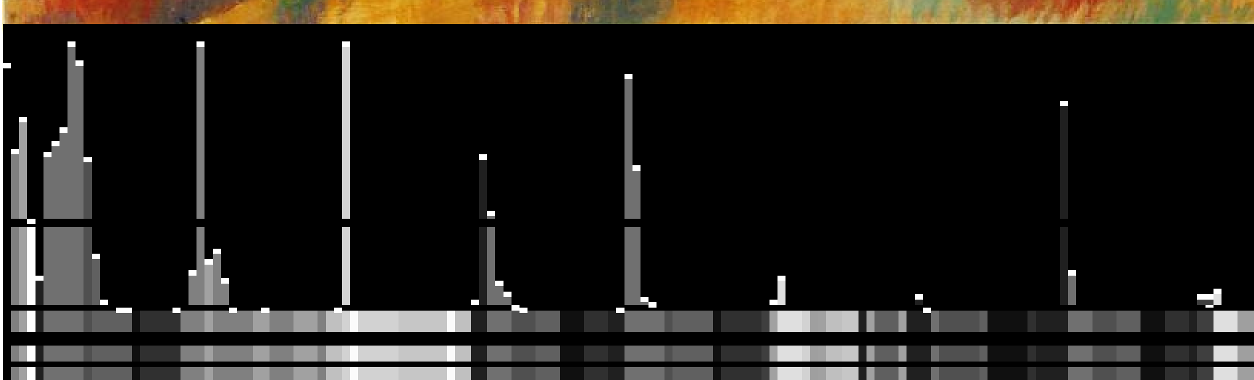

Users can now elicit, store, and retrieve the "imagery-based" information content—metadata and visual features—in visual media as easily as they query text documents.

Visual Information Retrieval

Amarnath Gupta and Ramesh Jain

In 1951, researcher and businessman Calvin Moores coined the term *information retrieval* [10] to describe the process through which a prospective user of information can convert a request for information into a useful collection of references. "Information retrieval," he wrote, "embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, or machines that are employed to carry out the operation." Moores was referring to textual document retrieval, but his description captures what an information retrieval system is expected to do, namely, help a user specify an expressive query to locate relevant information. Here, we extend this notion to nontextual information sources.

Consider a repository of 500,000 stock video clips and an advertising agency looking for just the right footage for a new client request. "Find me a clip" someone might say, "of about two seconds in which a red car racing along a hillside road on a bright day disappears as the road bends around the hill." Now imagine writing this query using a text search engine. To those with experience permuting keywords to locate the right document, the frustration is obvious. Although the query statement arouses similar mental images in most people, a textual specification that always fetches the right video clip is not easy to formulate. The difficulty arises partly because it is impossible to guarantee the video annotator and the user issuing an ad hoc query use simi-

(Opposite) "Carriere de Bibemas" by Paul Cézanne and color histogram of the painting. The color bands (bottom) represent the buckets into which the color space is divided. The vertical bars show the number of pixels in each color bucket.

MOST CURRENT SYSTEMS TRY TO MINIMIZE FALSE-NEGATIVE RESULTS AT THE EXPENSE OF AN INCREASED NUMBER OF FALSE POSITIVES.

lar expressions to describe a clip. There is also a deeper reason: The information sought is inherently in the form of imagery that a textual language, however powerful, is unable to express adequately, making query processing inefficient.

THE ROLE OF THE EMERGING FIELD OF visual information retrieval (VIR) systems is to go beyond text-based descriptors to elicit, store, and retrieve this "imagery-based" information content in visual media. The basic premise behind VIR systems is that images and videos are first-class information-bearing entities and that users should be able to query their content as easily as they query textual documents, without necessarily using manual annotation. Querying content-based alphanumeric information is a perfect example of a new paradigm as described by Henry Lieberman of MIT's Media Laboratory [9]: "It must fundamentally change the way we look at problems we have looked at in the past. It must give us a new framework for thinking about problems in the future . . . When experts in different fields look with curiosity and admiration at each other's domains, and search for commonalities and fresh perspectives on their own, truly new paradigms result." Indeed, the domain of VIR has inherited the analysis component of computer vision and the query component of database systems by tapping older disciplines of computer science: database management and information retrieval systems and image processing and computer vision.

To introduce VIR issues and techniques, we address three basic questions:

- What constitutes the "information content" of an image or video in the specific context of any application?
- With how much meaning can a user specify a search for a desired piece of information?
- How efficient and accurate is the retrieval process?

What Is Visual Information?

Two kinds of information are associated with a visual object (image or video): information about the object, called its metadata, and information contained within the object, called visual features. Metadata is alphanumeric and generally expressible as a schema of a relational or object-oriented database. Visual features are derived through computational processes—typically image processing, computer vision, and computational geometric routines—executed on the visual object.

The simplest visual features that can be computed are based on pixel values of raw data, and several early image database systems [8] used pixels as the basis of their data models. These systems can answer such queries as:

- Find all images for which the 100th to 200th pixels are orange if orange is defined as having a mean value of (red = 255, green = 130, and blue = 0).
- Find all images that have about the same color in the central region of the image as this particular one. The "central region" of the image can be specified by a coordinate system, and the expression "about the same color" is usually defined by computing a color distance. A variant of the Euclidean distance is often used to compare two color values.
- Find all images that are shifted versions of this particular image, in which the maximum allowable shift is *D*.

If the user's requirements are satisfied with this class of queries, data modeling for visual information is almost trivially simple. More realistically, however, a pixel-based model suffers from several drawbacks. First, it is very sensitive to noise, and therefore a couple of noise pixels may be sufficient to cause it to discard a candidate image for the first two queries. Second, translation and rotation invariance are often desirable properties for images. For example, for the third query, if the database contains a 15° rotated version of this image, the rotated version may not be reported by the system. Third, apart from noise, variations in illumination and other imaging conditions affect pixel values drastically, leading to incorrect query results.

These limitations are not to say that pixel-oriented models are without merit. Significant video segmentation results can be obtained by measuring pixel differences over time. For example, an abrupt scene change can be modeled by finding major discontinuities in time-plots of cumulative pixel difference over frames [7]. However, information retrieval based only on pixel values is not very effective by itself.

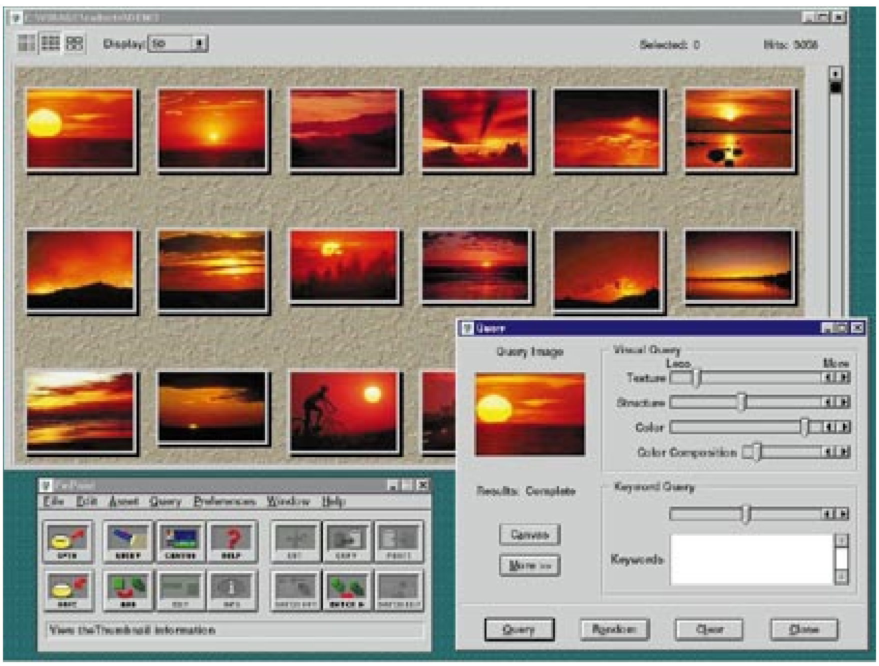

Figure 2. The result of searching the sunset picture in the Query Window, emphasizing color similarity. The next strongest emphasis is structure, which stands for the dominant edges in the picture (here, the horizon line and the outline of the sun).

Furthermore, consider a database of aerial images in which the only objects of interest are buildings, ground vehicles, aircraft, roads, and general terrain. Also imagine that a human interpreter draws bounding rectangles for each region in an image in which one or more of these five kinds of objects appear and labels the regions accordingly. Now we have a fairly precise specification of the information contained in the images. That information can be directly modeled by a relational database schema that maintains the location (bounding box) of each object type and a timestamp for each image. With some additional geometric processing added to this relational model, we can answer very complex queries:

- Is there any location where more than five ground vehicles are close to a building located in the middle of the general terrain?
- Have there been any changes in the position of the aircraft at this location in the past couple of hours?
- Which approach roads have been used by ground vehicles over the past few days to come close to the aircraft?

While these queries are meaningful, the most crucial part of information retrieval—information extraction—is performed by a human using his or her knowledge and experience in aerial-image interpretation. The reason this task requires a human is simple: Fully automatic interpretation of aerial images is still an unsolved research problem. On the other hand, if the human extracts the useful information, one can use a spatial database system to organize and retrieve the information. In a real-life aerial surveillance situation, this approach is unrealistic. For a battlefield application, the territory under surveillance is large enough to need several camera-carrying aircraft. Images from every aircraft, each image several MB in size, stream in at the video rate of 30 frames per second. The high influx of images mean error-free interpretation takes a long time; hence the simple image database scenario we painted is not practical for any time-critical operation.

MOST APPLICATIONS FOR VIR FALL between automated pixel-oriented information models and fully human-assisted database schemes. They do not require pixel-level queries; nor are they constrained to only a few object classes. For these middle-of-the-spectrum applications, visual information can be defined in terms of image-processing transformations computed on the visual object. Although many possible transformations yield meaningful visual features, here we explore several simple examples:

Color. Suppose all the images in a collection are colored. Color is typically characterized by two variables: hue and saturation. Hue denotes the spectrum of colors; saturation for any given hue indicates how much gray has been added to the pure color. Assume

THE DOMAIN OF VIR HAS INHERITED THE ANALYSIS COMPONENT OF COMPUTER VISION AND THE QUERY COMPONENT OF DATABASE SYSTEMS BY TAPPING OLDER DISCIPLINES OF COMPUTER SCIENCE.

that the system computes a 2D histogram of hue and saturation from each image, so bright red and pink occupy two different bins in the histogram. With such computation, a user can answer the following queries (all computing some form of color similarity between images):

- Find all images in which more than 30% of the pixels are sky blue and more than 25% of the pixels are grass green (an outdoor picture?).
- Sort the bins of this image in descending order and find the top five colors. Find all images with the same dominant colors.
- Measure the color distance between two images by computing first their binwise difference (subtracting the first bin of the histogram of image 2 from the first bin of the histogram of image 1 and so on for all bins) and then the sum of the differences over all the bins. Find all images within color distance D of this image.

Figure 1 includes the color histogram (obtained from [12]) of "Carriere de Bibemas" by Cézanne, and Figure 2 shows a color-weighted query made with the Virage Image Engine.

Color Composition. Compute the color histogram of each image as before. Then break up the image into its four quadrants and compute the local histogram for each. Continue this procedure recursively until the quadrants are as small as 16×16 pixels. The result is a data structure called a quadtree of histograms that is yet another abstraction of the original data. Since this abstraction contains some location information, it can be used for more queries, such as:

- Find all images with more than 20% red-orange pixels in the upper right quadrant, more than 20% yellow pixels in the upper left quadrant, and about 30% brown to dark brown pixels in the lower half of the image (a sunset picture?).
- Find all images with a red patch in the center of the image and with a blue patch around it.

Shape. Assume the collection to have clip-art images only. Clip-art images are usually composed of "pure" colors (constant spectral colors with little variation of hue and without added gray). Segment each image into a number of color regions so each region contains a connected set of points having the same pure color. For each segment, compute four properties: color, area, elongation (the ratio of the square of the perimeter and the area), and centrality (distance of the centroid of the region from the center of the image normalized by the image length). Therefore, each image can be abstracted as a list of segments, each having these four properties. Using this list for each image in the collection, we can answer the following queries:

- Find all images having a dominant white square in the center.
- Find all images containing two blue circles (i.e., elongation = 4π) and a red elliptical segment close to the center.

Face Retrieval. A well-known VIR research system is the eigenface image database developed at MIT's Media Laboratory [11]. The system geometrically manipulates each input face image to lie in the same standard coordinate system. The researchers trained their system with a large number of these face images to compute 20 features (called eigenfeatures) representing any human face to a fair degree of detail. Although these features do not correspond to significant physical parts of the face, like eyes, nose, and mouth, they capture enough mathematically robust "information" to find similar faces with good precision.

The purpose of these image transformations is to abstract a set of properties from visual objects sufficient to allow them to be queried. Hopefully, they serve to extract higher levels of information that are more robust, more intuitively meaningful, and more structured than raw pixel values. Not unexpectedly, as the transformations grow increasingly meaningful, they become more complex and more difficult to automate. For example, in the face-retrieval example, even the system's designers needed training to perform effective retrieval. In medical-image databases, fully automatic feature extraction is still a research problem. The general experience is that completely automated image analysis works well only for small, controlled domains and is very computation intensive. Moreover, controlling input in any database is not practical.

Moving from images to videos adds several orders of complexity. Most research and commercial efforts take the following approach: Consider a video clip as a large number of image frames with progressively varying image content. From this sequence, find the frames at which a significant transition in image content occurs. For example, a cut is an abrupt scene transition, and a fade is a gradual scene transition. The segments of the video between these transitions are called shots. Use some sampling strategy to extract some "key frames" from each shot. Treat each key frame as an image to perform the same analysis as can be performed on still images.

However, this approach does not make good use of the temporal and motion information inherent in videos. Videos contain three kinds of motion information: one due to movement of the objects within a scene, one due to motion of the camera, and one due to special post-processing effects, like image warping. Some systems [4] use the motion-encoding in compressed video formats (e.g., MPEG video files) to extract the motion information. These systems work well for isolated object motions in the scene. Some systems [3] disregard the whole problem of information extraction from videos and assume that symbolic descriptions of image sequences are available. These systems treat video information as a database research problem for spatiotemporal properties of rigid objects. Ideally, a video information system integrates all these different pieces into a single computational framework, but current research is not there yet.

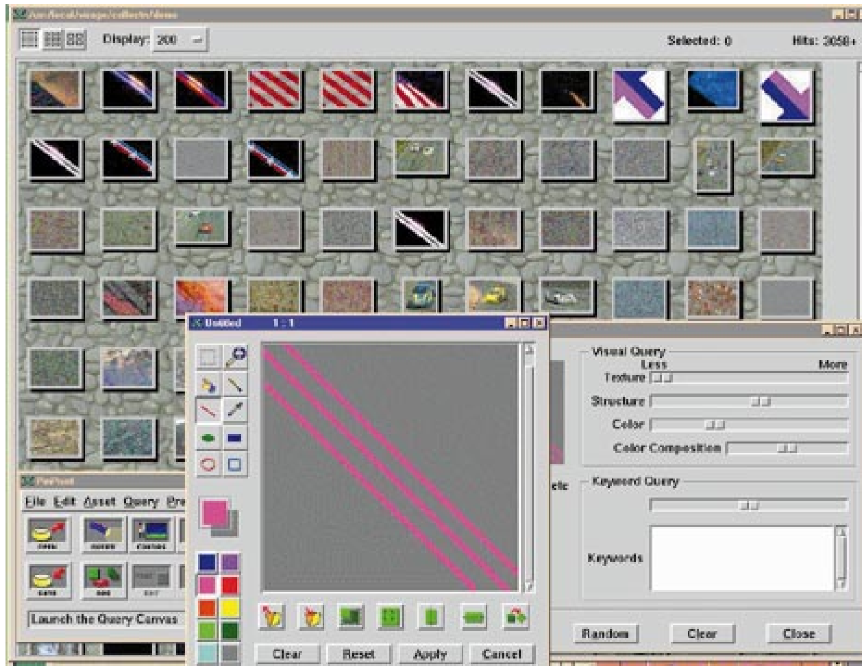

Figure 3. When a refinement is made to a Virage Canvas query—to indicate that, say, the user is much more interested in the "diagonal line"-ness of an image than in color—the results improve considerably. The road sign images containing diamonds rank much higher after the refinement.

Specifying a Visual Information Query

The primary intent of a visual query must be to cap-

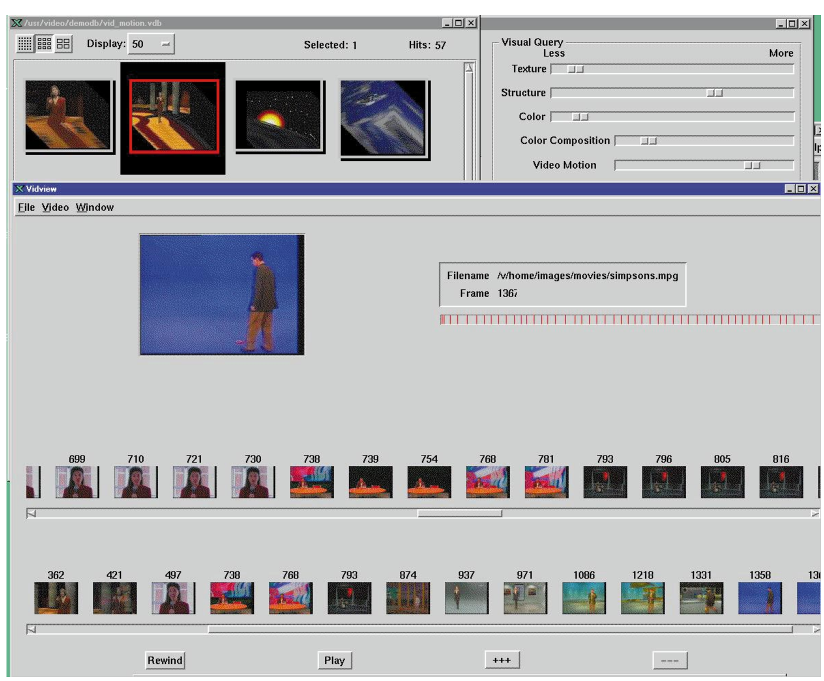

Figure 4. Query by video similarity. The first picture on the Results Panel shows the query video. The Vidview panel shows two rows of key frames from the clip. The first row oversamples (the deliberate picking of more frames than will be used) the video by computing the activity in the video. The second row refines the sampling by locating distinct key frames.

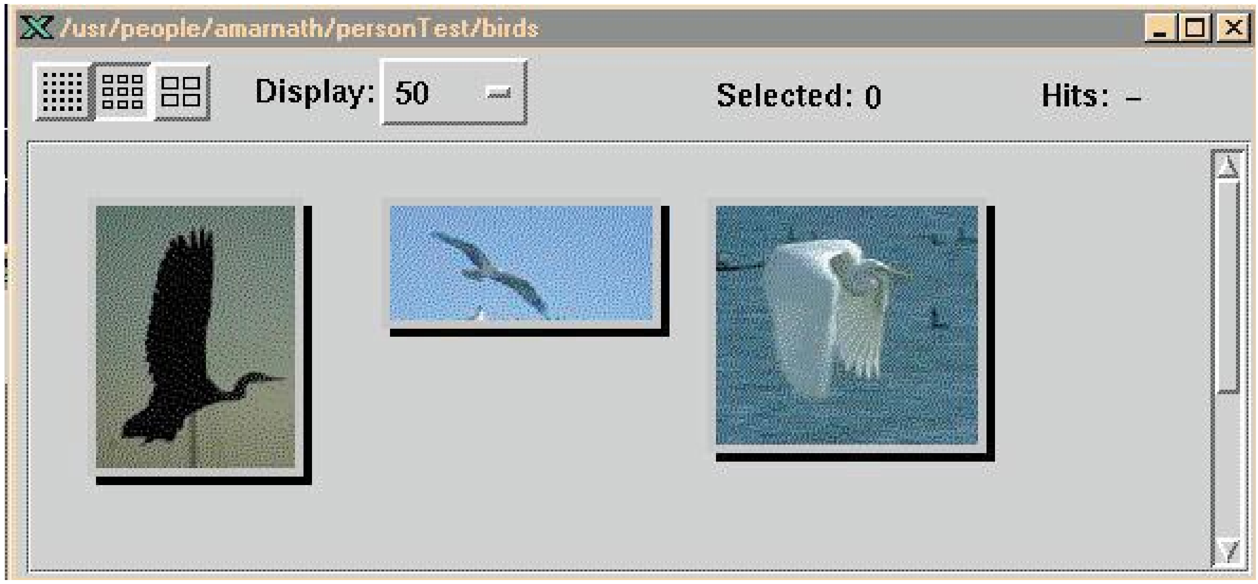

Figure 5. The semantics common to all images is a flying bird. An object-level query should find them similar, while an image-based query finds them different.

ture the user's mental image of a specific picture or video image. One style of research, evolving from traditional approaches, has developed both textual and visual query languages. A textual query language, such as PICQUERY+ [1], has constructs to "compose" a visual description through textually specified attributes and operators to specify spatial, temporal, "evolutionary" (e.g., `splits into`) relationships. In a visual query language [2], the user visually places object icons to specify relative locations, orientations, and sizes of objects within the desired image.

Recently developed image information systems, both research and commercial, lean more toward a query-by-example paradigm. There are two different ways an example can be provided. In the first, the example is pictorial; the user specifies a query either by providing another example image or by partly drawing what a desired image should look like. Figure 3 shows the result of a query-by-image-example in the Virage system. In the second [6], the user provides value-examples for one or more visual features, something like: an image with about 30% green and 40% blue, with a grass-like texture in the green part. The values not provided are in English but through visual tools that allow the user to choose colors and texture. Some recent systems also let users refine their queries. Query refinement can be done either by using a result image from a previous query to launch a new query or by modifying a result image with an image processing tool to specify the additional criteria the returned images must satisfy. Such operations might include erasing part of the image, changing the brightness level, or painting a part with a different color. Another kind of query refinement involves changing the relative weights of visual features and having the system re-rank the previous results with the new weights. A query can be refined in the Virage system by increasing the relative weight of a shape-like structure primitive.

Query specification is significantly more complex for videos. With the current level of development, systems are more concerned with correctly finding the transitions or developing the right algorithm for keyframe extraction and story grouping than with video queries. These systems typically offer simple video queries by example:

- Find a video collection with a keyframe like a given image.
- Rank the clips in the video collection in order of their similarity with a given video clip, in which the criteria for similarity can be specified and altered by setting and adjusting visual and motion parameters displayed to the user.
- Cue the given video clip to the frame that is like the given query image.
- Cue the given video to the frame that has the closest framewise similarity to the given query video.

Figure 4 shows the results of a video similarity query with the Virage system.

Most of the current VIR systems are limited in the query types they can handle. Development of a comprehensive language for visual assets is a far more difficult task. However, query specification for visual information should not be performed exclusively with an example- or specification-based paradigm but through a collection of different tools that—in concert—serve as a VIR "query language." Such a collection should include at least nine items:

An image-processing tool. Such a tool would interactively segment the image or modify the properties of a local region in the image. It might be used both during the image insertion process to aid the image analysis, and during the query to express search conditions on specific regions of the image. Such operations include changing the texture, pasting a different foreground object on the same background scene, or highlighting the edge of an unclear region of interest.

A feature-space manipulation tool. Such a tool would allow better specification of the search condition on the features instead of on an image. The histogram-based queries mentioned earlier exemplify this class of query. More generally, such a tool would allow the user to explore the feature space and to specify a neighborhood query. The user might ask: "If each image is viewed as a point in the *n*-dimensional feature vector space, find the nearest *x* images within distance *d* of this image." While most current systems support this kind of query, they execute it blindly and do not allow the user to interactively navigate in the feature space and modify query conditions based on this interaction.

An object specification tool. Such a tool would resolve the potential conflict between queries looking for search conditions on images and those looking for search conditions on objects recognizable within images. To illustrate this difference, the three images in Figure 5 are very different in their general image content but contain similar objects, especially for domain-specific systems in which the object of interest occupies only part of the image. The same is true for videos: An object for query might have to be specified by analyzing the image sequence by motion grouping.

A measurement specification tool. Such a tool would be used in any domain in which the size of the objects or regions in an image is an important concern. As required in several image domains, such as medical-image databases, this tool should allow the user to perform online measurements and provide tolerance conditions on a query region. It should also allow the user to retrieve from differently zoomed versions of the same image.

A classification tool. Such a tool would allow the user to perform a grouping operation on visual objects by specifying a grouping criterion on one or more visual features of interest. This grouping allows queries like: "Based only on elongation and texture, what are the major groups of tumor objects in this collection of magnetic resonance images? Display two images from each group."

A spatial arrangement tool. Such a tool would allow the user to specify location-sensitive queries and move query objects denoted by the object specification tool to position them in the place of interest. A query might be stated as: "Find all images containing all the same objects as this one but having them arranged differently." It should also allow the user to state whether the location sensitivity of the objects is absolute or relative and allow queries that include area range restrictions that can be imposed on image regions (e.g., all green regions should have an area between A1 and A2), or that mention Boolean combinations of spatial attributes (e.g., the red circles could be here or possibly here, but not there).

A temporal arrangement tool. Such a tool for video would specify temporal events as search conditions. We are referring not to semantic events like "the butler did it" but more to the change patterns of objects and images. For example, in a video collection, a query may ask: "Find all clips in which a freeze shot is followed by a jump cut into a very dynamic scene." This tool must work together with image motion descriptors and video segmentation primitives so the user can specify search conditions on temporal patterns of both image-related and object-related transitions.

An annotation tool. Such a tool would alleviate the one major limitation of example-based systems—that users may know exactly what they are looking for yet lack an example image to initiate the query. Ideally, the annotation tool should have capabilities similar to those of a text processing engine. However, it must allow different levels of annotation to be associated with objects or regions *within* an image frame, with a *whole* image frame, and with an *image group*. Such annotation is necessary for video objects for making annotations at the story level.

A data definition tool. Such a tool would enable applications in which the user has a prior set of models to characterize properties of the image. For a database of, say, chest X-rays or mug shots, the tool would help define a database schema (to define the contents), so the user can specify a query like: "Find other mug shots with similar facial features, but bigger eyes, wider lips, and a scar on the left side of the eyebrow." The other task of a data definition tool is to support an ontology and examples of words for cases in which visual descriptions with a schema are too complex to create. An example would be to create a set of image examples for the word "human" so these examples may be used to start a query on humans.

Comparing VIR Systems

IN ITS CURRENT STATE OF DEVELOPMENT, VIR faces several problems characteristic of any emerging field. As the principles and techniques behind VIR have matured and improved over the past five years, more commercial and research systems have become available. As this issue went to press there were at least 20 research groups working toward some form of generic or specialized VIR system, including at least three commercial products. These systems differ in application domain, choice of visual features, techniques employed for computing visual features, and query mechanisms supported. How can we compare these systems? Unfortunately, not not enough effort has been directed to establishing criteria for evaluating, benchmarking, and comparing VIR systems. This lack of effort is in part attributable to the subjective character of the domain. It is extremely difficult to set a gold standard for ranking a database of assorted images in terms of their similarity to a given image. A significant variation can be observed among rankings produced by different users and even between two assessments by the same user at different times. Despite these hurdles, it is important to develop a set of general criteria to assess the relative performance of VIR systems.

In our experience with user groups, we found users make at least two kinds of judgments when comparing VIR systems:

Goodness of retrieval. This judgment roughly corresponds to the extent to which a system's query results correspond to users' mental images of what should be retrieved by a system for a benchmark database. Some simple "measures" are surmised through user interaction, as in:

- "The system is good because querying with a flower garden example retrieves 90% outdoor scenes of which 80% are flower gardens." The user judges the system's goodness by the number of "correct" matches in the first few screenshots.
- "The color and texture of the third result are about right, but this shape in the middle is not." Here, the user implicitly measures the *dimension* and *degree of relevance* for each relevant result.
- "The result is poor because these three roses should have been ranked higher than the car and the baby on the grass." The criterion here is *relative rank* of relevant vs. irrelevant objects. A system in which for the same query some relevant images are ranked lower than irrelevant images is judged as performing worse than the system in which the top images are consistently more relevant than the images appearing in the lower ranks.
- "Why didn't that desert image I saw one time show up in the first two screens? Does it appear when I reduce the color weight? What if I also increase the texture weight?" The user in this case has shifted from an "image-browser" mode of search to an "image-locator" mode of search. Now the criterion for correctness is the *deviation from expected rank* of a reference image and the *incremental improvement achieved per query refinement operation*.
- "It is not clear what combination of weights can retrieve what I want. Is there a way to mark the results I liked and disliked and have the system figure out how to improve the results?" The user is referring to the need for *relevance feedback*, a mechanism through which the user looks at the responses produced by the system for a query and rates the result objects with a score of their relevance. The system uses this rating to modify and repeat the query, expecting to come up with more relevant results the next time.

The most noticeable aspect in these rough "measures" is that the users' judgment of goodness is based on how much of the retrieved data is good rather than on how much of the relevant data is retrieved [5]. Most current systems try to minimize false-negative results at the expense of an increased number of false positives. A balanced, optimized approach to VIR performance improvement is not yet a reality.

Effectiveness. As one moves from a general-purpose system to a more domain- and application-specific VIR deployment, user queries become more sophisticated and purposeful. The criteria of assessment also change in order to measure the effectiveness of retrieval in the specific context of the application problem. Based on the application problems for which we have customized the extensible Virage Image Engine, we offer several observations on the perception and the reality of effectiveness:

- In many specific applications, the process of visual feature extraction is limited by the availability of fast, implementable techniques in image processing and computer vision—and is never perfect. Therefore, it is necessary to treat the effectiveness of retrieval separately from the underlying image processing algorithms. However, users seldom make the distinction, judging the system's performance by the results of the retrieval—without realizing there may be no practical algorithm to effectively compute the features they want. Making the distinction while judging the efficacy of a VIR system is an important part of a user's education.
- We found it useful for the user to estimate how different the result would be from the given results by assuming the system produced perfect feature extraction. The results of similarity-based retrieval are generally not so sensitive to small errors in feature extraction. For example, in an ophthalmological application, although the feature extraction module did not find the complete length of every blood vessel in the optical fundus, a query like "Find all patients with a tortuosity (curliness) pattern like this patient's" produces almost the same results as though blood vessels were extracted perfectly. In the same vein (so to speak), queries involving aggregate values (such as density of microcalcifications in mammograms) produce fairly faithful results. Queries involving measurement (such as diameter of a lesion) work better with human-assisted feature extraction.
- More often than not, the critical issues influencing effectiveness are the choice of similarity functions and the selection of proper features. In a trademark-search application, selecting a moment-based shape feature made a dramatic improvement in effectiveness for the user. In an ophthalmological application, choosing a fuzzy similarity function made a significant difference over choosing a weighted Euclidean metric. Our conclusion: Making the system more effective is usually an engineering art. The real merit of a VIR system is its ability to allow enough extensibility and flexibility that it can be tuned to any user application.

What's Next?

Many aspects of VIR systems are important but not yet properly understood. An example is the delivery mechanism for visual information. Many users need to access images, but few can afford to maintain a large repository. The technology will be grossly underutilized if millions of users cannot access remote and distributed repositories. Users should be able to not only issue interactive queries but to use them in conjunction with their limited local resource and store some interesting results locally. Another equally important need is for VIR developers and researchers to recognize that to a user, information is a gestalt, and visual information retrieval, like structured databases and text retrieval, is only one part of it. It is mandatory for developers and researchers to take steps to make visual information easy to cross-reference from other modes of information. At the rate technology is advancing, we are hopeful that all these goals can be met in five to seven years. C

References

1. Cardenas, A.F., Ieong, I.T., Barker, R., Taira, R.K., and Breant, C.M. The knowledge-based object-oriented PICQUERY+ language system. *IEEE Trans. Knowl. Data Eng. 5*, 4 (Aug. 1993), 644–658.
2. Chang, S.K., and Hsu, A. Image information systems: Where do we go from here? *IEEE Trans. Knowl. Data Eng. 4*, 5 (Oct. 1992), 431–442.
3. Del Bimbo, A., Vicario, E., and Zingoni, D. Symbolic description and visual querying of image sequences with spatio-temporal logic. *IEEE Trans. Knowl. Data Eng. 7*, 4 (Aug. 1995), 609–622.
4. Dimitrova, N., and Golshani, F. Motion recovery for video content classification. *ACM Trans. Inf. Syst. 13*, 4 (Oct. 1995), 408–439.
5. Faloutsos, C., Barber, R., Flickner, M., and Hafner, J. Efficient and effective querying by visual content. *J. Intell. Inf. Syst. 3*, 3–4 (July 1994), 231–62.
6. Flickner, M., Shawney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steel, D., and Yonker, P. Query by image and video content: The QBIC system. *IEEE Comput. 28*, 9 (Sept. 1995), 23–31.
7. Hampapur, A. Designing video data management systems. Ph.D. dissertation, The Univ. of Michigan, Ann Arbor, 1995.
8. Iyengar, S.S. and Kashyap, R.L., Guest Eds. Special section on image-database systems. *IEEE Trans. Software Eng. 14*, 5 (May 1988), 608–688.
9. Lieberman, H. Intelligent graphics. *Commun. ACM 39*, 8 (Aug. 1996), 38–48.
10. Moores, C.N. Datacoding applied to mechanical organization of knowledge. *Am. Doc. 2* (1951), 20–32.
11. Pentland, A., Moghaddam, B., and Starner, T. View-based and modular eigenspaces for face recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Seattle, Wash., June 21–23). IEEE Computer Society Press, Los Alamitos, Calif., 1994, pp. 84–91.
12. The WebSeek demonstration web page, ADVENT laboratory, Columbia University, 1996. http://www.ctr.columbia.edu/webseek.

Amarnath Gupta (amarnath@virage.com) is a senior software scientist in Virage, Inc., a developer and vendor of VIR systems in San Mateo, Calif.

Ramesh Jain (jain@ece.ucsd.edu) is a professor of electrical and computer engineering in the University of California at San Diego and is the chairman of the board and founder of Virage, Inc.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© ACM 0002-0782/97/0500 $3.50