Query by Image and Video Content: The QBIC System

Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker

IBM Almaden Research Center

QBIC* lets users find pictorial information in large image and video databases based on color, shape, texture, and sketches. QBIC technology is part of several IBM products.

*To run an interactive query, visit the QBIC Web server at http://wwwqbic.almaden.ibm.com/.

Picture yourself as a fashion designer needing images of fabrics with a particular mixture of colors, a museum cataloger looking for artifacts of a particular shape and textured pattern, or a movie producer needing a video clip of a red car-like object moving from right to left with the camera zooming. How do you find these images? Even though today's technology enables us to acquire, manipulate, transmit, and store vast on-line image and video collections, the search methodologies used to find pictorial information are still limited due to difficult research problems (see "Semantic versus nonsemantic" sidebar). Typically, these methodologies depend on file IDs, keywords, or text associated with the images. And, although powerful, they

- don't allow queries based directly on the visual properties of the images,
- are dependent on the particular vocabulary used, and
- don't provide queries for images similar to a given image.

Research on ways to extend and improve query methods for image databases is widespread, and results have been presented in workshops, conferences,[1,2] and surveys.

We have developed the QBIC (Query by Image Content) system to explore content-based retrieval methods. QBIC allows queries on large image and video databases based on

- example images,
- user-constructed sketches and drawings,
- selected color and texture patterns,

Semantic versus nonsemantic information

At first glance, content-based querying appears deceptively simple because we humans seem to be so good at it. If a program can be written to extract semantically relevant text phrases from images, the problem may be solved by using currently available text-search technology. Unfortunately, in an unconstrained environment, the task of writing this program is beyond the reach of current technology in image understanding. At an artificial intelligence conference several years ago, a challenge was issued to the audience to write a program that would identify all the dogs pictured in a children's book, a task most 3-year-olds can easily accomplish. Nobody in the audience accepted the challenge, and this remains an open problem.

Perceptual organization—the process of grouping image features into meaningful objects and attaching semantic descriptions to scenes through model matching—is an unsolved problem in image understanding. Humans are much better than computers at extracting semantic descriptions from pictures. Computers, however, are better than humans at measuring properties and retaining these in long-term memory.

One of the guiding principles used by QBIC is to let computers do what they do best—quantifiable measurement—and let humans do what they do best—attaching semantic meaning. QBIC can find "fish-shaped objects," since shape is a measurable property that can be extracted. However, since fish occur in many shapes, the only fish that will be found will have a shape close to the drawn shape. This is not the same as the much harder semantical query of finding all the pictures of fish in a pictorial database.

0018-9162/95/$4.00 · 1995 IEEE

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

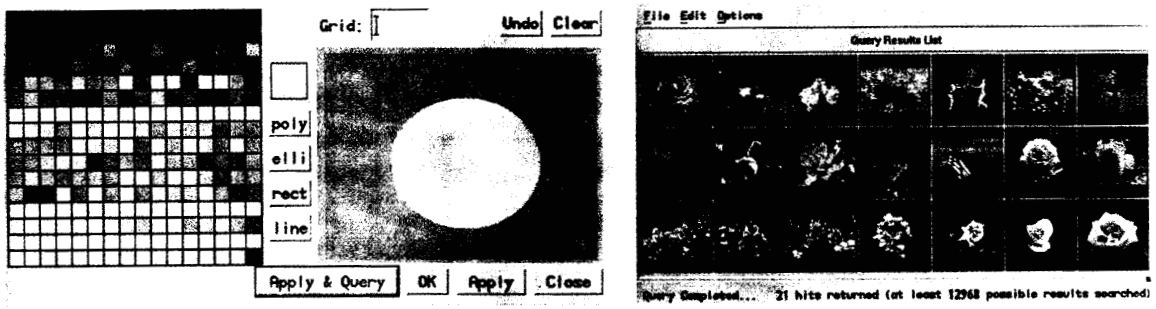

Figure 1. QBIC query by drawn color. Drawn query specification on left; best 21 results sorted by similarity to the query on right. The results were selected from a 12,968-picture database.

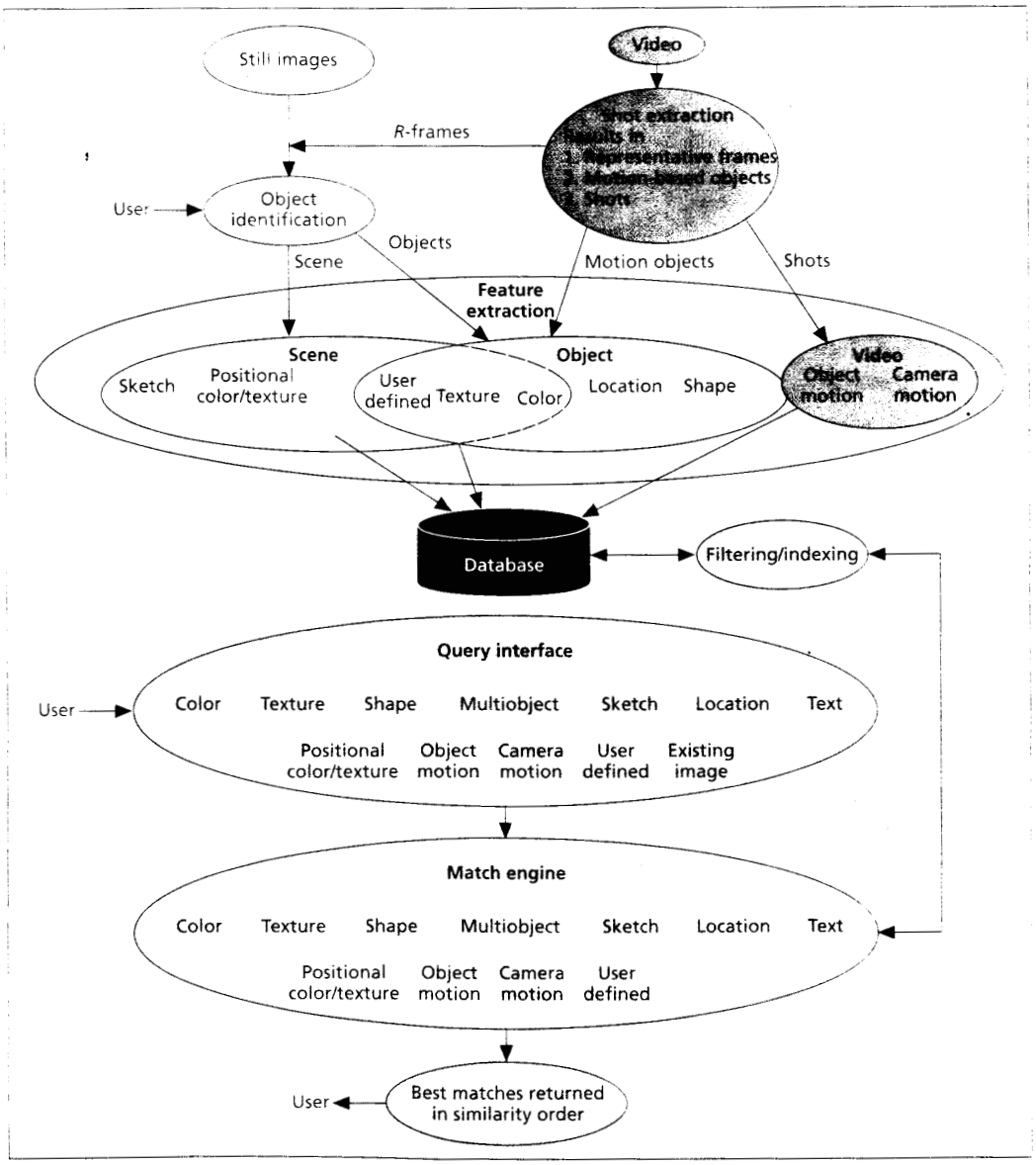

Figure 2. QBIC database population (top) and query (bottom) architecture.

Best Copy Available

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

- camera and object motion, and
- other graphical information.

Two key properties of QBIC are (1) its use of image and video content—computable properties of color, texture, shape, and motion of images, videos, and their objects—in the queries, and (2) its graphical query language in which queries are posed by drawing, selecting, and other graphical means. Related systems, such as MIT's Photobook[3] and the Trademark and Art Museum applications from ETL,[4] also address these common issues. This article describes the QBIC system and demonstrates its query capabilities.

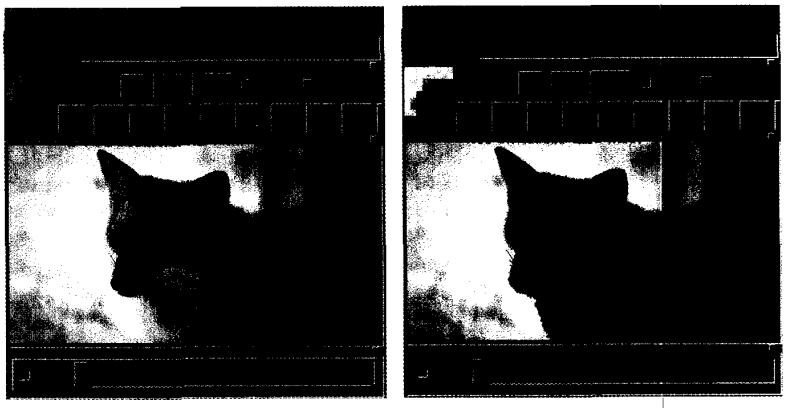

Figure 3. QBIC still image population interface. Entry for scene text at top. Tools in row are polygon outliner, rectangle outliner, ellipse outliner, paintbrush, eraser, line drawing, object translation, flood fill, and snake outliner.

U-M-I
BEST COPY AVAILABLE

QBIC SYSTEM OVERVIEW

Figure 1 illustrates a typical QBIC query.* The left side shows the query specification, where the user painted a large magenta circular area on a green background using standard drawing tools. Query results are shown on the right: an ordered list of "hits" similar to the query. The order of the results is top to bottom, then left to right, to support horizontal scrolling. In general, all queries follow this model in that the query is specified by using graphical means—drawing, selecting from a color wheel, selecting a sample image, and so on—and results are displayed as an ordered set of images.

To achieve this functionality, QBIC has two main components: database population (the process of creating an image database) and database query. During the population, images and videos are processed to extract features describing their content—colors, textures, shapes, and camera and object motion—and the features are stored in a database. During the query, the user composes a query graphically. Features are generated from the graphical query and then input to a matching engine that finds images or videos from the database with similar features. Figure 2 shows the system architecture.

Data model

For both population and query, the QBIC data model has

- still images or scenes (full images) that contain objects (subsets of an image), and
- video shots that consist of sets of contiguous frames and contain motion objects.

For still images, the QBIC data model distinguishes between "scenes" (or images) and "objects." A scene is an image or single representative frame of video. An object is a part of a scene—for example, the fox in Figure 3—or a moving entity in a video. For still image database population, features are extracted from images and objects and stored in a database as shown in the top left part of Figure 2.

Videos are broken into clips called shots. Representative frames, or *r*-frames, are generated for each extracted shot. *R*-frames are treated as still images, and features are extracted and stored in the database. Further processing of shots generates motion objects—for example, a car moving across the screen.

Queries are allowed on objects ("Find images with a red, round object"), scenes ("Find images that have approximately 30-percent red and 15-percent blue colors"), shots ("Find all shots panning from left to right"), or any combination ("Find images that have 30 percent red and contain a blue textured object").

In QBIC, similarity queries are done against the database of pre-extracted features using distance functions between the features. These functions are intended to mimic human perception to approximate a perceptual ordering of the database. Figure 2 shows the match engine, the collection of all distance functions. The match engine interacts with a filtering/indexing module (see "Fast searching and indexing" sidebar, next page) to support fast searching methodologies such as indexing. Users interact with the query interface to generate a query specification, resulting in the features that define the query.

DATABASE POPULATION

In still image database population, the images are reduced to a standard-sized icon called a thumbnail and annotated with any available text information. Object identification is an optional but key part of this step. It lets users manually, semiautomatically, or fully automatically identify interesting regions—which we call objects—in the images. Internally, each object is represented as a binary mask. There may be an arbitrary number of objects per image. Objects can overlap and can consist of multiple disconnected components like the set of dots on a polka-dot dress. Text, like "baby on beach," can be associated with an outlined object or with the scene as a whole.

Object-outlining tools

Ideally, object identification would be automatic, but this is generally difficult. The alternative—manual identification—is tedious and can inhibit query-by-content

** The scene image database used in the figures consists of about 7,450 images from the Mediasource Series of images and audio from Applied Optical Media Corp., 4,100 images from the PhotoDisc sampler CD, 950 images from the Corel Professional Photo CD collection, and 450 images from an IBM collection.*

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

Fast searching and indexing

Indexing tabular data for exact matching or range searches in traditional databases is a well-understood problem, and structures like *B*-trees provide efficient access mechanisms. In this scenario, indexing assures sublinear search while maintaining completeness; that is, all records satisfying the query are returned without the need for examining each record in the database. However, in the context of similarity matching for visual content, traditional indexing methods may not be appropriate. For queries in which similarity is defined as a distance metric in high-dimensional feature spaces (for example, color histogram queries), indexing involves clustering and indexable representations of the clusters. In the case of queries that combine similarity matching with spatial constraints on objects, the problem is more involved. Data structures for fast access of high-dimensional features for spatial relationships must be invented.

In a query, features from the database are compared to corresponding features from the query specification to determine which images are a good match. For a small database, sequential scanning of the features followed by straightforward similarity computations is adequate. But as the database grows, this combination can be too slow. To speed up the queries, we have investigated a variety of techniques. Two of the most promising follow.

Filtering

A computationally fast filter is applied to all data, and only items that pass through the filter are operated on by the second stage, which computes the true similarity metric. For example, in QBIC we have shown that color histogram matching, which is based on a 256-dimensional color histogram and requires a 256 matrix-vector multiply, can be made efficient by filtering. The filtering step employs a much faster computation in a 3D space with no loss in accuracy. Thus, for a query on a database of 10,000 elements, the fast filter is applied to produce the best 1,000 color histogram matches. These filtered histograms are subsequently passed to the slower complete matching operation to obtain, say, the best 200 matches to display to a user, with the guarantee that the global best 200 in the database have been found.

Indexing

For low-dimensional features such as average color and texture (each 3D), multidimensional indexing methods such as *R**-trees can be used. For high-dimensional features—for example, our 20-dimensional moment-based shape feature vector—the dimensionality is reduced using the K-L, or principal component, transform. This produces a low-dimensional space, as low as two or three dimensions, which could be indexed by using *R**-trees.

applications. As a result, we have devoted considerable effort to developing tools to aid in this step. In recent work, we have successfully used fully automatic unsupervised segmentation methods along with a foreground/background model to identify objects in a restricted class of images. The images, typical of museums and retail catalogs, have a small number of foreground objects on a generally separable background. Figure 4 shows example results. Even in this domain, robust algorithms are required because of the textured and variegated backgrounds.

We also provide semiautomatic tools for identifying objects. One is an enhanced flood-fill technique. Flood-fill methods, found in most photo-editing programs, start from a single object pixel and repeatedly add adjacent pixels whose values are within some given threshold of the original pixel. Selecting the threshold, which must change from image to image and object to object, is tedious. We automatically calculate a dynamic threshold by having the user click on background as well as object points. For reasonably uniform objects that are distinct from the background, this operation allows fast object identification

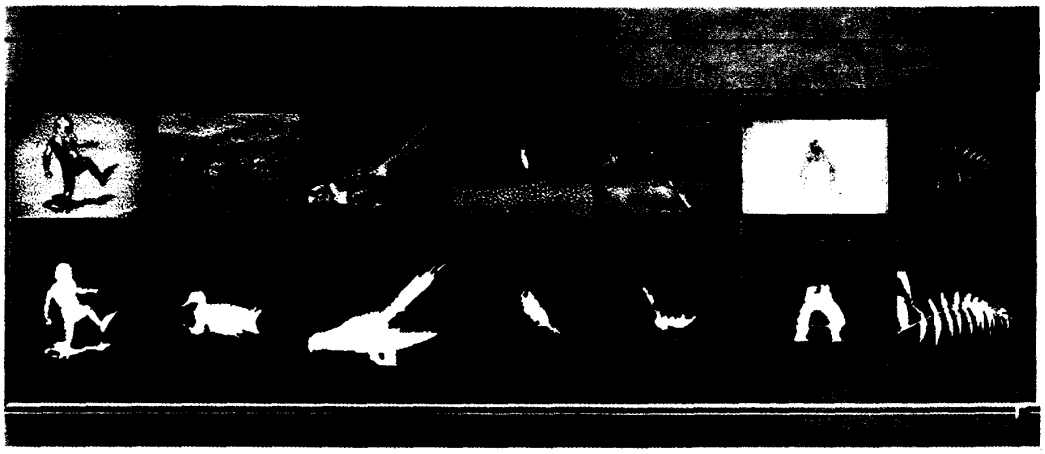

Figure 4. Top row is the original image. Bottom row contains the automatically extracted objects using a foreground/background model. Heuristics encode the knowledge that objects tend to be in the center of the picture.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

without manually adjusting a threshold. The example in Figure 3 shows an object, a fox, identified by using only a few clicks.

We designed another outlining tool to help users track object edges. This tool takes a user-drawn curve and automatically aligns it with nearby image edges. Based on the "snakes" concept developed in recent computer vision research, the tool finds the curve that maximizes the image gradient magnitude along the curve.

The spline snake formulation we use allows for smooth solutions to the resulting nonlinear minimization problem. The computation is done at interactive speeds so that, as the user draws a curve, it is "rubber-banded" to lie along object boundaries.

Figure 5. Scene cuts automatically extracted from a 1,148-frame sales demo from Energy Productions.

U-M-I
BEST COPY AVAILABLE

Video data

For video data, database population has three major components:

- shot detection,
- representative frame creation for each shot, and
- derivation of a layered representation of coherently moving structures/objects.

Shots are short sequences of contiguous frames that we use for annotation and querying. For instance, a video clip may consist of a shot smoothly panning over the skyline of San Francisco, switching to a panning shot of the Bay meeting the ocean, and then to one that zooms to the Golden Gate Bridge. In general, a set of contiguous frames may be grouped into a shot because they

- depict the same scene,
- signify a single camera operation,
- contain a distinct event or an action like a significant presence and persistence of an object, or
- are chosen as a single indexable entity by the user.

Our effort is to detect many shots automatically in a preprocessing step and provide an easy-to-use interface for the rest.

SHOT DETECTION. Gross scene changes or scene cuts are the first indicators of shot boundaries. Methods for detecting scene cuts proposed in the literature essentially fall into two classes: (1) those based on global representations like color/intensity histograms without any spatial information, and (2) those based on measuring differences between spatially registered features like intensity differences. The former are relatively insensitive to motion but can miss cuts when scenes look quite different but have similar distributions. The latter are sensitive to moving objects and camera. We have developed a method that combines the strengths of the two classes of detection. We use a robust normalized correlation measure that allows for small motions and combines this with a histogram distance measure.[5] Results on a few videos containing from 2,000 to 5,000 frames show no misses and only a few false cuts. Algorithms for signaling edit effects like fades and dissolves are under development. The results of cut detection on a video containing commercial advertisement clips are shown in Figure 5.

Shots may also be detected by finding changes in camera operation. Common camera transformations like zoom, pan, and illumination changes can be modeled as unknown affine 2×2 matrix transformations of the 2D image coordinate system and of the image intensities themselves. We have developed an algorithm[6] that computes the dominant global view transformation while it remains insensitive to nonglobal changes resulting from independently moving

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.