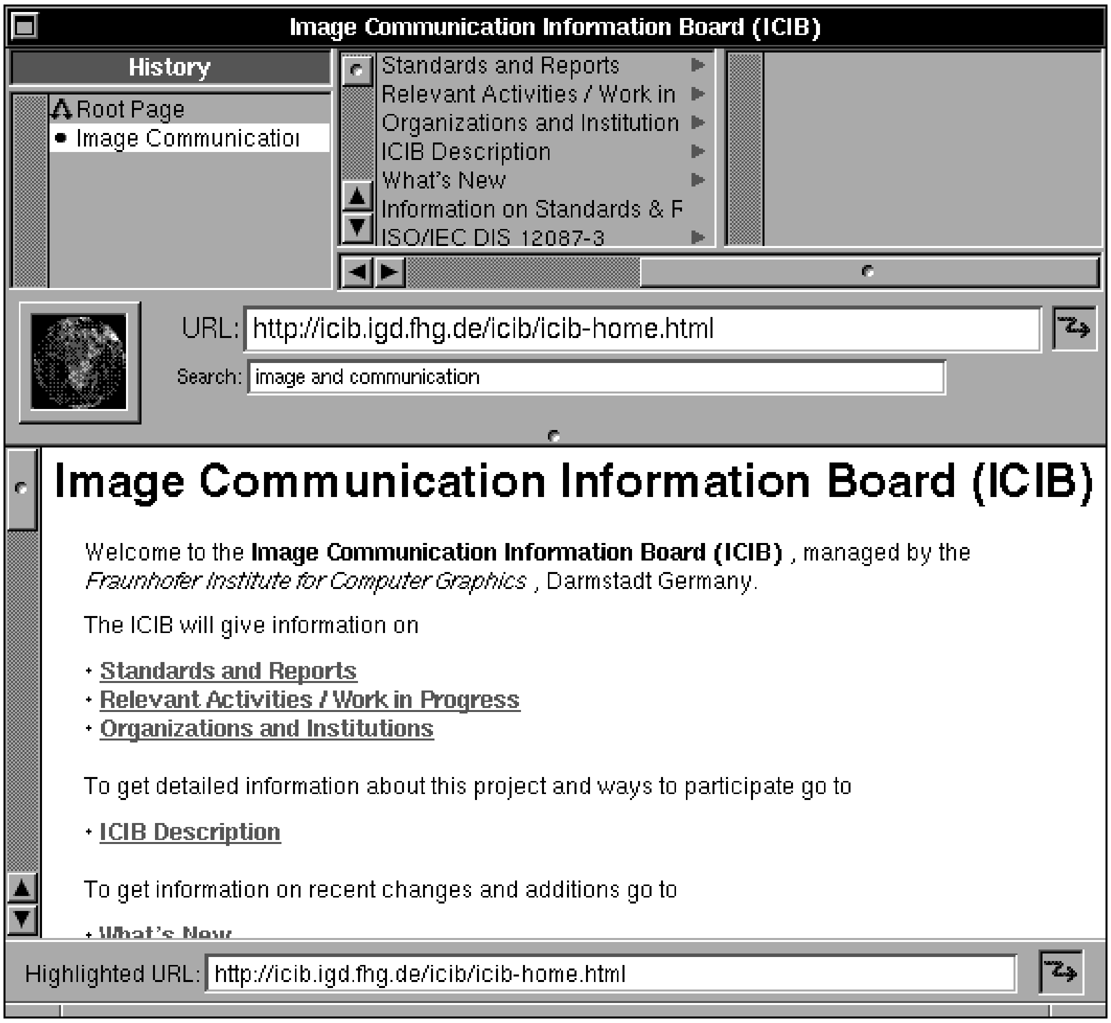

Figure 2: Entering a search query using the <ISMAP> input field

Since only few World Wide Web browsers support forms yet, a second CGI interface is also provided, which uses the *ISMAP* functionality for entering queries. The same mechanism can be employed for making free text queries from within documents. This is being demonstrated in figure 2: When encountering a document that has "<ISMAP>" in it's document header, browsers are supposed to present some means of entering keywords. Entering index queries is possible with almost all browsers in use today, the screenshots in figure 2 and 3 were made with the OmniWeb browsing tool, while figure 1 shows NCSA's Mosaic browser.

In this field a search query can be entered which gets sent to the server. The server then calls up a script to compute the query results and sends this result back to the client. This approach has the advantage that by using the information from where a query has been issued, it is possible to e.g. restrict a search to it's document sub-tree. However, this requires a server that is capable of handling queries, like e.g. CERN's httpd. A small patch to include this feature in the NCSA server is also part of the ICE distribution.

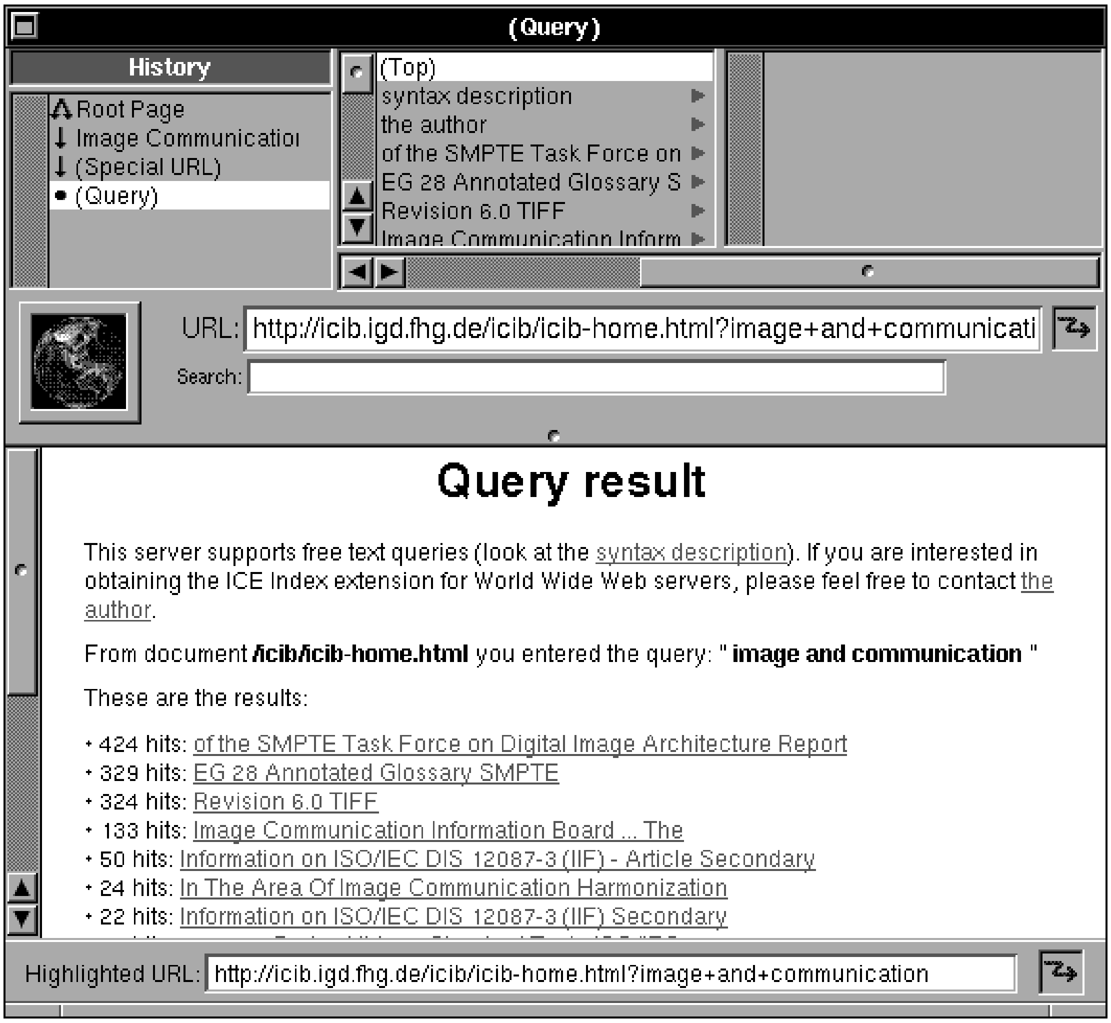

Figure 3: Query result

The result of the query request is shown in figure 3. A virtual document with a list of all matching documents is created. The documents are shown with their title, which is at the same time the source of a hyperlink leading to the document itself.

4. Conclusions

The continuous growth of the World Wide Web makes it very hard to keep track of all the services provided, and creates the need for further navigation and retrieval mechanisms. By integrating software tools for indexing and searching of documents, a World Wide Web archive can be accessed like a free text retrieval system. The ICE search engine allows for such free text searches, and provides special features like a thesaurus, fault tolerant matching and an indexer that is aware of virtual paths. Powerful free text searches can turn what used to be a complex graph of loosely coupled documents into a large free text database. The ICE software is freely available, a copy can be retrieved via anonymous ftp from `ftp.igd.fhg.de`, currently in the directory /`outgoing`.

The next step to take is to allow for topic searches not only on a Web archive, but on a world-wide scale. This can be achieved through an archie style indexing of World Wide Web servers, for example following the ALIWEB proposal. The ICE software can be used as a data-gathering script to provide an index file for archie like indexers such as ALIWEB (see `http://web.nexor.co.uk/aliweb/doc/aliweb.html` and [Kos94] for details).

This approach allows for world wide searches based not on file names, but on characteristic keywords provided by the document authors, for example through the HTML+ META element (which was added as a general way of enriching HTML documents with meta information).

In order to make a topic search, ALIWEB or a similar mechanism can be employed to identify a number of servers which are of interest, followed by an ICE free text search on the document tree. This two-staged search can be made accessible through a single CGI gateway program, thus providing an easy to use world-wide topic search mechanism.

References

[And93]
Marc Andreessen: NCSA Mosaic Technical Summary, NCSA Technical Document, NCSA Springfield, Champaign IL, May 8, 1993

[GeKr93]
Norbert Gerfelder, Detlef Krömker: Harmonisierung von Bildkommunikationsstandards, BERKOM Teilprojekt Nr. 2006 Zwischenbericht, May 1993

[Kah91]
Brewster Kahle: An Information System for Corporate Users: WAIS, Thinking Machines Corporation, April 1991

[Kos94]
Martijn Koster: ALIWEB - Archie-Like Indexing in the WEB, submitted to the WWW94 conference, CERN, Switzerland, May 1994

[LCG92]
Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, Bernd Pollermann: World-Wide Web: The Information Universe, in Electronic Networking: Research, Applications and Policy Vol.1 No.2, Meckler, Westport CT, Spring 1992

[Lee93]
Tim Berners-Lee: Uniform Resource Locators, Internet Engineering Task Force Draft, CERN, Geneva, Switzerland, July, 1993

[Lev65]
V.I. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals, Soviet Physics Doklady, Vol 10, 1965