USING RELEVANCE FEEDBACK IN CONTENT-BASED IMAGE METASEARCH

ANA B. BENITEZ, MANDIS BEIGI, AND SHIH-FU CHANG
Columbia University

MetaSeek is an image metasearch engine developed to explore the query of large, distributed, online visual information systems. The current implementation integrates user feedback into a performance-ranking mechanism.

The proliferation of online documents and search engines in the past few years has motivated the development of metasearch engines, which act as gateways that link users automatically and transparently to multiple and maybe competing search engines. Current metasearch engines use a variety of techniques (see the sidebar, "Related Work"), though all of them work with the fundamental searching unit of text. Automated visual information retrieval (VIR) systems work with image features—such as colors, textures, and shapes—in combination with text and other related information to query the increasing store of image data available on the Web.[1-4] New VIR systems pose special problems of heterogeneity and performance that motivate the development of metasearch engines in this domain.

We have developed a prototype content-based metasearch engine for images, called MetaSeek, to investigate the issues involved with efficiently querying large, distributed online visual information sources.[5-6] MetaSeek selects and queries the target image search engines according to their success under similar query conditions in previous searches. The current implementation keeps track of each target engine's performance by integrating user feedback for each visual query into a performance database.

We begin this article with a review of the issues in content-based visual query, then describe the current MetaSeek implementation. We present the results of experiments that evaluated the implementation in comparison to a previous version of the system and a baseline engine that randomly selects the individual search engines to query. We conclude by summarizing open issues for future research.

1089-7801/98/$10.00 ©1998 IEEE

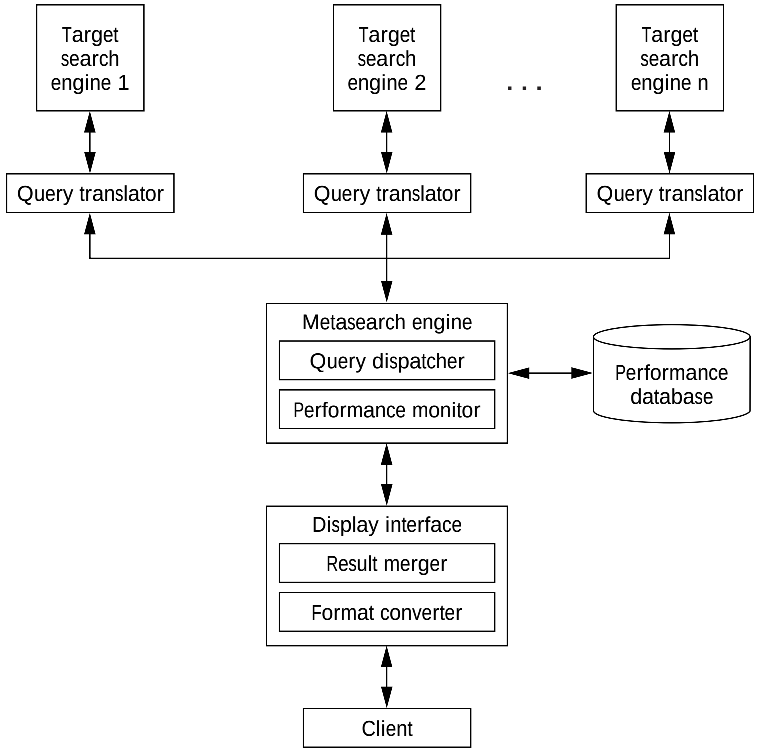


Figure 1. MetaSeek includes three basic components common to most metasearch engines: the query dispatcher, query translator, and display interface component.

CONTENT-BASED VISUAL QUERY

While text document retrieval is based on the words composing each document, there is no such specific searching unit for visual information; additional processing is required to obtain adequate and comparable representations of visual content features such as color, texture, and shape, among others. In this framework, image data can be viewed as arrays of feature vectors that constitute a first-level effort to characterize visual content. A distance function can then be used to compute the closeness of feature vectors—in other words, to provide an approximate measure of the visual similarity among images.

Content-based visual queries are usually initiated by selecting a query image from an initial set of images returned in a random or keyword search or, alternatively, by inputting a URL to an external image. Given the input image, the search engine computes the image's visual features (if necessary) and uses them to retrieve matched or most similar images from its database. Keyword-based search can be used to match images with particular subjects (for example, nature or people) and thus narrow the search scope.

Techniques such as clustering and indexing have been developed to efficiently handle visual query retrieval in large visual databases.[7-8] In clustering, the images in a database are categorized on the basis of content; the decision criteria can follow general classification methods or special learning techniques. Indexing techniques sort the visual data on the database or other storage device based on their visual features. Special indexing algorithms have been proposed for content-based image retrieval. They can be applied to the image feature vectors in visual databases to increase access efficiency.

Each VIR system supports a specific (and often proprietary) set of visual features with particular feature combinations, extraction methods, distance metrics, clustering strategies, and indexing algorithms. Some of these visual repositories allow queries to combine low-level visual features with text and other related information. MetaSeek provides a unique integrating tool in the heterogeneous environment of online VIR systems.

METASEEK

MetaSeek is a content-based metasearch engine for images on the Web. It automatically links users to multiple image search engines for online visual information sources. The overall architecture of MetaSeek is shown in Figure 1. The three main components of the system are quite standard for metasearch engines; they include the query dispatcher, query translator, and display interface component.

- Upon receiving a query, the dispatcher selects the target search engines to be queried by consulting a performance database at the MetaSeek site. This database contains performance scores of past query successes and failures for each supported search option.
- The query translators then translate the user query to suitable scripts conforming to the interfaces of the selected search engines.

- Finally, the display interface component merges and ranks the results from each search option, and presents them to the user.

MetaSeek evaluates the quality of the results returned by each search option based on user feedback. This information is used to modify the corresponding entries in a performance database.

A fundamental principle in the design of MetaSeek was to use Web resources as efficiently as possible while still providing quality results. This principle underlies many implementation decisions, such as querying only search engines that provided good results under similar query conditions in the past, and avoiding downloading images whenever there was a feasible alternative.

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 Hewlett-Packard platform. MetaSeek uses socket programming to communicate with the individual target search engines in a manner transparent to the user. HTTP commands are sent to the remote search engines to pose queries and download their results in a manner similar to that of Web browsers like Netscape and Mosaic.

The remainder of this section describes the MetaSeek query interface, the query dispatcher and performance database, and the display interface component.

Query Interface

MetaSeek currently supports the following four online search engines:

RELATED WORK

Metasearch tools can be classified first according to whether they have automatic or manual query dispatchers. Automated dispatchers decide the search engines to be queried with no direct control from the users. They may query either all supported target search engines or a select group for each specific query based on some sort of knowledge. In manual metasearch engines, the selection process is entirely up to the user. Most manual metasearch engines activate only one search engine at a time and display the resulting data in the original format of the search engine that produced it.

Many approaches have been proposed for metasearching. The following overview highlights those that have contributed most to our work on MetaSeek.

The Gloss (Glossary-of-Servers Server) project[1] uses a meta-index to estimate which databases are potentially most useful for a given query. The meta-index is constructed by integrating the indexes of target databases. For each database and each word, the number of documents containing the word is included in the meta-index. This approach has two main drawbacks: first, it requires each search engine to cooperate with the metasearcher by supplying up-to-date indexing information, and second, as the number of databases increases, the complexity may become prohibitive.

The SavvySearch meta-search tool[2] employs a meta-index approach for selecting relevant search engines based on the terms in a user's query. Previous experience about query successes and failures is tracked to enhance selection quality. This system combines automatic and manual query dispatching. When a query is submitted, SavvySearch proposes a search plan of target search engines into different ordered steps that the user can either regard or disregard. Their experimental findings 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 questions about the overall efficiency of the system.

The ProFusion system[3] is a Web metasearch 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) that represent the suitability of each search engine for each category. It uses these confidence factors to merge the search results into a re-weighted 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 individual search engines and other metasearchers, demonstrating its ability to retrieve more relevant information and present fewer duplicate pages.

REFERENCES

1. L. Gravano, H. Garcia-Molina, and A. Tomasic, "The Effectiveness of Gloss for the Text Database Discovery Problems," *Proc. ACM SIGMOD*, ACM, New York, 1994; available at http://elib.stanford.edu/TR-STAN:CS-TN-93-2.
2. D. Dreilinger and A.E. Howe, "Experiences with Selecting Search Engines Using Metasearch," *ACM Trans. Information Systems*, Vol. 15, No. 3, July 1997, pp. 195-222.
3. Center for Research, Inc., University of Kansas, ProFusion metasearch engine, at http://profusion.ittc.ukans.edu/profusion/.

- VisualSeek,[1]
- WebSeek,[2]
- QBIC,[3] and
- Virage.[4]

Each of these systems presents individual functionalities and limitations.

VisualSeek, QBIC, and Virage accept example images and match them to images in the database according to visual features such as color and texture. VisualSeek and QBIC support customized searches that allow users to interactively draw visual sketches or specify external images, whereas Virage allows users to weight the importance of each feature in the search. QBIC also supports image retrieval based on keywords.

WebSeek, on the other hand, is a semiautomatic image and video search and cataloging engine. It supports text-based and content-based searching using color histogram only. MetaSeek currently supports only the text-based searching capabilities of WebSeek.

The current MetaSeek user interface supports random browsing and image retrieval based on visual content and keywords. To query based on visual content, the user can either select an example image from one of the supported databases or input the URL to an external image. MetaSeek can search for matching images based on two visual features—color and texture—which can be either selected individually or combined.

Figure 2 illustrates the user interface for the MetaSeek search engine. Because not all the target search engines have the same query options, MetaSeek's query dispatcher is programmed to know each database's searching capabilities and to send queries only to appropriate search engines (the dispatch mechanism is explained in the next section). MetaSeek may query multiple search options simultaneously from the same target search engine.

The user can specify the number of search options to be searched simultaneously, the maximum waiting time, and the category of interest by using pull-down menus. The number of queries to be sent to the individual search engines can be adjusted to account for network load. During periods of low network traffic, for example, the user can increase the number of concurrent queries. The maximum waiting time prevents the query system from stalling if a search engine happens to be down, unreachable, or experiencing significant delays.

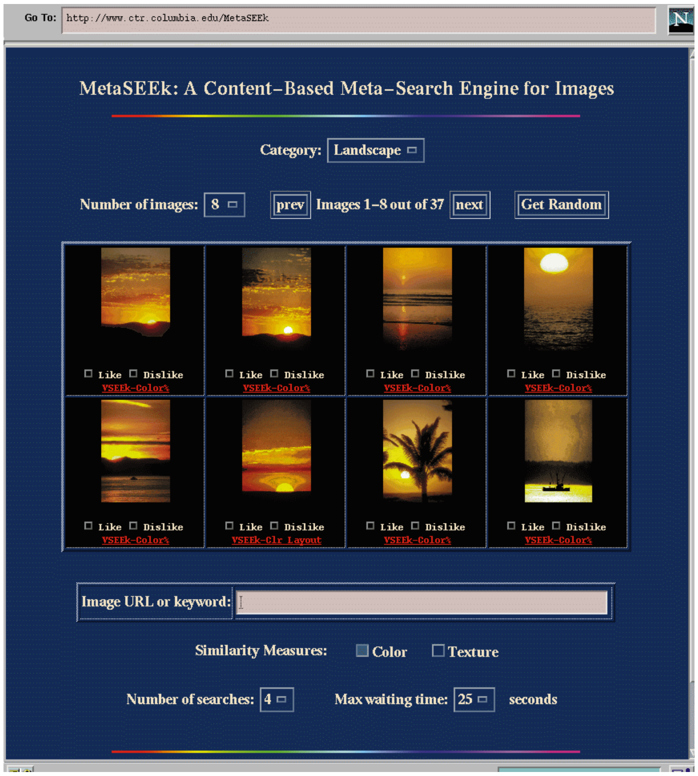
In selecting a category other than "All" (for example, Nature or Animals), the user is expected to submit input images belonging to that category. The query dispatcher processes this information to improve the recommendation mechanism for target search engines.

Query Dispatcher and Performance Database

Upon receiving a query, the dispatcher selects the target search engines and search options to be queried. A *search option* is defined as a query method on a specific search engine—for example, a texture-based query of the VisualSeek search engine. The dispatcher makes the selection based, first, on the type of query submitted to the metasearch engine. For example, if the user requests random image samples or a keyword query, the system simply poses queries to the search engines that allow these actions (QBIC, Virage, and VisualSeek support random samples; QBIC and WebSeek support keyword queries).

For content-based visual queries, however, the selection of target search engines is based on the performance database that contains scores indicating how each search option performed in the past on every query image. In MetaSeek, a content-based visual query is specified by a query image, a group of features, and a semantic category. Upon receiving the query, MetaSeek searches the performance database and retrieves the query image's performance scores. The query dispatcher will select the search options with the highest scores that agree with the user feature and semantic category selections.

How about new images that have not been queried before and therefore have no performance scores in the database. The simplest solution would be to randomly select the search options to be queried. Instead, MetaSeek recommends remote search options by relating new queries to past ones for which performance data exists. Query images in the performance database are clustered into several classes based on visual content. The system maintains a cluster structure for each feature query: color, texture, and combination of both. When the user issues a query with a new query image, the system downloads the image and matches it to the corresponding clustering structure to obtain a list of the most similar clusters. Images from the few closest clusters are selected and presented to the user. The dispatcher can then recommend suitable search engines based on the average performance scores of the cluster selected by the user. Finally, the new image will be added to the performance database for future queries.



Go To: http://www.ctr.columbia.edu/MetaSEEk

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

Category: Landscape

Number of images: 8 prev Images 1–8 out of 37 next Get Random

□ Like □ Dislike VSEEk-Color%	□ Like □ Dislike VSEEk-Color%	□ Like □ Dislike VSEEk-Color%	□ Like □ Dislike VSEEk-Color%
□ Like □ Dislike VSEEk-Color%	□ Like □ Dislike VSEEk-Clr Layout	□ Like □ Dislike VSEEk-Color%	□ Like □ Dislike VSEEk-Color%

Image URL or keyword:

Similarity Measures: ■ Color □ Texture

Number of searches: 4 Max waiting time: 25 seconds

Figure 2. The MetaSeek query interface. In this case, eight preview images are displayed from a CBV query that generated 37 total results. The URL in the upper left corner identifies the query image.

Many approaches have been proposed for clustering visual data to support efficient image retrieval. We decided on the K-means clustering algorithm because of its simplicity and reduced computation requirements.[8] The clustering algorithm is not executed every time a new image is added to the database, but every 10 new images. MetaSeek uses the Tamura algorithm for computing the texture feature vector for an image and the color histogram algorithm for the color feature vector. The distance between two feature vectors is calculated using the Euclidean distance.

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.

Semantic categorization. We use semantic categorization to constrain the search scope. Visual databases usually contain special types of images. For example, we have observed that the QBIC database currently includes many images of people, whereas the Virage's database has more images in other categories. If a user is interested in finding the picture of a baby, QBIC is more likely to yield appropriate results in less time.

MetaSeek offers the possibility of searching within a specific category and maintains separate performance databases and clustering structures for each category. To activate this feature, users select a category from a pulldown menu. The current version of MetaSeek includes 10 categories: animals, art, buildings, flowers, geography, landscapes, nature, objects, people, and transport.

Database structure. The metasearch database contains the feature vectors (color and texture), the performance scores, the clustering class, and the semantic category for all images that have been queried on MetaSeek. All this information is needed by the dispatcher to recommend suitable search engines for incoming queries. The database is organized into a hierarchical structure. The images are first classified according to their semantic meaning (for example, animals), the category selected by the user. The K-means clustering algorithm is then used to cluster the images in each semantic category into classes on the basis of color, texture, or both features.

Figure 3 on the next page presents the conceptual structure of the database. At the lowest level, the image entries contain the information shown in Table 1. The image name is the complete URL address of the image, which may be located on a local or remote site. The feature vectors for color and texture are calculated using the color histogram and Tamura algorithms, respectively. The K-means clustering algorithm uses these feature vectors in organizing the images based on visual content.

The query dispatcher uses the performance scores to decide target search engines; the scores are updated according to user feedback every time the image is queried.

A score of NA means that the search option cannot accept the image with that score as query input. For example, Virage does not accept external images as query input; therefore, it can only be queried when an image from its own database is selected. Since the image in Table 1 is from an external URL, the scores for the Virage search engine are set to NA.

Performance monitoring. Because we expect the performance of individual search engines to change over time as they improve their algorithms and add new images to their databases, we decided to construct the performance metrics from accumulated user feedback. For every image that has been queried, the performance database keeps a vector of performance indexes, one index for each search option supported by the individual search engines (as shown in Table 1). The performance of each search option 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 a result image is updated every time the user submits that image as the query input of a content-based visual query. The selection of a result image increments the performance metric by one of the search options that returned the visited image. If an image is not selected, the performance score remains unchanged. Users can also specify whether they like or dislike a particular result image, which will in turn increment or decrement the performance metric of the corresponding search engines. Table 2 lists these modifications of the database.

Bootstrapping the performance database. The feedback method of acquiring knowledge about search engine performance requires a continuous process of iterative user query sessions to accumulate results. The process will generate nontrivial entries in the database, but it must be seeded first through a training period.

One way to select target search engines during the training period is to use all search engines or randomly selected engines. However, this approach is likely to generate inefficient or unreliable training results. In view of this, we used an offline probing process to establish initial performance scores. The probing procedure is carried out prior to the training period. In other words, the initial probing process establishes the machine-generated performance scores to be used in the training period. The performance scores based on user feedback from the training period are used for later online operations, during which the performance database is continuously updated through user feedback.

Experimental findings in metasearch engines for text databases suggest that a performance index approach can be effective in making search engine selection decisions.[9] However, the potentially large amount of knowledge required to make these decisions raises questions about the overall efficiency of the system. Automatically probing the search engines with a set of sample images solves this problem. The probe provides some base performance knowledge to start the training of the metasearch engine. Once enough knowledge has been accumulated from training MetaSeek, the query dispatcher discards this “artificial” knowledge and bases its search engine selection decisions exclusively on accumulated training performance.

The strategy for probing the search engines and collecting initial performance data tries to emulate users’ judgment for result images as relevant or nonrelevant. First, 40 probe images are selected for each semantic category. Every probe image in this set is then posed to all search options in all search engines, and the results are downloaded for further analysis. For each result image within a visual distance α from the probe image, the performance

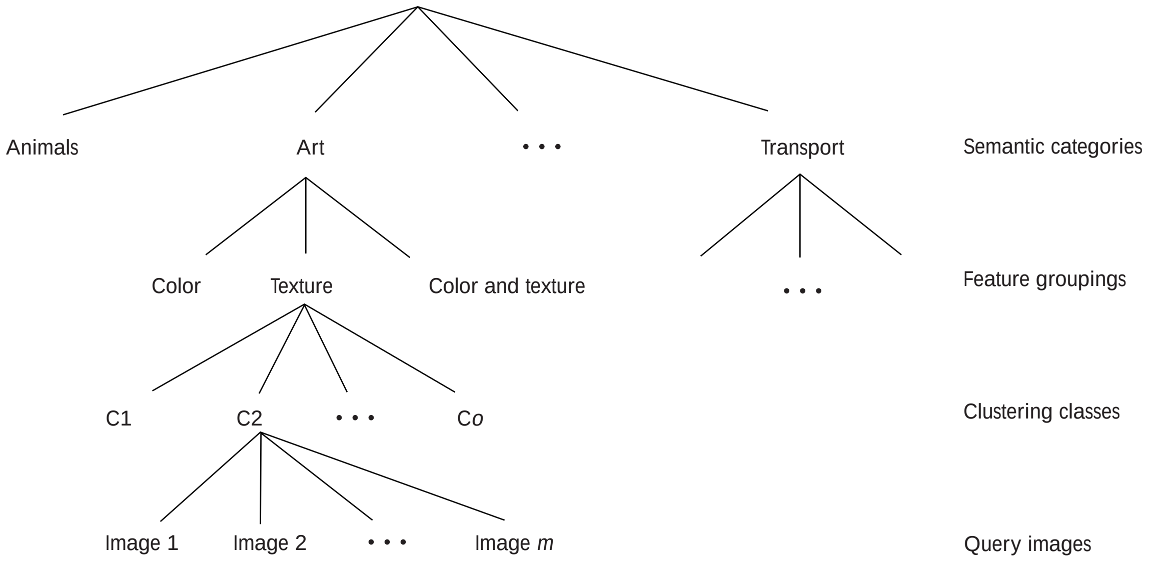


Figure 3. Conceptual structure of the metasearch database for MetaSeek.

score of the search option that returned the image will be increased. On the other hand, the performance score of the search option that returned the image will be decreased for each result image not within a visual distance δ from the sample image.

When the probing is finished, the clustering algorithm is applied to the images in the sample probe based on color, texture, and both features. We determine the thresholds α and δ through subjective evaluation of query results and their feature distances by an expert.

The strategy for training the metasearch engine begins with the selection of 20 training images for each semantic category. After posing each image in this set to MetaSeek, an expert user evaluates the results returned from each search option as relevant or nonrelevant. The set of probing images and the set of training images are completely independent.

Table 1. Example image entry in the MetaSeek database.

Image URL	http: / / www.world.com / hello.gif	
Color Feature Vector	(0.0000, 0.3456, 0.0456, ... 0.5677)	
Texture Feature Vector	(0.3456, 0.4545, 0.6767)	
Performance Score Vector:		
Search Option		
QBIC color percentages		7
QBIC color layout		9
QBIC texture		3
VisualSeek color percentages		–1
VisualSeek color layout		3
VisualSeek texture		6
Virage color		NA
Virage composition		NA
Virage texture		NA

Query Display Component

Once the results are retrieved from each individual search engine, the MetaSeek display component organizes and presents them to the user. This process depends on the type of action requested by the user: random samples, keyword search, or image search. The results from a random image or a keyword search are merged randomly and presented to the user. The order of these results is not important.

In content-based visual queries, the display mechanism is more elaborated. Image search based on visual content returns a ranked list of images starting with the image whose vectors most closely match those of the query image. MetaSeek performs additional ranking of these images by using the scores in the performance database. The result images returned by each query option are interleaved before displaying them to the user. The performance scores of the query image will determine the display order and the number of images in each interleaved group of the results from each search option. For example, suppose that images are retrieved from two search options with performance scores of 2 and 1. The display component will display two images from the search option having a score of 2, and one image from the search option with a score of 1 until all the returned images are displayed.

This merging algorithm is not meant to replace the ranking algorithms in the target search engines. Instead, it is used to cope with the heterogeneity among them. Unlike the consistent distance metrics used in text search engines, each visual search engine uses different algorithms and metrics. To evaluate the similarities of the returned images from different engines with the query input, a common set of features could be computed and compared locally in the metasearch engine. This option, however, would be too costly for the network since most of the images would have to be downloaded to compute the feature vectors. The merging method avoids this problem by simply rank interleaving the images and ignoring 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. (Note that MetaSeek shows only the icons of the result images. Full-resolution images are not required because new feature extraction processes are not performed.)

Table 2. Performance metrics assigned values.

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

EXPERIMENTS AND EVALUATION

We developed this version of MetaSeek with the primary objective of investigating whether or not our recommendations of search engines for incoming queries were appropriate. Another important objective was to determine the utility of semantic and clustering categorization for improving the metasearch engine's performance.

We conducted a set of experiments, selecting 12 images of the semantic category "Animals" as the set of target images. Beginning with random samples, we tried to find each target image by successively querying the metasearch engine based on visual content and by selecting one of the result images as the query input to the next iteration. In this process, we monitored the number of queries necessary to find each target image and the average recommendation precision of these queries. This experiment was performed twice to evaluate the improvement of the recommendation mechanism with experience. In other words, after finding all the target images, we tried to find them again following the same procedure.

The recommendation precision is a measure of the accuracy of the search option recommendation. It is calculated as the number of relevant recommended search options over the total number of recommended search options with respect to a given query. We calculated the precision of all queries by querying a maximum of four top-ranked search options simultaneously. Search options with a positive user judgment (number of liked images minus the number of disliked images) are considered to be relevant to the query.

We collected data and compared three different systems: the current version of MetaSeek, the previous version of the system,[5] and a baseline metasearch engine. The previous version of MetaSeek lacks the semantic and clustering categorization of the current system. When the previous version receives a query not encountered before, the dispatcher recommends target search engines according to the average performance score of the images in the database closest to the query image in terms of visual feature distance. The baseline metasearch engine does not use any past performance of the different search engines in selecting the target search engines. Upon receiving a query, it randomly selects a set of possible search options and queries them.

When a query is submitted to the current or previous version of MetaSeek, the system queries only those search options that provided the most desirable results in the past, according to the information in the performance database. For these two systems, the number of queries required to find an image is expected to decrease as the system accumulates more knowledge. However, in the baseline system, the number of queries is expected to change randomly.

The variation of precision should present a similar pattern with increasing values.

The experimental results are reported in Figures 4 and 5 for the three different systems. The search number indicates the target image that we were trying to locate. Each search number corresponds to the same image in all of the graphs. Table 3 shows the mean number of queries and precision for the first and second passes of the experiment, and the percentages of searches that produced better results—lower number of queries or higher precision—from the first to the second pass. For these experiments, both the current and previous version of MetaSeek were trained with the same set of training images. The set of training images and the set of target images did not contain common images.

The current MetaSeek offers better overall performance than the other two systems. The number of queries required to retrieve the desired image tends to decrease considerably as the knowledge of the system increases. The recommendation precision clearly tends to increase with time. On the other hand, the previous version of MetaSeek outperforms the baseline system with a higher mean precision and a lower mean value of the number of queries for finding an image.

The same experiments were conducted for a set of 12 target images of the semantic category "Landscape." The experimental results obtained were very similar to the ones presented in Figures 4 and 5.

FUTURE WORK

Ongoing work with MetaSeek addresses the research issues related to the query of large, distributed image repositories.

User Feedback and Performance Monitoring

The experimental results demonstrated that an image metasearch engine can increase its efficiency if the search engines and search options are selected according to their performance for different classes of queries as determined from user feedback. Given the importance of performance monitoring and user feedback in the metasearch engine, we are thinking about more sophisticated approaches to implement these two stages. In any case, poor results are guaranteed if the system does not count with the collaboration of the users.

Customized Search

The MetaSeek engine can be further improved by adding capabilities such as support for customized search. QBIC and VisualSeek allow the user to customize the search by manually specifying visual sketches as query input. The customized search on these two systems is supported for color percentages and color layout, which allow the user to specify different color amounts or different color locations, respectively.

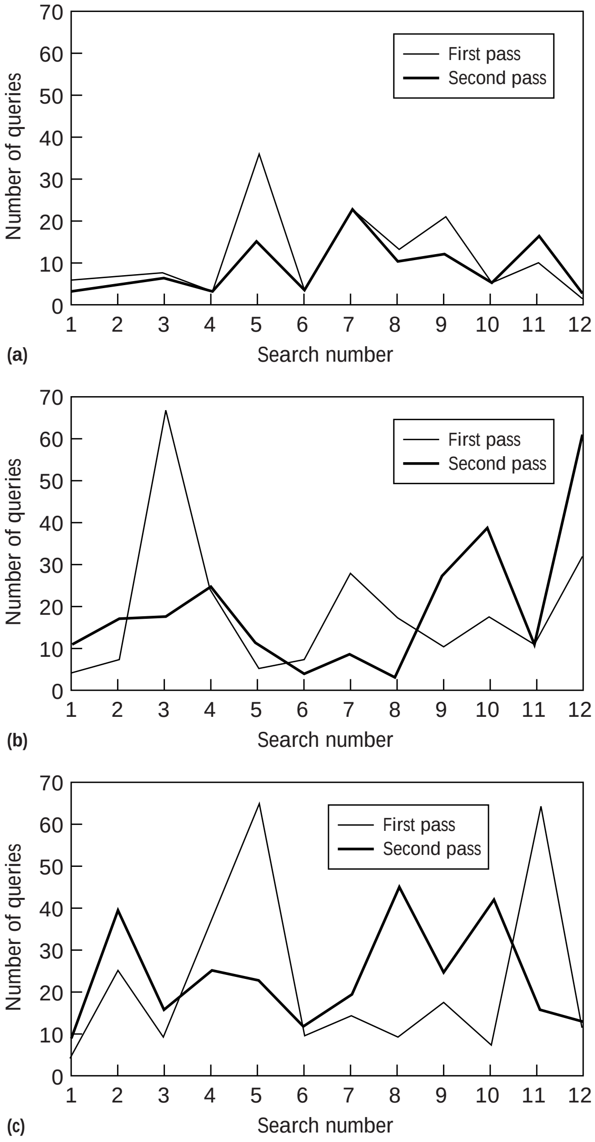


Figure 4. Number of queries to find target images: (a) for current MetaSeek implementation, (b) for previous MetaSeek implementation, and (c) for baseline metasearch engine.

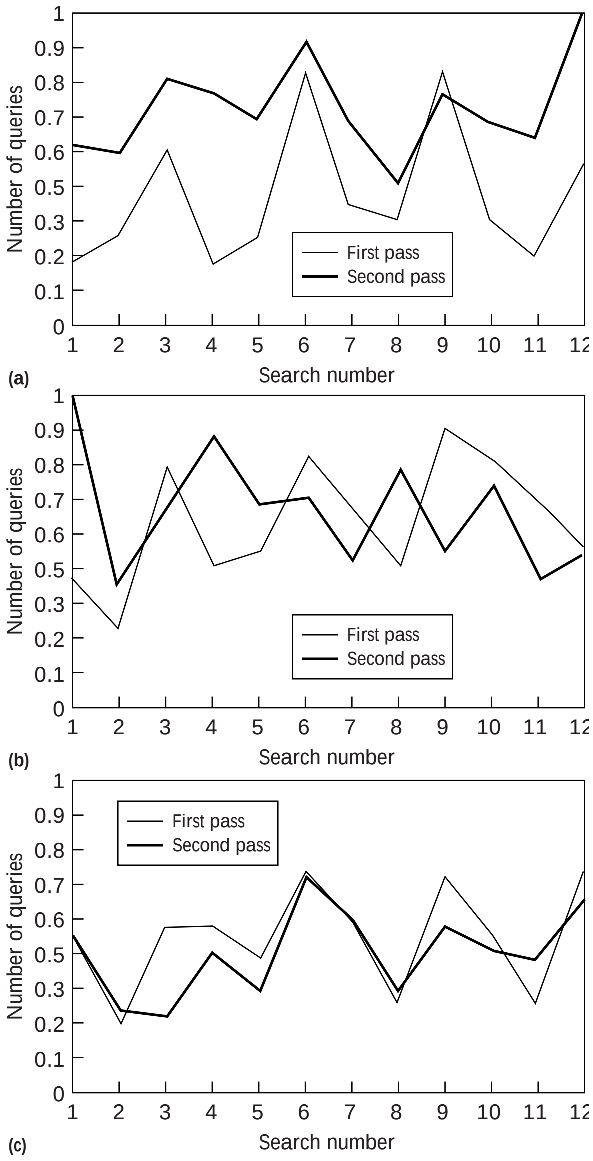


Figure 5. Precision trend (a) for current MetaSeek implementation, (b) for previous MetaSeek implementation, and (c) for baseline metasearch engine.

Table 3. Summary of the experimental results for category "Animals."

	Number of queries			Precision		
	First- pass mean	Second-pass mean	Improvement	First- pass mean	Second- pass mean	Improvement
Current MetaSeek implementation	11.08	8.58	67%	0.42	0.70	92%
Previous MetaSeek implementation	15.67	16.00	42%	0.58	0.62	50%
Baseline metasearch engine	17.17	18.17	25%	0.49	0.45	42%

A customized search would require additional user interface programming. The approach we have in mind will create an “artificial” image with the visual content the user specifies in the visual sketch. This image will then be submitted as an input image to visual content queries on search engines that support URL search.

Visual Features

In the current version of MetaSeek, the visual features for each image queried on the system are computed using color histogram and Tamura texture algorithms. These feature representations were selected for simplicity and execution efficiency rather than optimality. Furthermore, although almost a dozen representations have been proposed for texture alone, none has been accepted as the ultimate solution. In fact, it is very unlikely that these low-level features will ever offer such a solution because human perception of them is highly subjective.

For this reason, we want to explore a system that uses a larger set of features and adapts it automatically or semiautomatically to user preferences. The new system would recommend not only search engines but also feature representations based, probably, on training examples.

The rapid growth of image databases makes scalability an important property of VIR techniques. The K-means clustering algorithm currently used to categorize the collected images lacks scalability. We have tried to minimize this limitation by waiting every 10 new images before computing a new database structure, rather than applying it each time a new image is added. Nevertheless, we are considering a new categorization mechanism that would permit extending the database without changing its structure.

New Databases

New target search engines can be added to MetaSeek manually at this time. We would like to automate this procedure in future versions of the search engine. We are investigating a program that would allow willing online search engines to register with the metasearch engine. The registering program could also help register not-so-cooperative search engines.

In both cases, we believe that the interaction between a new target database and the metasearcher should not begin blindly. Knowledge is very expensive. In subscribing, a search engine could provide statistical data that reflected the content of its visual database. This data would not need to be resubmitted because of the adaptive capabilities of MetaSeek. Noncooperative search engines could be probed by a set of sample images.

Learning Algorithm

Machine learning addresses the question of how to build computer programs that improve their performance at some task through experience. Machine-learning algorithms have proved to be very useful in a variety of application domains. A well-defined learning problem requires a well-specified task, performance metric, and source of training experience.

MetaSeek is a system designed to learn to recommend the most suitable search engines to incoming user queries. It has improved its performance—as measured by the retrieval of better search results in less time—through experience obtained from user feedback. Clearly, MetaSeek can be classified as a machine-learning problem, and we are reviewing approaches to machine learning in related research areas.

CONCLUSION

Of the four online search engines currently supported, VisualSeek and WebSeek are academic visual information systems built at Columbia University, whereas QBIC and Virage are online demos of commercial systems. We neither know the design choices (feature extraction methods and clustering strategies, among others) nor have unrestricted access to the visual databases of the two latter systems. At this point, we feel that we need further control over the individual search engines to evaluate and compare the effectiveness of different approaches for some design choices we face. We are creating our own network of individual search engines to solve this problem.

Our network of search engines will emulate the heterogeneity of the real-world search engines as much as possible. In any event, the network will allow us to do more extensive experiments and to reach better conclusions about the suitability of different design options without inundating commercial VIR demos with our queries. Nonetheless, the most successful approaches will be integrated into the current version of MetaSeek. ■

REFERENCES

1. J.R. Smith and S. Chang, “VisualSeek: A Fully Automated Content-Based Image Query System,” *Proc. ACM Conf. Multimedia*, ACM Press, New York, 1996; also available at http://www.ctr.columbia.edu/VisualSeek/.

2. J.R. Smith and S. Chang, "An Image and Video Search Engine for the World-Wide Web." *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases V*, IS&T/SPIE97, 1997.
3. M. Flickner et al., "Query by Image and Video Content: The QBIC System," *Computer*, Vol. 28, No. 9, Sept. 1995, pp. 23-32; also available at http://wwwqbic.almaden.ibm.com/.
4. J. R. Bach et al., "Virage image search engine: an open framework for image management," *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases IV*, IS&T/SPIE96, 1996; available at http://www.virage.com/cgi-bin/query-e.
5. S.-F. Chang et al., "Visual Information Retrieval from Large Distributed On-line Repositories", *Comm. ACM*, Vol. 40, No. 12, Dec. 1997, pp. 63-71.
6. M. Beigi, A.B. Benitez, and S.-F. Chang, "MetaSeek: A Content-Based Metasearch Engine for Images," *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases VI*, IS&T/SPIE98, 1998; available at http://www.ee.columbia.edu/~ana/publications/.
7. C. Faloutsos, *Searching Multimedia Databases by Content*, Kluwer Academic Publishers, 1997.
8. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1997.
9. D. Dreilinger and A.E. Howe, "Experiences with Selecting Search Engines Using Metasearch," *ACM Trans. Information Systems*, Vol. 15, No. 3, July 1997, pp. 195-222.

Ana Belen Benitez is a PhD student at Columbia University. Her research interests are in content-based visual query and visual distributed databases. She received a degree in telecommunications engineering from the Higher Technical School of Telecommunication Engineering at the Polytechnic University of Catalonia (UPC), Barcelona, Spain, a master's degree from Columbia University in 1996. She is a student member of ACM.

Mandis Beigi is a PhD student at Columbia University. Her research interests are quality of service, service differentiation, and network monitoring. She received a bachelor's degree in electrical engineering from the State University of New York at Stony Brook in 1993, and an MS in the same field from Columbia University in 1995. She works at the IBM T.J. Watson Research Center.

Shih-Fu Chang is an associate professor in the Department of Electrical Engineering and New Media Technology Center in Columbia University. He is also active in Columbia's Digital Library Project. His current research interests include content-based visual query, networked video manipulation, video coding, and communication. He received a PhD in electrical engineering and computer science from the University of California at Berkeley in 1993. Chang holds two US patents in the area of visual search and compressed video processing. Among other honors, he is the recipient of an ONR Young Investigator award for 1998-2001.

Readers may contact the authors at Columbia Univ., Dept. of Electrical Engineering and New Media Technology Center, New York, NY 10027, email {ana, mandis, sfchang}@ee.columbia.edu.

IEEE INTERNET COMPUTING

AUTHOR GUIDELINES

IEEE Internet Computing actively solicits a variety of articles useful to software/hardware designers and developers working in the domain of Internet technology and applications.

IC articles are peer-reviewed for technical accuracy before they are accepted for publication. They are subsequently edited for presentation to a broad audience of computer professionals. Accordingly, all articles must include a general introduction (250-500 words) that explains the subject matter to nonspecialist readers and places the article in the broader context of Internet technologies. In general, we look for the following kinds of articles:

- Reports: Describing particular research or development projects and their potential or actual use. Should include techniques, processes, tools, and environment (type of application domain, hardware and software platform, types of people building and using the system, etc.) sufficient to let readers judge whether the work is relevant to their situation.
- Surveys: Reviewing how a specific technology is applied or a real-world problem is solved today in industry and the most promising research and development activities under way to advance the same.
- Tutorials: Giving step-by-step instructions on how to solve a specific problem or perform a specific development task.

Referees use the following criteria in their evaluations:

- Relevance: significance of the technology and its application or effect.
- Originality: extent to which the material is novel or nonobvious to a broad audience of computer professionals who are applying Internet technologies in their work.
- Validity: soundness of the technical method and conclusions.
- Presentation: clarity of expression and suitability for a broad audience within the profession.

To submit an article for review, e-mail an abstract and, preferably, a URL for the full article to the editor-in-chief or another member of the *IEEE Internet Computing* editorial board. If the article is accepted for review, you will be referred to the Publications Office for instructions to begin the peer-review process. Detailed author guidelines are available at http://computer.org/internet/edguide.htm.