

# Storage and Retrieval of Feature Data for a Very Large Online Image Collection<sup>‡</sup>

Chad Carson and Virginia E. Ogle  
Computer Science Division, University of California at Berkeley, Berkeley CA 94720  
carson@cs.berkeley.edu, ginger@cs.berkeley.edu

## Abstract

*As network connectivity has continued its explosive growth and as storage devices have become smaller, faster, and less expensive, the number of online digitized images has increased rapidly. Successful queries on large, heterogeneous image collections cannot rely on the use of text matching alone. In this paper we describe how we use image analysis in conjunction with an object relational database to provide both textual and content-based queries on a very large collection of digital images. We discuss the effects of feature computation, retrieval speed, and development issues on our feature storage strategy.*

## 1 Introduction

A recent search of the World Wide Web found 16 million pages containing the word “gif” and 3.2 million containing “jpeg” or “jpg.” Many of these images have little or no associated text, and what text they do have is completely unstructured. Similarly, commercial image databases may contain hundreds of thousands of images with little useful text. To fully utilize such databases, we must be able to search for images containing interesting objects. Existing image retrieval systems rely on a manual review of each image or on the presumption of a homogeneous collection of similarly-structured images, or they simply search for images using low-level appearance cues [1, 2, 3, 4, 5]. In the case of a very large, heterogeneous image collection, we cannot afford to annotate each image manually, nor can we expect specialized sets of features within the collection, yet we want to retrieve images based on their high-level content—we would like to find photos that contain certain objects, not just those with a particular appearance.

## 2 Background

The UC Berkeley Digital Library project is part of the NSF/ARPA/NASA Digital Library Initiative. Our goal is to develop technologies for intelligent access to massive, distributed collections comprising multiple-terabyte databases of photographs, satellite images, maps, and text documents.

---

*Copyright 1996 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.*

**Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**

---

<sup>‡</sup>This work was supported by an NSF Digital Library Grant (IRI 94-11334) and an NSF graduate fellowship for Chad Carson.

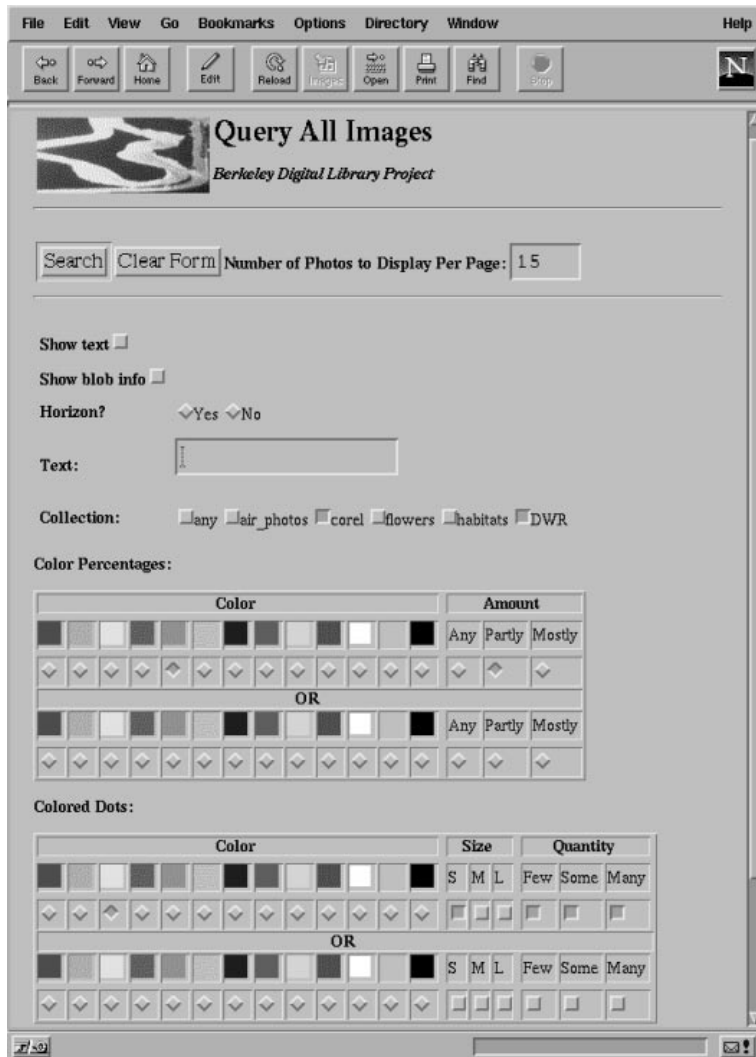


Figure 1: WWW query form set up for the “sailing and surfing” query.

In support of this research, we have developed a testbed of data [6] that as of this writing includes about 65,000 scanned document pages, over 50,000 digital images, and several hundred high-resolution satellite photographs. This data is provided primarily by public agencies in California that desire online access to the data for their own employees or the general public. The testbed includes a large number of (text-based) documents as well as several collections of images such as photos of California native species and habitats, historical photographs, and images from the commercial Corel photo database. The image collection include subjects as diverse as wildflowers, polar bears, European castles, and decorated pumpkins. It currently requires 300 GB of storage and will require more than 3.4 TB when it is complete. Image feature data and textual metadata are stored in an Illustra database.

All data are now being made available online using access methods developed by the Berkeley Digital Library project. The data is accessible to the public at <http://elib.cs.berkeley.edu/> via forms, sorted lists, and search engines. Image queries can rely on textual metadata alone, such as the photographer’s name or the photo’s caption, or they can employ feature information about the image, such as color information or the presence of a horizon in the image (see figure 1).

### 3 Content-Based Querying

Most work on object recognition has been for fixed, geometric objects in controlled images (for example, machine parts on a white background), which is not very useful for image retrieval in a general setting such as ours. However, a few researchers have begun to work on more general object recognition [7].

The current focus of our vision research is to identify objects in pictures: animals, trees, flowers, buildings, and other kinds of “things” that users might request. This focus is the direct result of research by the user needs assessment component of the Digital Library project [8]. Interviews were conducted at the California Department of Water Resources (DWR), which is a primary source of the images used in the Digital Library project testbed as well as one of its main users. Employees were asked how they would use the image retrieval system and what kinds of queries they would typically make. The DWR film library staff provided a list of actual requests they had handled in the past, such as “canoeing,” “children of different races playing in a park,” “flowers,” “seascapes,” “scenic photo of mountains,” “urban photos,” “snow play,” and “water wildlife.”

As the user needs assessment team discovered, users generally want to find instances of high-level concepts rather than images with specific low-level properties. Many current image retrieval systems are based on appearance matching, in which, for example, the computer presents several images, and the user picks one and requests other images with similar color, color layout, and texture. This sort of query may be unsatisfying for several reasons:

- Such a query does not address the high-level content of the image at all, only its low-level appearance.
- Users often find it hard to understand why particular images were returned and have difficulty controlling the retrieval behavior in desired ways.
- There is usually no way to tell the system which features of the “target” image are important and which are irrelevant to the query.

Our approach is motivated by the observation that high-level objects are made up of regions of coherent color and texture arranged in meaningful ways. Thus we begin with low-level color and texture processing to find coherent regions, and then use the properties of these regions and their relationship with one another to group them at progressively higher levels [9]. For example, an algorithm to find a cheetah might first look for regions which have the color and texture of cheetah skin, then look for local symmetries to group some regions into limbs and a torso, and then further group these body segments into a complete cheetah based on global symmetries and the cheetah body plan.

## 4 Implementation

### 4.1 Finding Colored Dots

As a first step toward incorporating useful image features into the database, we have searched for isolated regions of color in the images. Such information can be useful in finding such objects as flowers and people.

We look for the following 13 colors in each image: red, orange, yellow, green, blue-green, light blue, blue, purple, pink, brown, white, gray, and black. We chose these colors because they match human perceptual categories and tend to distinguish interesting objects from their backgrounds [10].

We use the following algorithm to find these “colored dots”:

1. Map the image’s hue, saturation, and value (HSV) channels into the 13 perceptual color channels.

- Filter the image at several scales with filters which respond strongly to colored pixels near the center of the filter but are inhibited by colored pixels away from the center. These filters find isolated dots (such as in a starry sky) and ignore regions that are uniform in color and brightness (such as a cloudy sky).
- Threshold the outputs of these filters and count the number of distinct responses to a particular filter.

Responses at a coarse scale indicate large dots of a particular color; responses at finer scales indicate smaller dots. The number of dots of each color and size is returned, as is the overall percentage of each color in the image. A  $13 \times 6$  matrix is generated for each image. Rows in the matrix represent the 13 colors that are identified. Six integers are associated with each color: the percentage of the image which is that color, and the number of very small, small, medium, large, and very large dots of that color found. (These sizes correspond to dots with radii of approximately 4, 8, 16, 32, and 64 pixels, respectively, in  $128 \times 192$  pixel images.)

While these dot counts and percentages contain no information about high-level objects, they are a first step toward purely image-based retrieval. A number of combinations of the dot and percentage data yield interesting results; the following are a few examples:

Query	Percentages	Dots <sup>a</sup>	Text	Datasets	Precision <sup>b</sup>
Sailing & Surfing (fig. 2)	blue-green > 30%	# VS yellow $\geq 1$	—	Corel, DWR	13/17
Pastoral Scenes (fig. 3)	green > 25% AND light blue > 25%		—	all	85/93
Purple Flowers (fig. 4)		# S purple > 3	—	all	98/110
Fields of Yellow Flowers		# VS yellow > 15	—	all	63/74
Yellow Cars		# L yellow $\geq 1$ OR # VL yellow $\geq 1$	“auto” <sup>c</sup>	all	6/7
People (fig. 5)	orange > 1%	# L pink $\geq 1$ OR # VL pink $\geq 1$	—	Corel, DWR	19/69

<sup>a</sup>The different dot sizes (very small, small, medium, large, and very large) are abbreviated VS, S, M, L, and VL, respectively.

<sup>b</sup>“Precision” is the fraction of returned images that contain the intended concept. “Recall,” the fraction of images in the database containing the intended concept that are returned, is not a feasible measure in this case because we do not know how many instances of the intended concept are in the database.

<sup>c</sup>There are 132 “auto” images; restricting the query to images with large yellow dots reduces the number to seven.

## 4.2 Storage of Feature Data

Because of the size of the image collection and its associated metadata, we must use a database to manage both textual and image content information. Our chief priority is to store this data in such a way as to facilitate the fastest possible retrieval time in order to make rapid online browsing feasible. Therefore, we do not store the images themselves in the database, and we store metadata in a way that circumvents the need for joins on two or more tables. In addition, because image content analysis is time-consuming and computationally expensive, we do this analysis ahead of time and store the results in the database rather than using run-time functionality provided by the database. Another concern



Figure 2: Representative results for the “sailing and surfing” query. (Color images are available at <http://elib.cs.berkeley.edu/papers/db/>)



Figure 3: Representative results for the “pastoral” query.



Figure 4: Representative results for the “purple flowers” query.



Figure 5: Representative results for the “people” query.

related to image analysis is the need to support continual development of new analysis techniques and new feature data. We want to be able to add new features and modify existing features painlessly as our vision research progresses. In this section we describe how our approach to storing image feature data meets these goals.

Each of the five image collections is stored in its own table with its own particular attributes. The collection of DWR images has 24 textual attributes per image, including a description of the image, the DWR-defined category, subject, and internal identification numbers. The wildflowers table contains 14 attributes per image such as common name, family, and scientific name. The Corel stock images have very little metadata: an ID number, a disk title such as “The Big Apple,” a short description, and up to four keywords such as “boat, people, water, mountain.” The various image collections have very few textual attributes in common, other than a unique ID assigned by the Digital Library project and at least a few words of textual description from the data provider. Given the diversity of the overall collection and the likelihood of acquiring additional dissimilar image collections in the future, we do not want to support a superset of all image attributes for all the collections in one table. In addition, we have found that most users of our system want to direct a fairly specific query to a particular collection.

On the other hand, the addition of image feature data presents a more homogeneous view of the collection as a whole. Using image feature information to find a picture of sailboats on the ocean does not require any collection-specific information. Our approach is to support both text-based queries directed to a specific collection at a fine granularity (“find California wildflowers where common name = ‘morning glory’”) and text/content-based queries to the entire collection (“find pictures that are mostly blue-green with one or more small yellow dots”). The separate tables for each collection are used for collection-specific queries, while collection-wide queries can be directed to an aggregate table of all images. This supertable contains selected metadata for every image in the repository: the collection name, the unique ID, a “text soup” field which is a concatenation of any available text for that image, and the feature data.

We have experimented with different ways of storing the types of feature data that have been developed so far, and we continue to try different techniques as new features are developed. Storage of Boolean object information, such as the presence or absence of a horizon in the image, is straightforward; we simply store a Boolean value for a “horizon” attribute. As our vision research proceeds and new kinds of objects can be identified, they can be concatenated onto an “objects” attribute string, so that each image has just one list—the objects that were found in that image. In this manner, we eliminate the need to record a “false” entry for each object not found in an image. This text string can be indexed, and retrieval is accomplished using simple text matching. However, more complex color and texture features, such as colored dot information, require careful planning in order to ensure fast retrieval, development ease, and storage efficiency. Interestingly, the complexity of the stored feature data is inversely related to the capability of the image analysis system: as computer vision systems become more adept at producing high-level output (e.g., “flower” instead of “yellow dot”), the question of storage and retrieval becomes simpler, because the level of detail of the stored information more closely matches the level of detail of desired queries.

## Storing Image Features as Text

In general, we store image feature data as text strings, and we use text substring matching for retrievals. Dot information is stored in one text field per image. Any nonzero number of dots in an image is categorized as “few,” “some,” or “many” and stored in this field, separated by spaces. For example, a picture of a sky with clouds might have a few large white dots and a large amount of blue, so its dot field would be “`mostly_blue large_white_few`.”

We have found that storing feature data as text yields the best results in terms of development ease, extensibility, and retrieval speed. We have experimented with other methods, such as storing dots as integer or Boolean values, and we have considered a compact encoding scheme for the feature data in order to save storage space and possibly cut down on retrieval time. But conservation of storage space is not a high priority for our project, and we have found that for fast retrieval time the use of text is satisfactory.

There are several advantages to using text instead of other data types. Most images have few significant objects and only two to five significant colors; each color typically has just a few of the dot attributes represented. The current implementation of dots would require 78 ( $13 \times 6$ ) integer values, and most of them would be zero. Using one dots text string per image allows us to store only the features that are present in that image. This has an added benefit during the development stage, when vision researchers are testing their results on the image database—feature data can be concisely displayed in a readable form on the results page with little effort on the developer’s part.

Using text also means that incremental changes to stored feature data do not require elaborate re-encoding or new attribute names. Text-based queries are simple to construct because there is just one dots field, as illustrated in the following example:

To find an image with “any kind of white dots” using text, we simply use wildcards in the select statement:

```
where dots like '%white%'
```

The equivalent integer expression requires five comparisons:

```
where VS_white  $\geq$  1 or S_white  $\geq$  1 or M_white  $\geq$  1 or L_white  $\geq$  1 or VL_white  $\geq$  1
```

Integer-based queries must be more carefully constructed to make sure that all possibilities are included in each expression. Such factors contribute to a faster development time if a text-based method is used, a bonus for a system like ours that is continually changing.

## 5 Future Directions

In the future we plan to investigate more efficient ways to store numerical feature data such as colored dots. However, as our image analysis research progresses, we expect to be able to use low-level feature information (shape, color, and texture) to automatically identify higher-level concepts in the images, such as trees, buildings, people, animals of all kinds, boats, and cars. As high-level information like this becomes available, the need to store low-level features like dots will decrease.

Currently most of the feature data we have developed is stored in a single table—the supertable that includes all the images in the collection. Although queries on this table can include text and can be directed to individual collections, no categorization of text is provided, because the primary purpose of the form is to make content-based queries. We plan to extend the content-based capability to the query forms for each individual collection so that users who know that particular collection can take advantage of the stored feature data. One collection that we think will benefit greatly from the use of content-based queries is the California wildflower collection. Users will be able to request pictures of a named flower in a particular color, such as “blue morning glories and not white morning glories,” or even search for the names of flowers using color cues alone: “pink flowers with yellow centers” and “flowers with large purple blossoms.”

## 6 Acknowledgments

We would like to thank David Forsyth, Jitendra Malik, and Robert Wilensky for useful discussions related to this work.



## References

- [1] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, et al. Query by image and video content: the QBIC system. *IEEE Computer*, 28(9):23–32, Sep 1995.
- [2] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao-fe Shu. The Virage image search engine: An open framework for image management. In *Storage and Retrieval for Still Image and Video Databases IV*. SPIE, Feb 1996.
- [3] U. Shaft and R. Ramakrishnan. Content-based queries in image databases. Technical Report 1309, University of Wisconsin Computer Science Department, Mar 1996.
- [4] Michael Swain and Markus Stricker. The capacity and the sensitivity of color histogram indexing. Technical Report 94-05, University of Chicago, Mar 1994.
- [5] A.P. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *Int. Journal of Computer Vision*, to appear.
- [6] Virginia E. Ogle and Robert Wilensky. Testbed development for the berkeley digital library project. *D-lib Magazine*, Jul 1996.
- [7] J. Ponce, A. Zisserman, and M. Hebert. *Object representation in computer vision—II*. Springer LNCS no. 1144, 1996.
- [8] Nancy Van House, Mark H. Butler, Virginia Ogle, and Lisa Schiff. User-centered iterative design for digital libraries: The cypress experience. *D-lib Magazine*, Feb 1996.
- [9] J. Malik, D. Forsyth, M. Fleck, H. Greenspan, T. Leung, C. Carson, S. Belongie, and C. Bregler. Finding objects in image databases by grouping. In *International Conference on Image Processing (ICIP-96), special session on Images in Digital Libraries*, Sep 1996.
- [10] G. Wyszecki and W.S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. Wiley, second edition, 1982.