MetaSEEk: A Content-Based Meta-Search Engine for Images

Mandis Beigi, Ana B. Benitez, and Shih-Fu Chang*
Department of Electrical Engineering & New Media Technology Center
Columbia University, New York, NY 10027

ABSTRACT

Search engines are the most powerful resources for finding information on the rapidly expanding World Wide Web (WWW). Finding the desired search engines and learning how to use them, however, can be very time consuming. The integration of such search tools enables the users to access information across the world in a transparent and efficient manner. These systems are called meta-search engines. The recent emergence of visual information retrieval (VIR) search engines on the web is leading to the same efficiency problem. This paper describes and evaluates MetaSEEk, a content-based meta-search engine used for finding images on the Web based on their visual information. MetaSEEk is designed to intelligently select and interface with multiple on-line image search engines by ranking their performance for different classes of user queries. User feedback is also integrated in the ranking refinement. We compare MetaSEEk with a base line version of meta-search engine, which does not use the past performance of the different search engines in recommending target search engines for future queries.

Keywords: MetaSEEk, meta-search engine, content-based visual query, color search, texture search, performance monitoring, World Wide Web

1. INTRODUCTION

The explosive growth of the World Wide Web has motivated the development of many search engines to assist the unmanageable task of navigating the Web. They try to satisfy the users' information needs for newspaper articles, software, movie reviews, books, music recording, images, video, etc. Two types of search engines can be found on the Web: large-scale robot-based and specialty search engines. Large-scale search engines try to index the contents of the entire World Wide Web, but usually fail to disseminate between desired data and unneeded information. On the other hand, specialty search engines are more focussed databases, which can not be applied to general topics.

Experienced users of the Internet would begin to query the appropriate specialty search engines to obtain desirable results, and continue querying general search engines when the specialized engines fail to yield helpful information. Nevertheless, the proliferation of search engines has replaced the problem of finding information on the Internet with the problem of knowing where search engines are, what they are designed to retrieve, and how to use them. Consequently, searching the Web for specific information has become a very time consuming and inefficient task for even the most expert users.

This situation has motivated the recent research and development in integrated search or meta-search engines [1]. Meta-search engines serve as common gateways, which automatically link users to multiple or competitive search engines. They accept requests from users, sometimes, along with user-specified query plans to select target search engines. The meta-search engines may also keep track of the past performance of each search engine and use it in selecting target search engines for future queries. Many approaches have been proposed for meta-searching. Section 2 presents an overview of these approaches, the majority of which have been designed for text databases.

Digital images and video are becoming an integral part of human communications [2]. The ease of creating and capturing digital imagery has trigged the recent development of visual information retrieval (VIR) systems on the web

Further author information –
M.B.: Email: mandis@ctr.columbia.edu
A.B.B.: Email: ana@ctr.columbia.edu
S.C.: Email: sfchang@ctr.columbia.edu

[3,4,5]. These systems usually provide methods for retrieving digital images by using examples and/or visual sketches. In order to query the visual repositories, the visual features of the imagery, such as colors, textures, shapes, etc, are used in combination with text and other related information. Everyday, users are finding new VIR systems on-line what leads, once more, to the problem of efficiently and effectively retrieving the information of interest.

We have developed a prototype meta image search engine, MetaSEEk, to investigate the issues involved with efficiently querying large, distributed on-line visual information sources. Our meta-search engine, MetaSEEk, adopts the principle that Web resources should be used efficiently. For each query, MetaSEEk selects the target engines that may the desire results by weighing search tools' successes and failures in similar query conditions. The implementation of the meta-search engine is described in section 3.

Section 4 describes the experiments, the evaluation measures and the comparison results between the MetaSEEk prototype and a base line search-engine that randomly selects the search engines to send the queries to. An interesting issue examined in MetaSEEk is the reliability of the selection and ranking of the remote search engines for different type of queries. Another important technical aspect is the heterogeneity among the different remote search engines and possible technical approaches to enhance interoperability. Finally, section 5 closes with concluding remarks and open issues for future research.

2. RELATED RESEARCH

Meta-search engines serve as common gateways, linking users to multiple search engines in a transparent manner. Working meta-search engines include three basic components, as depicted in Figure 1[1]. The dispatching component selects target search engines for each query. The query interface component translates the user-specified query to compatible scripts to each target search engine. The display interface component merges the query results from each search engines, removes duplicates and displays them to the user in a uniform format.

At the present time the wealth of meta-search engines on the WWW is still growing. Many approaches have been proposed for meta-searching. We overview a few of these efforts.

The GlOSS (Glossary-of-Servers Server) project [6] uses a meta-index to estimate which databases are potentially most useful for a given query. This meta-index is constructed by integrating the indexes of each one of the target databases. For each database and each word, the number of documents containing that word is included in the meta-index. The two main drawbacks of this approach are: first, it requires each of the search engines to cooperate with the meta-searcher by supplying up-to-date indexing information, and second, as the number of databases increases, the administrative complexity may become prohibitive.

The Harvest system [7] is being designed and built by the Internet Research Task Force Research Group on Resource Discovery. Harvest consists of several subsystems: a Gatherer collects indexing information and a Broker provides a flexible interface to this information. It is intended to be a scalable form of infrastructure for building and distributing content, indexing information, as well as for accessing Web information

Wide Area Information Servers (WAIS) [8] divide its indices among the databases into multiple levels with the top-level index containing a "directory of servers". Given a query, the "directory of services" is searched and the query is then forwarded to selected databases.

MetaCrawler [9] is a meta-search service developed at the University of Washington that integrates a set of general Web search engines. When a query is submitted, MetaCrawler dispatches queries to each one of those search engines, retrieves the HTML source of all the returned documents, and applies further analysis to clean up unavailable links and irrelevant documents. MetaCrawler obtains high precision but at the cost of network utilization.

The SavvySearch meta-search tool [1] employs a meta-index approach for selecting relevant engines based on the terms in a user's query; previous experience about query successes and failures is tracked to enhance selection quality. SavvySearch selects resources for an individual user's query and balances resource consumption against expected result quality by querying the most relevant search engines first. Their experimental finding suggest that a meta-index approach

can be effective in making search engine selection decisions. However, the potentially large amount of knowledge required to make decisions raises some questions about the overall efficiency of the system.

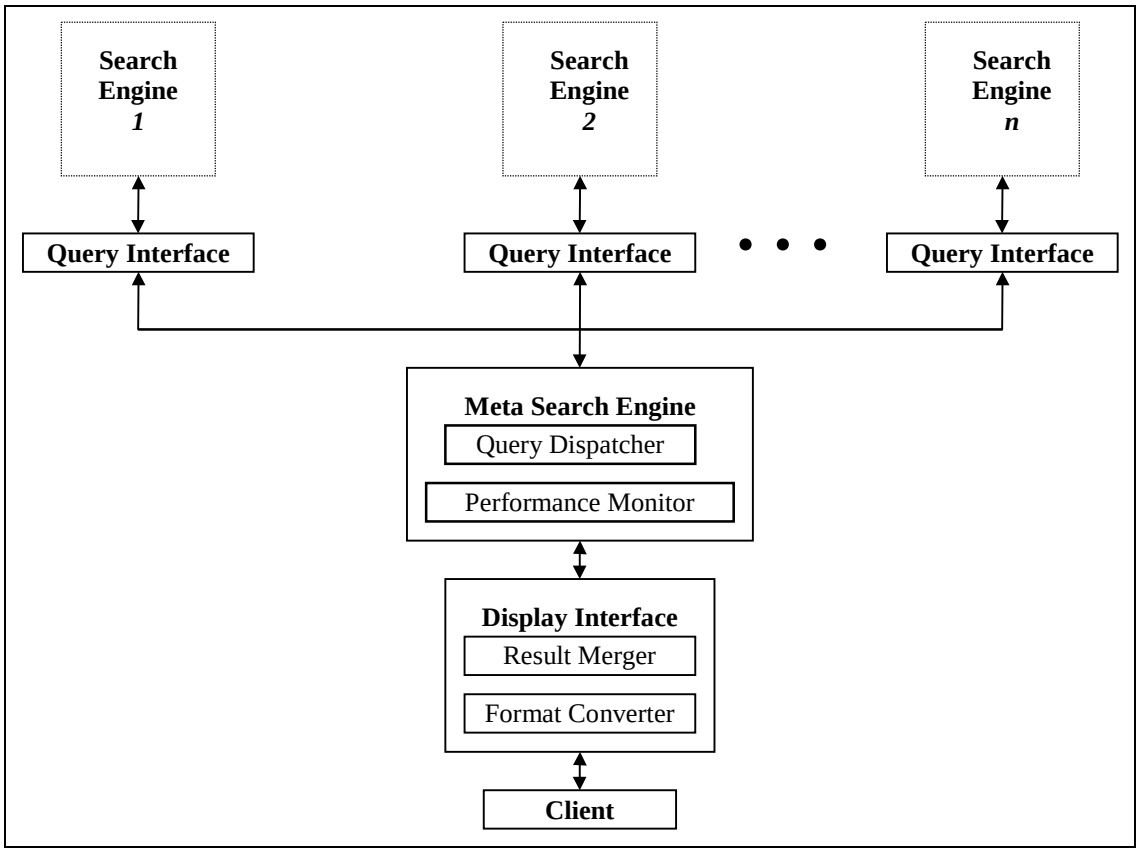

Figure 1: Basic components of a meta-search engine

Other automated Web meta-searchers are Dogpile, Metafind and Metasearch. These systems basically dispatch queries to each one of their search engines that they target and present the returned documents to the user in a uniform manner. Many manual query dispatch search engines are also available on the Web. Tools such as All-in-One, CUSI, search.com, Infi-Net's META search and InterNIC are essentially pages full of forms to sending queries to a number of different search engines. The selection process is entirely up to the user – they must type their query into a separate form for each query submission. Only one search engine is activated at a time, and the results appear in the native format of whichever search engine produced them.

The ProFusion system [10] is a Web meta-search engine that supports both manual and automatic query dispatch. In automatic query dispatch, ProFusion analyzes the incoming queries, categorizes them, and automatically picks the best search engines for the query based on a priori knowledge (confidence factors) which represents the suitability of each search engine for each category. It uses these confidence factors to merge the search results into a re-weight list of the returned documents, removes duplicates and, optionally, broken links and presents the final rank-ordered list to the user. ProFusion's performance has been compared to the individual search engines and other meta searchers, demonstrating its ability to retrieve more relevant information and present fewer duplicate pages.

3. METASEEK

MetaSEEk is an integrated search engine, which serves as a common gateway, linking users to multiple image search engines. It includes three main components as shown in Figure 1. The query interface component accepts search queries from the user and translates them to the specific query interfaces used by each target search engine. The dispatching component decides which search engines the query should be sent to. The display component merges the results and ranks them for displaying. MetaSEEk evaluates the performance of each query method on a search engine for future queries based on the user's feedback.

Queries can be submitted to MetaSEEk at http://www.ctr.columbia.edu/MetaSEEk. The underlying system is implemented in C and currently runs on a HP platform. MetaSEEk uses socket programming for opening ports to send the queries to the individual target search engines and to download their results. HTTP commands are sent to the remote search engines in a similar manner to web browsers such as Netscape and Mosaic.

3.1. Content based image query

There are several methods, which may be used to retrieve images based on their visual contents. Several systems use visual features such as texture, color, shape, and structure [3,4,5]. For example, texture can describe the coarseness, contrast, roughness, and presence/absence of directionality of each image. Another method may be based on the amounts of different colors in each image. The color amounts can either be used to describe the entire color content of each image or they can describe the color amounts in local regions of the image. These methods can be used separately or can be combined in calculating the similarity measures for the content-based image query. Different search engines use various methods and support alternate combinations. They also use different algorithms for calculating the similarity measures and their distances.

3.2. The query interface component

MetaSEEk currently supports the following target search engines: VisualSEEk, WebSEEk, QBIC and Virage. In the current version of MetaSEEk, the user interface allows for browsing of random images retrieved from the remote search engines. The user can select a method for querying such as color and texture. These two methods can be selected individually or they can be combined. Another popular search technique used in the image search engines is search based on keywords. This kind of search is used in search engines for querying documents as well as images. Image search based on visual content usually returns a ranked list of images which have the highest similarity to the query input, which could be an example image or a visual sketch. Keyword-based search may be used to match images with particular subjects (e.g., nature and people) and narrow down the search scope.

MetaSEEk allows a search based on example images, URLs or keyword text. No all the search engines support all these options. The query dispatching component of MetaSEEk makes the decision on which search engines the queries should be sent to. This component is explained in detail in the next subsection.

The user can specify a value for the maximum waiting time which is used to prevent the query system from stalling if a target search engine happens to be down or unreachable. Figure 2 shows the user interface for the MetaSEEk search engine.

3.3. The query dispatching component

MetaSEEk queries the search engines that first, support the method of the query selected by the user (i.e. color and/or texture), and second, have high past performance scores. The performance scores are calculated every time a query is made and are based on the user's feedback. The calculation of the performance scores is explained in section 3.5.

MetaSEEk keeps track of the performance scores of the search engines with respect to each query on every image. These performance scores are indexed in the database and will be used to select the target search engines for each new query. MetaSEEk also stores the visual feature vector for each queried image in case the queried image is not already in the database. In this case, when the user issues a new query, a set of old queries with the most similar feature vectors with that of the new query will be used. The queried images' performance scores will be used to select the search engines for the new query. This approach basically finds the best query examples from the past and follows its route to selecting the remote search engines, which have done well for that past queries.

Figure 2: The query interface of MetaSEEk

Once a query is made by selecting, or specifying the URL address of an image, MetaSEEk searches its database for the specified image to find the best search options based on the previous queries on that image. A search option is a query method on a specific search engine. For example, a query to the VisualSEEk search engine based on texture is considered as a search option. If the image is not found in the database, the best search options will be decided using the previous queries of the most similar images. The most similar images are selected from the database by comparing the locally calculated feature vectors for color and texture. The Tamura algorithm [11] is used for computing the texture feature vectors. For color, the feature vectors are calculated using the color histogram algorithm. The distance between two feature vectors is calculated using the Euclidean distance. These feature vectors will then be added to the database for future queries. Note that other feature vectors can be used if necessary. Our performance monitoring and search engine recommendation framework is general and can accommodate different feature vectors.

The database used for monitoring the performance of the different search options contains the following information:

- The image name, which includes the complete URL address of an image located on a local or remote site.
- The locally calculated feature vectors for color and texture.
- A list of the performance scores for all the possible search options. For example: 3:2, 1:5, 0:1, -1:3, -1:4, -NA:0, NA:6, NA:7, NA:8. In descending order, the first integers correspond to the scores and the integers proceeded by the ':' signs correspond to the indices of the search options. For example, these indices may correspond to search options shown in Table 1.

Index	Option
0	QBIC texture
1	QBIC color percentages
2	QBIC color layout
3	VisualSEEk color percentages
4	VisualSEEk color layout
5	VisualSEEk texture
6	Virage color
7	Virage composition
8	Virage texture

Table 1: MetaSEEk indexing example

As can be seen from the above example, the indices are sorted based on their scores. A score of NA means that the specific query option corresponding to that index is not available on the search engine and that it can not be used for making a query. Since some search engines do not have support for a search based on a URL address of an image outside their own database, they can only be queried when an image from their own database is selected. Therefore, for all the images outside of their databases, the scores corresponding to query options that relate to other search engines will be set to NA.

3.4. The display component

Once the results are retrieved from each individual search engine, they need to be merged and displayed to the user. These results are ranked in the order of the closest to the farthest match. MetaSEEk performs additional ranking of the returned images by using the query images' performance scores. The result images returned by each query option are interleaved before displaying them to the user. The performance scores will determine the order of the displayed images and the number of images in each interleaved group for each query's results. For example, if the images returned by two query options have performance scores of 2 and 1, MetaSEEk will continue to display 2 images from the query option having a score of 2, and 1 image from the query option with a score of 1 until all the returned images are displayed to the user. MetaSEEk will also do some cleaning up by removing duplicates if necessary.

The use of the above specific merging algorithm is not meant to replace the ranking algorithms used in each remote search engine. Instead, it is used to cope with the heterogeneity among different algorithms used in different search engines.

Unlike the consistent distance metrics used in text search engines, each visual search engine uses different algorithms and metrics. In order to evaluate the similarities of the returned images (from different engines) with the query input, a common set of features could be computed for all the returned images and be compared with the query input's. This option, however, would be too costly for the network since most of the images would have to be downloaded in order for the feature vectors to be computed. The selected merging method ignores the actual distances between the images, therefore it is possible for an image with a lower similarity measure to the input query to be displayed before an image with a higher similarity.

3.5. Performance monitoring

The dispatching component of the meta-search engine determines which query method on a remote search engine should be executed. As mentioned earlier, the decision is made based on the performance metrics calculated from the past queries and whether or not the target engine supports the specified method of the query. The performance metric is a signed integer where a positive number indicates a good performance and a negative number corresponds to a poor performance. The performance metrics of the query image is modified every time the user sends a query. A visit of an image increments by one the performance metric of the search engine that returns the visited image. If an image is not selected (i.e. no visit) the performance score remains unchanged. The user can also specify if he/she likes or dislikes a particular image, which will in turn increment, or decrement the performance metric of the corresponding search engines. These modifications of the database are shown on Table 2.

Event	Score
Visit	+1
No visit	0
Like	+1
Dislike	-1

Table 2: The assigned values for the performance metrics

As mentioned earlier, MetaSEEk removes all the duplicate images returned by different search options on different search engines. If the user, however, clicks on the like/dislike button of an image having a duplicate, MetaSEEk will increment/decrement the performance scores for all the search options that had returned that image even though the duplicates will not be all displayed to the user.

4. EXPERIMENTS AND EVALUATION

This first version of MetaSEEk has been developed with the primary objective of investigating whether or not our recommendations of search engines for incoming queries were appropriate. When a query is submitted to MetaSEEk, the system only queries those search options that have provided the most desirable results in the past, according to the information in the performance database.

A set of experiments was conducted to evaluate the performance of MetaSEEk: the reliability of the selection and ranking of the remote search engines for different user queries. These experiments were intended to provide a quantitative measure of how useful and precise the performance. The goal is to find the best images that the user is searching for as quickly as possible in a small number of queries made by the user. Therefore, one experiment would involve taking note of the number of queries until a desired image or set of images is found as the size of the performance database grows in time. Another experiment would be to note what fraction of the returned images from the search engines the user likes/dislikes as the size of the performance database grows.

We collected data and compared two different systems: the MetaSEEk and a base line meta-search engine. The MetaSEEk prototype includes all the advanced features mentioned in section 3. On the other hand, the base line meta-seach engine does not use any past performance of the different search engines in selecting the target search engines. For each incoming query, it randomly selects a set of search options and queries them. This system does not distinguish between new and past queries because it does not keep or use any past performance scores.

Since MetaSEEk keeps track of the performance results for every query, the number of queries made by the user for finding an image is expected to decrease as the size of the database grows and the system accumulates more knowledge. However, if the performance database is not used, the number of queries is expected to change randomly. The number of images the user likes is expected to grow and the number of disliked images is expected to decrease as the performance database grows in time. If the performance results are not used, these numbers will change randomly and are not expected to follow any particular patterns.

The results of these experiments are shown on the graphs in Figure 3 and Figure 4 for the two different systems: MetaSEEk and the base line meta-search engine. All the necessary data for performing these experiments was taken in a few-days of duration. As can be seen from these graphs, recommendations based on past performance scores have made a considerable amount of improvement in MetaSEEk. The number of queries made by the user till the desired image is found tends to decrease as the knowledge of the system increases. Differently, random variations are observed in the case of the base line meta-search engine. The statistics for the like/dislike experiment do not show any special pattern or improvement for the MetaSEEk system. That can be caused by limited duration of our experiments. More thorough knowledge may be accumulated over a longer period of experiments. Currently, we are continuing conducting more of such experiments and establish a larger knowledge base.

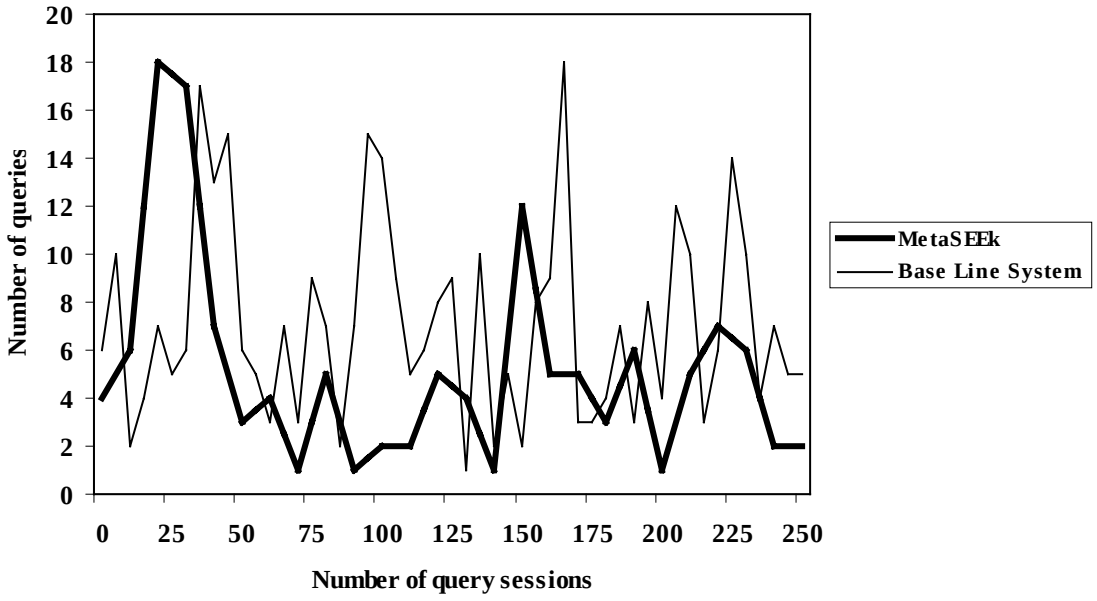

Figure 3: Number of queries till wanted image is found for MetaSEEk and base line systems

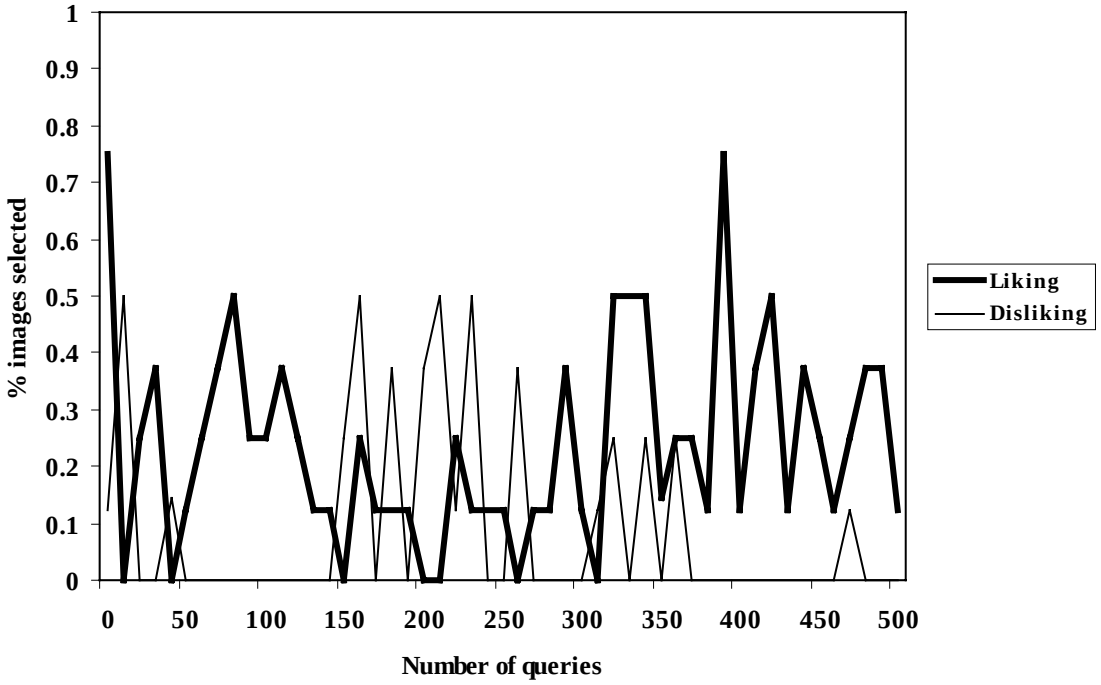

Figure 4: a) Like/Dislike trend for MetaSEEk prototype

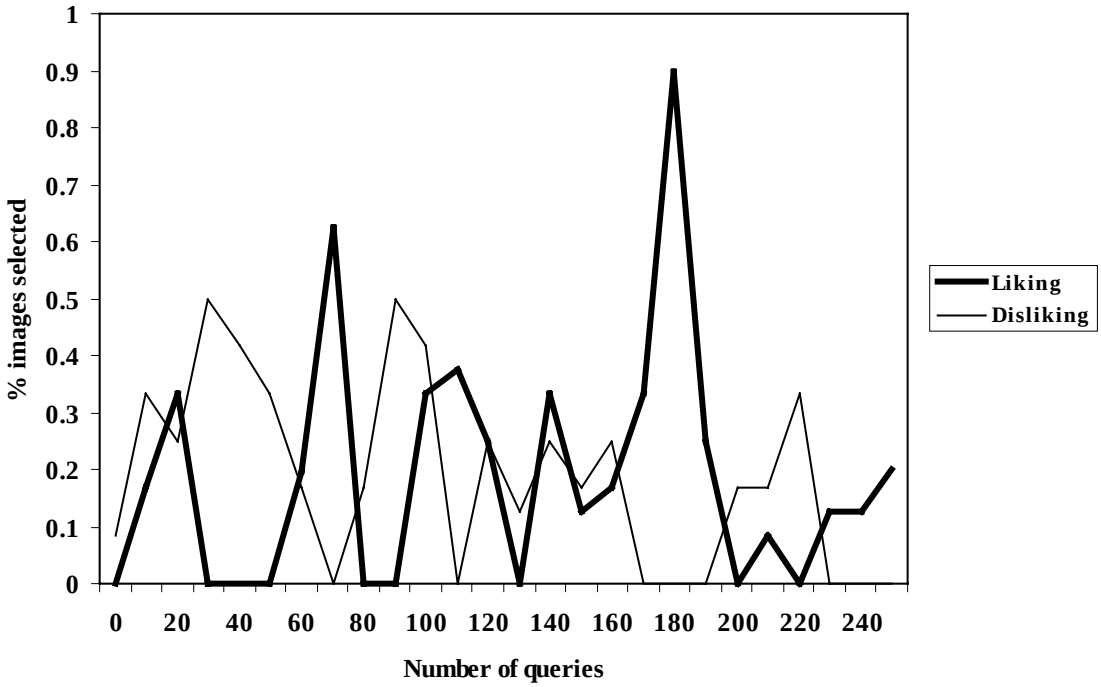

Figure 4: b) Like/Dislike trend for base line prototype

5. CONCLUSION AND OPEN ISSUES

The proliferation of text search engines on the Web has motivated the recent research in meta-search engines. In the same trend, impelled by the growing wealth of VIR systems on the WWW, we have developed a prototype meta image search engine, MetaSEEk, to explore the issues involved in querying large, distributed, on-line visual information system sources. Our goal is to investigate novel techniques for enhancing interoperability of distributed VIR systems, rather than ranking the performance of individual systems.

MetaSEEk uses performance scores to recommend remote search engines and query methods to send the query to. These performance scores are constructed by accumulating their successes and failures in the past queries. When a query is submitted to MetaSEEk, the most relevant query methods and search engines are selected by weighing their performance records to predict the ones that are likely to produce relevant results. If MetaSEEk receives a new query not encountered before, the system will match it to the content of the database in order to obtain a list of the most similar past queries. Averaging their performance indexes, MetaSEEk is able to recommend suitable search engines. MetaSEEk updates the performance ranking with new user feedback.

As the experiments show, the performance of a meta image search system can be greatly improved by implementing an intelligent integrated searching engine. This improvement involves the speed at which the desired images are found and how likely the user is satisfied with the results.

For future versions of MetaSEEk, we are considering an alternative approach to relate the new queries to the past ones: to cluster the past queries into several classes each of which share similar visual features. Performance scores are averaged over each cluster of past queries. Each new query will be classified to the closest class, whose performance records will be used to recommend the remote search engines. Users may intervene with the selection process by providing some search plans, which will override the recommendation of the meta-search engine. The performance scores may be accumulated over all past queries or a limited recent period. Because of the transient status of the networks, accumulation over a short period might be more desirable. The period interval may depend on the characteristics of the networks (e.g., Internet vs. Intranet) or user preference.

We also plan to investigate new techniques for merging the results retrieved from each individual search engine. The implemented merging method ignores the actual distances between images; therefore more suitable images may be displayed as having less similarities. The information available in the database and user feedback may be used to refine the way the result images are displayed to the user.

The MetaSEEk meta-search engine can be further improved by adding capabilities such as a support for customized search. QBIC and VisualSEEk allow the user to customize the search by graphically/manually specifying visual sketches as query input. The customized search on these two systems is supported for color percentages and color layout to allow the user to manually specify the amounts of different colors, or to specify different color locations, respectively. A customized search would require additional user interface programming and is not currently supported by MetaSEEk.

As MetaSEEk allows image retrieval by entering keywords, we are also considering the possibility of keeping another database with search engine performance records for keyword search. Our goal would still be to investigate novel and efficient techniques for enhancing the user's interoperability with distributed VIR systems.

6. REFERENCES

[1] Daniel Dreilinger and Adele E. Howe, "Experiences with Selecting Search Engines Using Meta-search", to appear in *ACM Transactions of Information Systems*, 1997.

[2] Shih-Fu Chang, John R. Smith, Mandis Beigi and Ana B. Benitez, "Visual Information Retrieval from Large Distributed On-line Repositories", to appear in *Communications of ACM*, 1997.

[3] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer Magazine*, Vol.28, No.9, pp. 23-32, 1995.

[4] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain and C. Shu, "Virage image search engine: an open framework for image management", *Symposium on Electronic Imaging: Science and Technology --Storage & Retrieval for Image and Video Databases IV*, **IS&T/SPIE'96**, Feb. 1996.

[5] J. R. Smith and S.-F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System" *ACM Multimedia Conference*, Boston, MA, Nov. 1996. (WWW: http://www.ctr.columbia.edu/VisualSEEk, ftp: //ftp.ctr.columbia.edu/CTR-Research/advent/public/papers/96/smith96f.ps)

[6] Luis Gavarno, Hector Garcia-Molina, and Anthony Tomasic, "The Effectiveness of Gloss for the Text Database Discovery Problems", *Proceedings of the ACM*, **SIGMOD'94**, Minneapolis, May 1994.

[7] Mic Bowman, Peter B. Danzig, Udi Manber, Michael F. Schwartz, Darren R. Hardy, and Duane P. Wessels. "Harvest: A scalable, customizable discovery and access system", *Technical report*, University of Colorado-Boulder, 1995.

[8] B. Kahle and A. Medlar, "An Information System for Corporate Users: Wide Area Information Server", ConneXions – The Interoperability Report, Vol. 5, No. 11, pp. 2-9, Nov 1991.

[9] Erik Selberg and Oren Etsioni, "Multi-service search and comparison using the MetaCrawler", *Proceedings of the 4th International World Wide Web Conference*, Dec 1995.

[10] Center for Research, Inc., University of Kansas. ProFusion meta-search engine.
http://www.designlab.ukans.edu/profusion/

[11] H. Tamura and S. Mori and T. Yamawaki, "Textural Features Corresponding to Visual Perception", *IEEE Trans. Syst., Man, Cybern*, Vol. 8, No. 6, 1978.