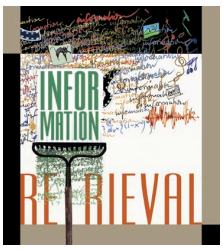
Next-Generation Web Searches for Visual Content



Although visual media account for fully 73 percent of the Web's content, search engines such as ImageScape are only now beginning to sort through these images efficiently.

Michael S. Lew
Leiden University

Major search engines such as Hotbot (http://www.hotbot.com) help us find text on the Web, but typically have few or no capabilities for finding visual media. Yet many Web users—such as magazine editors or professional Web site designers—need to find images using just a few global features. With hundreds of millions of sites to search through,[1] and 73 percent of the Web devoted to images, finding exactly the image you need can be a daunting task.

My colleagues and I developed a prototype system called ImageScape (http://skynet.liacs.nl) to find visual media over intranets and the Web. The system integrates technologies such as vector-quantization-based compression of the image database and k-d trees for fast searching over high-dimensional spaces. ImageScape allows queries for images using

- keywords,
- semantic icons, and
- user-drawn sketches.

Keyword queries offer perhaps the most intuitive query method because they directly relate to the user's vocabulary. Further, HTML provides the ALT field to specify descriptive text. For example, in the following HTML tag, the image of a whale is referenced by the filename, whale.jpg, and the ALT text.

```
<IMG SRC="whale.jpg" ALT="A
  Humpback Whale">
```

However, images frequently lack descriptive text, which eliminates the possibility of text-based searching. In this situation, only content-based methods—those that directly use an image's pictorial information—are feasible.

PICTORIAL-CONTENT-BASED QUERIES

In the early to mid-1990s, IBM's highly influential QBIC[2] system conducted visual searches for similar images on picture databases. This paradigm, shown in Figure 1, displays an initial set of images. The user selects an image, then the search engine ranks the database images by similarity to the selected image with respect to color, texture, shape, or all of these criteria, as Figure 2 shows. This approach requires minimal specialized knowledge from the user, a significant advantage.

Web media search engines such as Webseek,[3] PicToSeek,[4] and ImageRover[5] use the query-by-similar-images paradigm. However, they differ in how they find the initial set of images. In particular, Webseek and ImageRover use text queries to narrow the initial set of images, and PicToSeek asks the user to supply an initial image.

As with any query paradigm, query by similar image has its share of problems. First, it does not let the user run searches based on only part of an image. Suppose, for example, the image content contains a person on a beach underneath a sunset. When the user clicks on the image, the system doesn't know whether the user wants to focus on the person, the beach, or the sunset. Further, the current generation of query-by-similar-image systems uses feature vectors based on global color schemes, texture, and shape. Unfortunately, images that have the same global features can have different picture content. Using local features can overcome this problem and help detect visual concepts such as faces and beaches.

In contrast, ImageScape does not touch upon the

0018-9162/00/$10.00 © 2000 IEEE

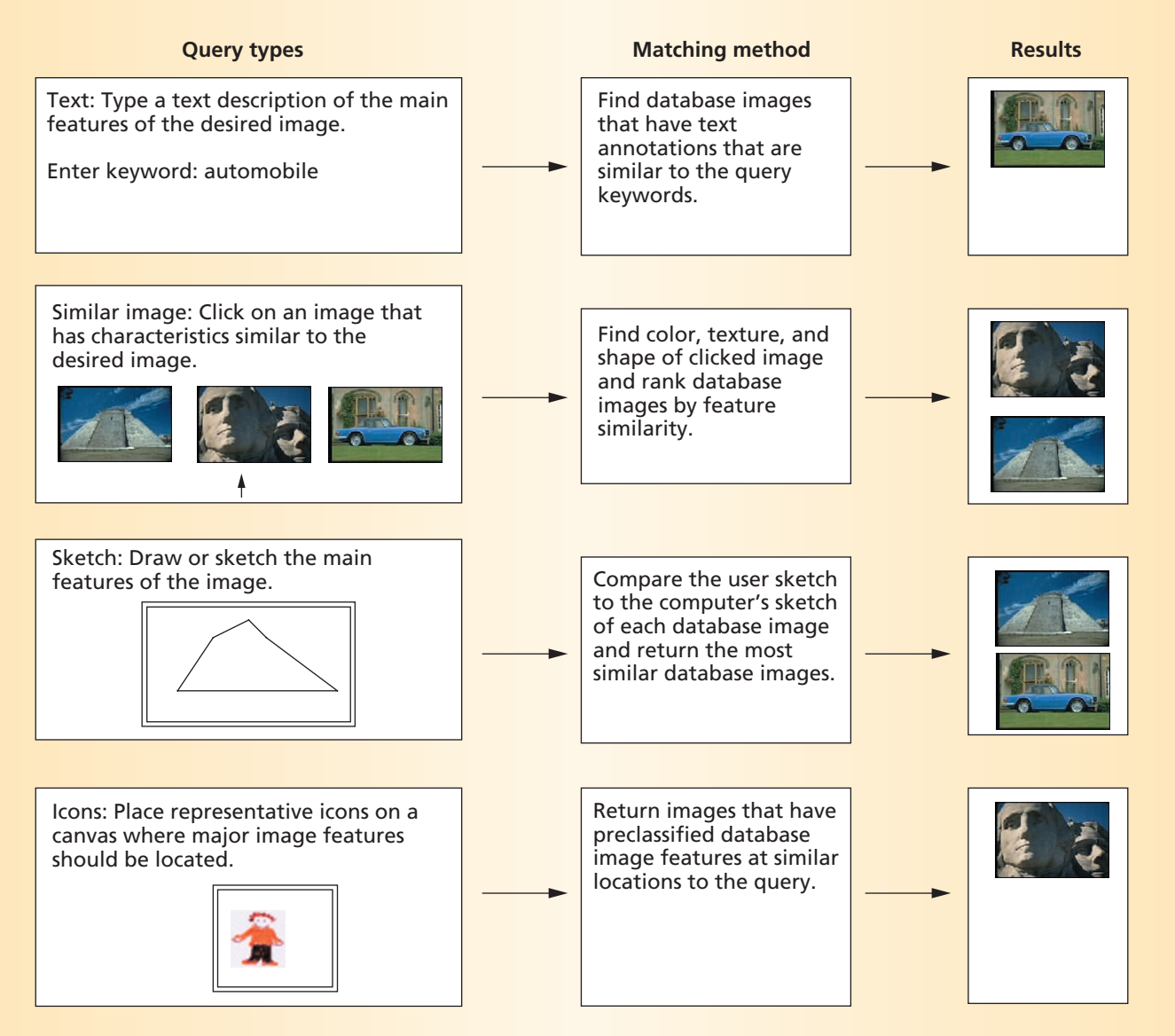


Figure 1. Text and image search paradigms.

query-by-similar-images paradigm. It focuses on techniques for learning visual concepts so that it can use the query-by-icons paradigm. In this paradigm, the user places the icons on a canvas in the position where they should appear in the goal image. Doing so allows the user to explicitly create a query for images of people under a sky, for example. In this context, the database images must be preprocessed for the locations of the available object or concept associated with each icon. The system then returns those database images most similar to the content of objects and concepts specified in the iconic user query. The query-by-icons paradigm has the advantages that users can make a query using their own vocabulary and they can specify the importance of local pictorial features.

We also investigated the query-by-sketch paradigm. In this paradigm, the user creates a query by drawing a rough sketch of the goal image, with the assumption that the sketch will correspond to the object edges and contours. The system then returns the database images with shapes that most closely resemble the user sketch. Sketch queries thus allow the user to directly specify which part of the image is important. Making effective sketch-based queries requires a robust shape matcher.

IMAGESCAPE SYSTEM OVERVIEW

In the ImageScape system, we chose to focus on text and visual media because they are the Web's domi-

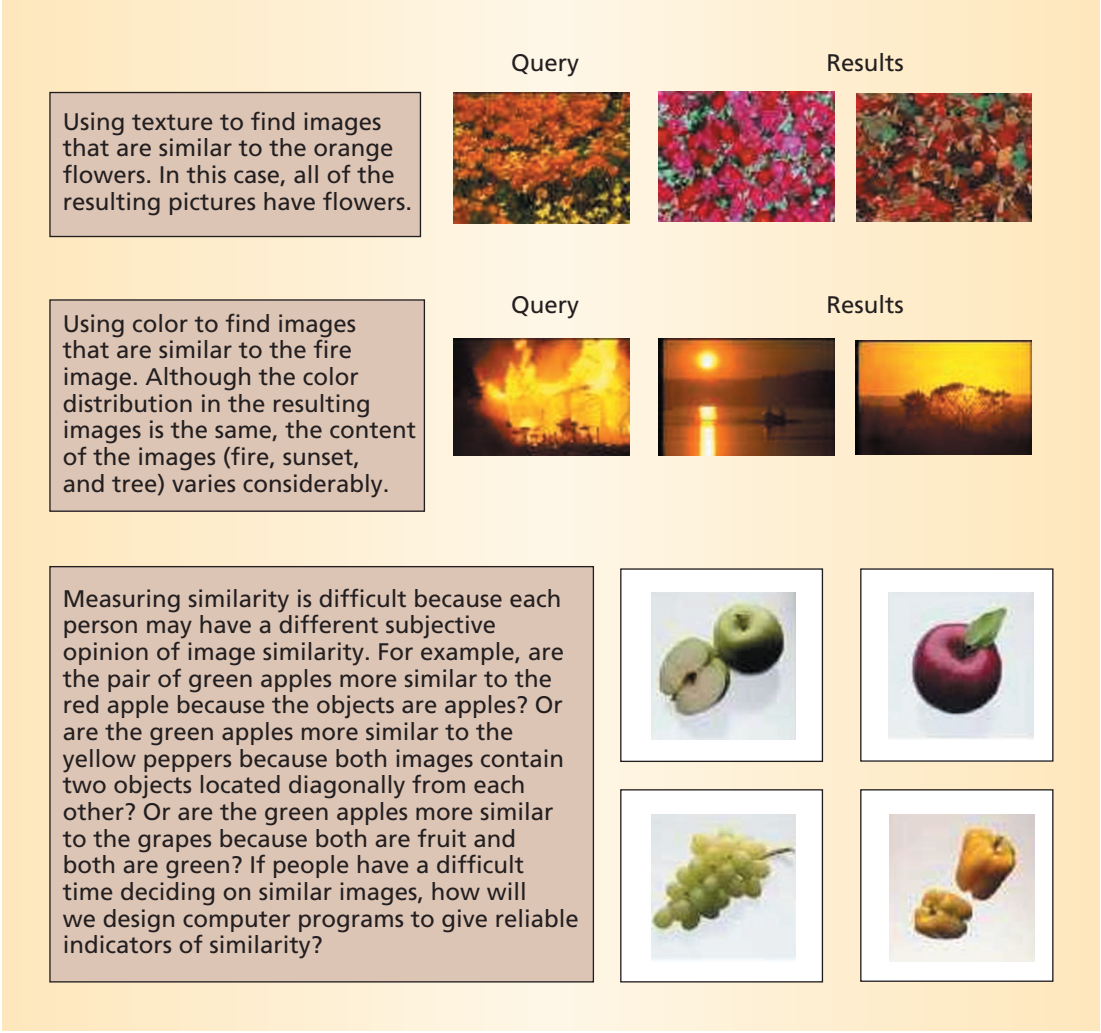


Figure 2. Examples of similar-image-based queries.

nant media. Figure 3 shows the system overview, including the relationships between server, client, and the Web. Continuously sending agents to the Web, the ImageScape system retrieves text, image, and video information.

When ImageScape brings an image to the server, pattern recognition algorithms detect features such as faces, sand, water, and so on, that pertain to the semantic icons and computer sketches. The analysis module creates a thumbnail, a low-resolution copy of the image requiring minimal storage space, and stores the feature vectors in an optimized representation for searching. When a user sends an image query from a Web-based Java browser or client program to the server, the matcher module compares the sketches or semantic icons to the feature database and sends the best-ranked images back to the browser. The primary modules consist of

- vector-quantization-based database compression,
- sketch queries and computer-generated sketches from images,
- visual-concept detection,
- matching of the icons or sketches with the database images, and
- Java client connection to the host server for visual query input and processing and the collection and indexing of the media from the Web.

WEB-BASED MEDIA COLLECTION, INDEXING, AND STORAGE

We can visualize the Web as a graph in which the nodes are Web sites and the edges are hyperlinks at those sites. ImageScape's search procedure performs a priority-based breadth-first search on the hyperlinks found from an initial set of Web sites. The priority is proportional to the site's rate of change and query rate.

Sites that are more likely to have changed since the last visit receive greater priority for a revisit. Further, sites that appear more frequently in the query results also receive greater priority. The Robot Exclusion Protocol also constrains Web searches by specifying the directories the robot can download.

When the robot downloads the images, the system reduces them to thumbnails and stores them in a compressed vector quantization-based database. The system stores similar image blocks with pointers instead of copies.

Storing the media in a compressed database offers the dual advantages of lower storage costs and faster reads from magnetic storage devices. The feature vectors used for indexing the images are stored in k-d trees.[6] These trees are binary-tree representations of the feature space that have a near-logarithmic search performance for finding nearest neighbors or similar images in high-dimensional spaces.

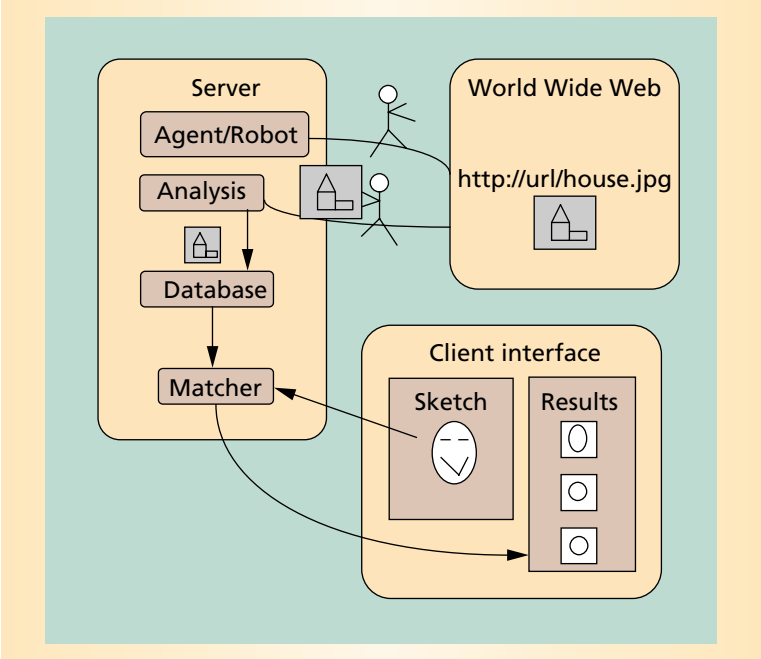


Figure 3. A diagram of ImageScape, a multimedia Web search engine.

SKETCH QUERIES

Our sketch search engine compresses the user sketch at the Java client, sends it to the shape-matching engine, decompresses it, then compares it to each image in the database based on shape similarity. Consequently, the most similar database images are returned to the Java client at the Web browser. The prevalent question is how to measure the shape similarity between the user sketch and a database image. Our starting point for shape comparison was the theory of invariant moments.[7] We derived the moment invariants from shape-statistical moments by normalizing first by the centroid and then by the shape area. Moment invariants have proven useful in two-dimensional shape recognition and can be implemented in real time. However, they can be sensitive to small changes in the shape contour, which we refer to as the local-shape-matching problem.

To solve this problem, we turned to the theory of active contours.[8] Specifically, an active contour is a spline that deforms to fit the particular image based on internal and external forces. The internal forces of the active contour hold the active contour together (elasticity forces) and keep it smooth (bending forces). The external forces guide the active contour toward image features such as high-intensity gradients or edges. The optimal contour position is computed to minimize total energy. The deformation energy is the total energy required to move the active contour from its initial position to its final position. We use the deformation energy to measure the shape similarity between closely matching shapes.

In summary, we split the shape-matching process into two parts: We address global shape-matching with moment invariants and measure local shape-matching by elastic deformation energy, as Figure 4 shows.

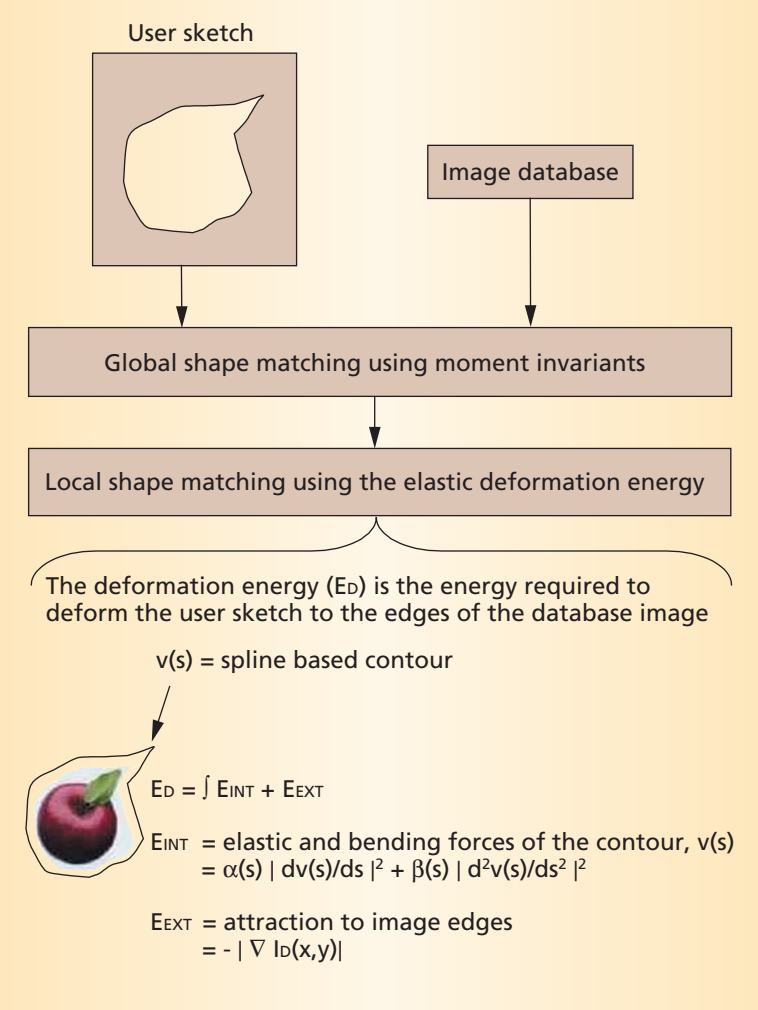


Figure 4. The process of matching a user sketch to the database images.

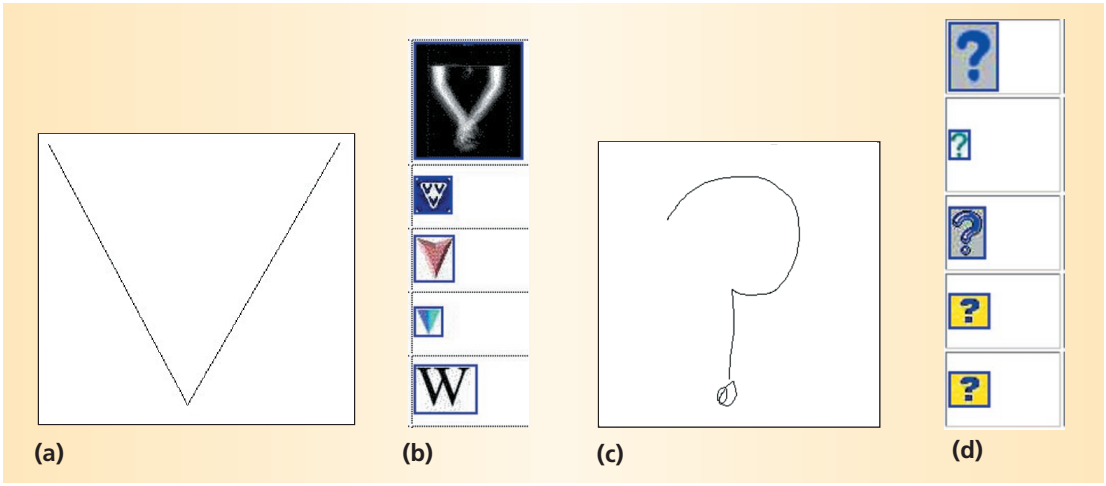


Figure 5. User sketches for (a) the letter V and (c) the question mark symbol. Using shape similarity, the shape matching engine returned the images in (b) for the V and the images in (d) for the question mark symbol.

Figure 5 shows examples of sketch queries and results for the letter V and for a question mark symbol. The results for the V show a variety of database images with roughly similar shapes. For the question mark symbol sketch, the query found several different question mark symbols.

LEARNING VISUAL CONCEPTS AND SEMANTIC QUERIES

Visual-concept detection is essential for the ImageScape search engine because it lets the computer understand our notion of an object or concept. For example, to find an image with a beach under a blue sky, most systems require the user to translate the concept of beach to a particular color and texture. In our system, the user has access to icons that represent concepts such as blue sky and beach. ImageScape can place these icons spatially on a canvas to create a query for a beach under a blue sky.

Rosalind Picard[9] reported promising results in classifying blocks in an image into "at a glance" categories, which people can classify without logically analyzing the content. Picard's method exploits the strengths of multiple feature models. More recently, Aditya Vailaya, Anil Jain, and Hong Jiang Zhang[10] reported success in classifying images as city versus landscape. They found that the edge direction features are effective because city images typically have long lines along the buildings and streets. Natural scenes can be separated because the edges typically curve or consist of short lines from the contours of hills, trees, or grass. Regarding object detection, the recent surge in face recognition research has motivated the development of robust methods for face detection in complex scenery. These methods use techniques such as positive and negative face clusters, neural networks, and information theory.[11]

As an example of human face detection, we use a method that finds human faces in complex backgrounds, then we extend the method to include color, texture, and shape. The Kullback relative information is generally regarded as one of the canonical methods of measuring discriminatory power—how effectively a feature discriminates between two classes. Specifically, we formulated the problem as discriminating between the classes of face and nonface as Figure 6 shows and used the Kullback relative information to measure the class separation, which is the distance between the classes in feature space.

For each pixel, we calculate the Kullback relative information based on the class intensity distributions. The brighter pixels have greater relative information or class separation. The greater the class separation, the easier it is to discriminate between classes. In Figure 6, the image on the right shows that the eye regions have greater discriminatory power than the nose region.

Detecting the faces begins by passing a window over multiple scales—copies of the image at different resolutions—and classifying the window's contents as face or nonface. We perform the classification by using a minimum distance classifier in the feature space defined by the most discriminatory features found from the Kullback relative information.

GENERALIZING TO MULTIPLE MODELS

In face detection, we used pixels, which have the greatest class separation or discriminatory power. Instead of finding the pixels that maximize the class separation, we found the color, texture, and shape features that maximize class separation and minimize the correlation between features. We define this set of features as a discriminatory model. Note that minimizing correlation between features is important because the minimum distance classifier assumes that the features are independent. In summary, we define the visual learning algorithm shown in Figure 7 as follows:

1. Assume that there are M scalar features, each of which has been normalized to 0 to 255.
2. Measure the distribution of the positive examples **F**[x, y], x = 1 to M; y = 0 to 255.
3. Measure the distribution of the negative examples **G**[z, y], z = 1 to M; y = 0 to 255.
4. Calculate the Kullback relative information, **K**[x] from **F** and **G**.

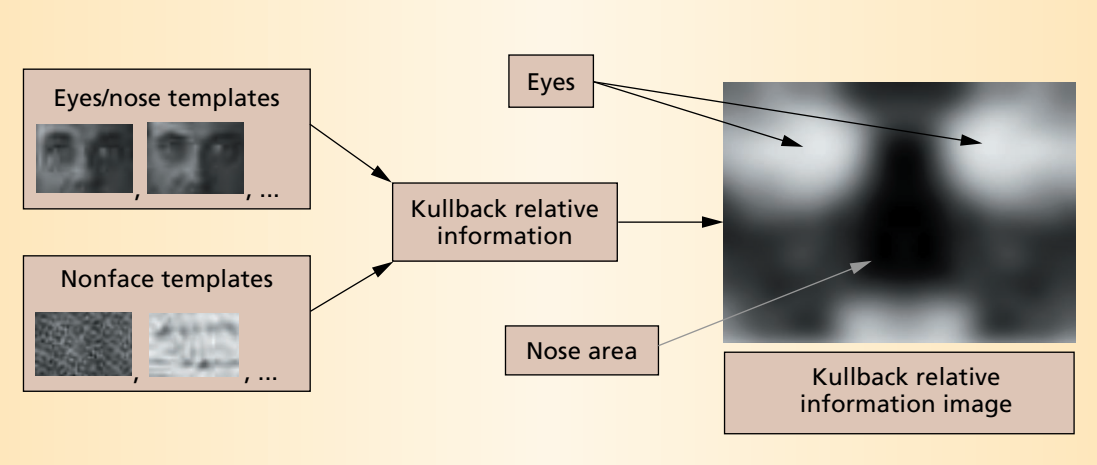


Figure 6. Finding discriminant features. First, we compile a large set of normalized positive (eyes and nose templates) and negative (nonface templates) examples, which comprise two classes: face and nonface. From these examples, we find the distribution of intensity for each pixel in each class.

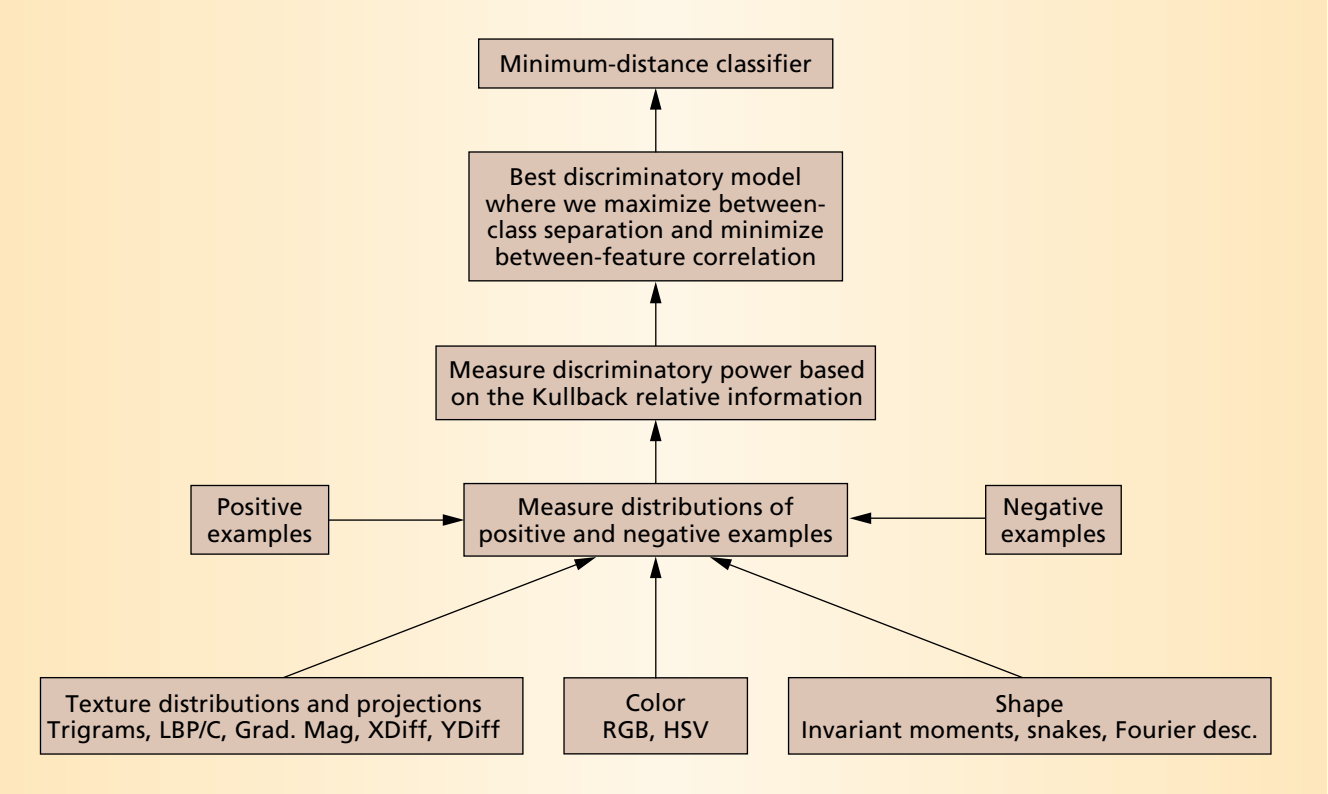


Figure 7. Selecting the best discriminatory model of N features from texture, color, and shape features.

5. Calculate the correlation between features, **C**[x,z] from **F** and **G**.
6. Define the N most informative features as the N features that maximize the Kullback relative information and minimize the correlation between features. This set of features is the discriminatory model, $\mathbf{D}_m[u]$, u = 1 to N.
7. Use $\mathbf{D}_m$ in a minimum-distance classifier.

For each object we want to detect, we collect a large set of positive and negative examples. We then measure a variety of texture, color, and shape features and calculate the Kullback relative information for each one. The candidate features for the system include the texture, color, and shape information from every pixel, as Figure 7 shows.

For the texture models, we use texture distribution models such as LBP, LBP/C, Grad. Mag., XDiff, YDiff,[12] and Trigrams, which are a variant of LBP on the edge image. Texture distribution methods represent a complex texture by measuring the frequency with which a set of atomic textures appears in the

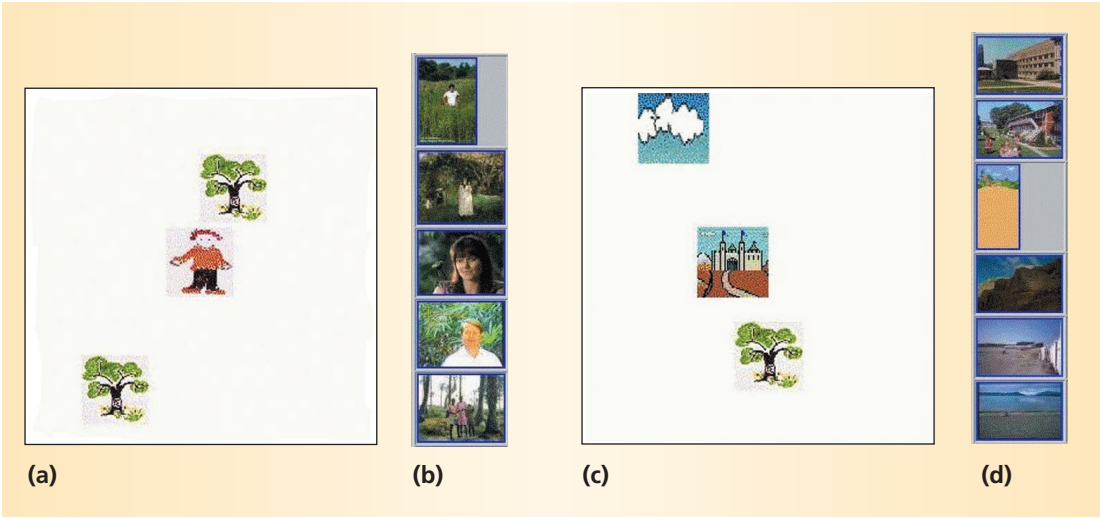


Figure 8. Two examples of query by icons. In (a) the query is made for a person with trees and grass above and below him; (b) displays the results of this query. In (c) the query seeks a picture with sand and stone beneath sky and above trees and grass; (d) displays the results of this query.

image. For shape comparison, we use the features derived from active contours or snakes,[8] invariant moments, and Fourier descriptors.

We place these features into a feature vector, which we then use in a minimum-distance classifier to determine whether or not the search engine detects the visual concept. In the icon queries in Figure 8, the objects' locations are found beforehand, then the matches are ranked by the average sum of squared distance of the query objects to the objects in the database images.

Images dominate the Web's content, but the search systems for finding them have yet to mature. Most Web content is not indexed. Commercial and academic institutions are working on new paradigms for visual search that include searching by icons, sketches, and similar images. These paradigms have the potential to bring the majority of Web information to any individual with a browser.

Each of the query paradigms we describe has associated issues, however. The query-by-similar-image paradigm requires minimal user knowledge, but does not let the user specify particular aspects of the picture to be found. Sketch- and icon-based query methods show promise, but require more research to find the leading algorithms' breaking points. Researchers also must address performance aspects such as efficiency and accuracy so that we can make objective comparisons between visual information retrieval systems. Meanwhile, ours is the only Web search engine that allows both sketch- and icon-based queries. Future research will focus on the fusion of multiple visual learning techniques such as neural networks and decision trees, combining them toward improving visual-concept detection accuracy. ❖

References

1. S. Lawrence and C. Giles, "Searching the World Wide Web," *Science*, 3 Apr. 1998.
2. M. Flickner et al., "Query by Image and Video Content: The QBIC System," *Computer*, Sept. 1995, pp. 23-32.
3. J.R. Smith and S.F. Chang, "Visually Searching the Web for Content," *IEEE MultiMedia*, July-Sept. 1997, pp. 12-20.
4. T. Gevers and A. Smeulders, "PicToSeek: A Content-Based Image Search System for the World Wide Web," *Proc. Visual 97*, Knowledge Systems Institute, Chicago, 1997, pp. 93-100.
5. L. Taycher, M. Cascia, and S. Sclaroff, "Image Digestion and Relevance Feedback in the ImageRover WWW Search Engine," *Proc. Visual 97*, Knowledge Systems Institute, Chicago, 1997, pp. 85-91.
6. R. Egas et al., "Adapting k-d Trees to Visual Retrieval," *Proc. Visual 99*, Springer-Verlag, Berlin, 1999, pp. 533-540.
7. M.K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRA Trans. Information Theory*, Feb. 1962, pp. 179-187.
8. A. Del Bimbo and P. Pala, "Visual Image Retrieval by Elastic Matching of User Sketches," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Feb. 1997, pp. 121-132.
9. R. Picard, "A Society of Models for Video and Image Libraries," *IBM Systems J.*, Vol. 35, No. 3, 1996, pp. 292-312.
10. A. Vailaya, A. Jain, and H. Zhang, "On Image Classification: City vs. Landscape," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, IEEE Press, Piscataway, N.J., 1998, pp. 3-8.

11. M. Lew and N. Huijsmans, "Information Theory and Face Detection," *Proc. Int'l Conf. Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1996, pp. 601-605.
12. T. Ojala, M. Pietikainen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions," *Pattern Recognition*, Vol. 29, No. 1, 1996, pp. 51-59.

Acknowledgments

I gratefully acknowledge the contributions of Kim Lempinen, Leiden Institute of Advanced Computer Science, Leiden University, and Daniel Lewart, University of Illinois, Urbana-Champaign, to the project that developed the ImageScape system.

Michael S. Lew *is a Leiden University Fellow and co-director of the Media Lab at the Leiden Institute for Advanced Computer Science, Leiden University, The Netherlands. His research interests include visual-concept detection, interactive video, Internet searching, and human-computer intelligent interaction. He received a PhD from the University of Illinois, Urbana-Champaign. He is a member of the IEEE. Contact him at mlew@liacs.nl.*