Indexing Flower Patent Images Using Domain Knowledge

Madirakshi Das, R. Manmatha, and Edward M. Riseman, University of Massachusetts

THE AUTHORS USE A FLOWER-PATENTS DATABASE TO DESCRIBE A NEW APPROACH TO INDEXING A SPECIALIZED DATABASE. THEY USE THE COLOR AND SPATIAL DOMAIN KNOWLEDGE AVAILABLE FOR THE DATABASE FOR SEGMENTING THE FLOWER REGIONS FROM THE BACKGROUND AUTOMATICALLY. THE RETRIEVAL SYSTEM PROVIDES A PERCEPTUALLY CORRECT RETRIEVAL WITH NATURAL LANGUAGE AND EXAMPLE IMAGE QUERIES.

THE ADVENT OF THE INFORMATION revolution has enormously increased the amount of information that people and organizations must handle. Using this information effectively requires tools to manage this information, including those for searching, retrieving, and classifying it. Several good search engines exist for text in ASCII form. However, no good tools of comparable performance for retrieving images are available yet.

Traditionally, people have manually annotated image databases using textual keywords. They then retrieve images based on the manually assigned keywords. Manual annotation is slow, expensive, and impractical for today's large image databases. In addition, manual annotations suffer from many limitations; they can be inaccurate (especially for large databases), and they cannot encode all the information present in an image. Thus, there has been much interest recently in content-based image retrieval, where the goal is to find images in the database that are partially or completely similar to a query or example image.

This article explains how to query a database of flower patent images using both an example flower image and color names. This database consists of images that have been digitized from photographs submitted as a part of applications for flower patents to the US Patents and Trademark Office. This database must be queried by both example images and color name, so that both those checking new patent applications and those buying patents for cultivation can use it.

Unlike many other color-based retrieval systems,[1,2] our system ensures that the indexing process uses only the flower's color rather than colors in the entire image (see the "Literature survey and related work" sidebar at the end of the article). The retrieval system links a natural-language color classification derived from the ISCC-NBS color system and the Windows X color names to the flower's color. Users may query the database by either using natural-language queries to describe a flower's color or by providing an example image of the flower. Our goal is to find a way to use the domain knowledge available for specialized databases to provide better retrieval performance than do general-purpose retrieval strategies.

Background on image retrieval

The basic step toward meaningful retrieval is to ensure that the image descriptions used to index the database relate to the image's semantic content. This requirement is difficult to meet for content-based image retrieval. Unlike text where the natural unit, the word, has a semantic meaning, the pixel—which is the natural unit of an image—by itself has no semantic meaning. In images, meaning is found in objects and their relationships. However, segmenting images into such meaningful units (objects) is, in general, an unsolved problem in computer vision. Fortunately, we can often directly correlate many image attributes, such as color, texture, shape, and appearance, with the problem's semantics. For instance, logos or product packages (for example, a box of Tide laundry detergent) have the same color wherever they are found. A leopard's coat has a unique texture, while Abraham Lincoln's appearance is uniquely defined. Often, we can use these image attributes to index and retrieve images.

However, we must use these attributes with care if they are to correlate with the semantics of the problem. For example, many image-retrieval systems use color to retrieve images from general collections.[1,2] A picture of a red bird used as a query might retrieve not only pictures of red parrots but also pictures of red flowers and red cars. Clearly, this is not a meaningful retrieval as far as most users are concerned. If, however, the collection of images were limited to those containing birds, the results retrieved would be restricted to birds and probably would be far more meaningful from the user's viewpoint.

Although many image-retrieval algorithms have been focused on retrieving images from general image collections, we believe that restricting image-retrieval to specialized collections of images or to specific tasks will be more successful. The restriction to specific domains does not make the task any less interesting or useful. In fact, some of the most successful work in the area of image-retrieval has been in specialized methods for retrieving faces similar to a face-image query from a database of face images.[3]

The nature of the task often modifies the approach taken to image-retrieval. When flower images are indexed, a flower of a different color should not be considered to be a match. However, in trademark retrieval, color plays no role. A trademark is considered identical to another trademark even if their colors are different. Trademarks are a good example of a task where all types of images occur but where the task is very specific (to find trademarks that are visually similar). Trademark images have text associated with them, which permits searching both on the visual content as well as on the text. There has been some work on interfacing text and image-retrieval to retrieve trademarks.[4] The use of text retrieval allows additional constraints. For example, two visually identical trademark images are considered conflicts only if they are used for similar goods and services.

We have developed a better approach for indexing a specialized database by exploiting the knowledge available for the domain covered by a flower patents database. Although all images in the database depict flowers, there is no uniformity in the size and location of the flowers in the image or the image backgrounds, as Figure 1 shows. There are two main challenges with this application: segmenting the flower from the background, and describing the flower's color in a way that matches human perception and allows flexible querying by example and by natural-language color names.

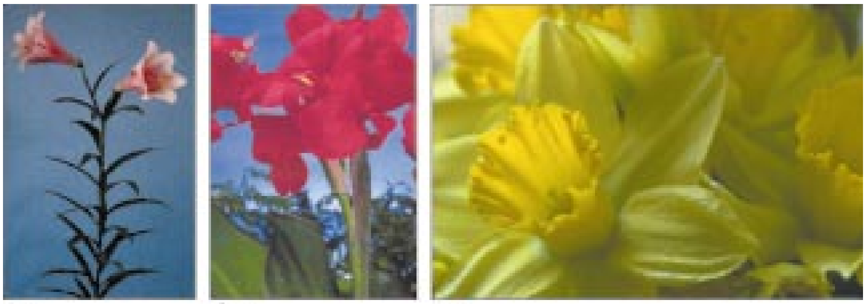

(a) (b) (c)

Figure 1. Example of database images showing different types of backgrounds.

We would like to use this domain's characteristics to automate the segmentation and indexing process. Most of the domain knowledge is in the form of natural-language statements; translating them into rules to build automated algorithms is difficult. For example, like most natural subjects, a lot of color-based domain knowledge is known for the flower domain: for example, flowers are rarely green, black, gray, or brown. Examples of information in other domains would be facts such as the following: mammals are rarely blue, violet, or green; and outdoor scenes often have blue and white skies and green vegetation. However, we can use such information effectively only when a mapping from the 3D color space to natural-language color names is available.

There has been work on perceptual organization of the color space in the area of image indexing without mapping the perceptual groups obtained to natural-language color names.[5] These approaches are not very useful for translating natural-language rules about color into computer-usable information. However, they provide good indexing tools when the object of interest has been presegmented from the background. In the reverse approach, applications such as face identification using skin tones and automatic target recognition (where the part of the color space that corresponds to the object of interest is identified) have mapped color-domain knowledge onto the 3D color space.[6] Modeling the distribution of color points in objects is an important issue in this approach. E. Saber and others model the set of pixels in each natural object as a Gaussian probability density function in annotating natural scenes.[7] Yihong Gong and Masao Sakauchi detect regions corresponding to a specified color model.[8] We have constructed a mapping to a natural-language color name space using color names from the ISCC-NBS system and the color names defined in the Windows X system for this purpose.[9]

Work in the areas of color image segmentation and modeling of the appearance of colored objects is also relevant to our work. Researchers have used color histograms[10] in different color spaces,[11,12] for color image segmentation. However, none of the systems above identifies the object of interest; hence, these systems cannot distinguish the background elements from the foreground elements. Q. Huang and others have studied automatic foreground and background disambiguation based on multiple features such as color, intensity, and edge information,[13] but these techniques assume relatively smooth backgrounds and objects with sufficient contrast. Because we would like to use color domain knowledge, the color space needs to be mapped to colors as perceived by humans.

We have developed an iterative segmentation algorithm that uses available domain knowledge to provide a hypothesis, marking some colors as background colors and then testing the hypothesis by eliminating those colors. Evaluating the remaining image provides feedback about the correctness of the hypothesis, and the algorithm generates a new hypothesis when necessary after restoring the image to its earlier state.

Segmenting the flower from the background

The first step in indexing the flower patent database by flower color is to extract the flower from the background. There is no general solution to the problem of extracting the object of interest from an image. However, for a specialized domain such as flowers, we can use domain knowledge to automatically extract a region from the image that has a high probability of being a flower region. The types

red	reddish orange	reddish purple
reddish brown	green	bluish green
purplish red	brown	greenish blue
purplish pink	yellow green	orange
orange yellow	blue	yellowish brown
yellow	purplish blue	yellowish pink
olive brown	pink	greenish yellow
yellowish green	violet	brownish pink
olive	purple	brownish orange

(a)

very pale	very light	brilliant	vivid
pale	light	grayish	moderate
strong	dark grayish	dark	deep
blackish	very dark	very deep	

(b)

red	green	brown	brown
blue	purple	yellow	yellow
violet	black	gray	gray

(c)

Figure 2. The ISCC-NBC system: (a) hue names; (b) hue modifiers; and (c) color classes derived from grouping hue names and adding the neutral colors.

of information available for this application can be categorized into *color-based* and *spatial* domain knowledge. Because color-based domain knowledge is available for natural-language color descriptions, and because providing color name-based retrieval is also one of our goals, we have designed the retrieval system to map the color space to commonly used color names. The next task is to segment the image using both color and spatial domain knowledge. These steps constitute the offline processing phase of indexing the database.

Mapping from color space to names. To be useful, tables must map points on a 3D color space to color names in a way that agrees with human perception of colors. We use two sources for names:

- the ISCC-NBS (Inter-Society Color Council–National Bureau of Standards) color system, which produces a dense map from the Munsell color space to names; and
- the colors defined by the Windows X system, which provides a sparse mapping from the RGB (red-green-blue) space to 359 names.

The ISCC-NBS system uses a standard set of base hues (see Figure 2a) and generates 267 color names using hue modifiers (see Figure 2b). This gives a color system that we can easily decompose into a hierarchy of colors where we can use the full color name, partial names, base hues, or coarser classes comprising groups of base hues (see Figure 2c).

The color names in the ISCC-NBS system often have simpler, commonly used alternatives; for example, "very pale yellowish white" in the ISCC-NBS system is "ivory," and "light brownish yellow" is "khaki." Our color tables extract the simpler names, such as "ivory" and "khaki," from the definitions in the Windows X system; these color names often are derived from commonly known objects of the same color.

The raw image data available encodes color in the RGB space using 24 bits per pixel. This produces 2^{24} possible colors, which is far more than the number of distinct colors that a human can perceive. The distances between points in this space also do not represent the perceived distances between colors. We have used the HSV (Hue-Saturation-Value) color space,[14] discretized into $64 \times 10 \times 16$ bins, as an intermediate space to reduce the number of colors and to keep perceptually similar colors in the same neighborhood. The color tables map each point on the discretized HSV space to a color defined in the Windows X system. They map points with no exact map to the nearest color name, using the city block measure to compute distances. They also map each point to the ISCC-NBS name (see Figure 3). We use the ISCC-NBS name to produce a color hierarchy so that queries can be general (for example, blue) or specific (for example, pale blue). We also use this color structure to segment the flower from its background. Using color names from two sources improves the chances of finding a name that matches the user's natural-language query.

RGB ($256 \times 256 \times 256$)	(245, 195, 40)	(233, 150, 122)
HSV ($64 \times 10 \times 16$)	(7, 8, 15)	(2, 5, 14)
Xcolornames (359)	Goldenrod2	Dark salmon
ISCC-NBS colornames (267)	Strong yellow	Dark brownish pink
Color classes (12)	Yellow	Pink

Figure 3. Example of the color representations used.(RGB is red, green, blue; HSV is the Hue-Saturation Value; and ISCC-NBS is the Inter-Society Color Council–National Bureau of Standards.)

Iterative segmentation with feedback. We need to segment the regions corresponding to flowers from the rest of the image before we can accurately describe the flower's colors. We isolate the flower regions from the background using domain knowledge about the color of flowers and knowledge about the distribution of background regions in photographs.

Using domain knowledge. Because we have mapped the 3D color space to natural-language color names, we can use color-based domain knowledge of the type discussed earlier. We can eliminate most of the frequently occurring elements of the background in flower images by deleting pixels from color classes that do not represent flower colors. Black and gray are mostly from the image's shadow regions; brown pixels come from shadows as well as branches and soil, and green pixels are from foliage and vegetation.

In addition to color-based domain knowledge, we can derive additional rules from domain knowledge about the spatial distribution of the flower and background in the database images. An observation that can help identify background regions is that background colors are usually visible along the periphery of the image. If this observation were always true, the automatic indexing system could detect the background color with certainty by analyzing the colors present in the margins of the image. However, as Figure 1 shows, the image's margins can fall into one of three types:

- the flower may be totally embedded in the background;
- the background and flower regions may interlace along the margins; or
- the flower may fill the whole image.

We can derive some useful guidelines from the fact that the images in the database are photographs depicting flowers. First, the flower itself will occupy a reasonable part of the image. Second, if flowers are present near the image boundaries, they will be present throughout the image because the object of interest (the flower, in this case) is unlikely to be confined solely to the image periphery. The background can have other colored objects, but they usually do not dominate the main subject, the flower.

We also know that the flower images were submitted as part of a patent application. Therefore, we can conclude that there is a single type of flower, although the image may contain many types of flowers. Because of this, we can select one prominent segment as a flower region out of multiple segments without losing information. The goal is to isolate a region in the image where we can obtain a good description of the flower's color, not to detect all flower regions in the image.

Segmentation strategy. Our approach to extracting a region that has a high probability of being a part of a flower is to use the knowledge discussed above to successfully eliminate background colors until the remaining region contains only flower areas. This entails generating a hypothesis identifying the background colors. However, because the hypothesis could be wrong, we use a feedback mechanism from the segmentation results to redirect our choice of background colors and try a different hypothesis.

We use the connected-components algorithm whenever we need to identify segments in the image, where each segment is a connected component. The connected components algorithm runs after converting the image to a binary image, where the only two classes of pixels are those that have been eliminated and those that remain. Figure 4 outlines the algorithm used to produce a segment for estimating the flower color.

The indexing system labels image pixels by their color classes as well as by their nearest Windows X system color-name. We use a coarse-to-fine strategy when using the color labels: we use the color class description first; we use finer color-name distinctions only when necessary. In the first step, the indexing system eliminates pixels belonging to the color classes black, gray, brown and green, because these are nonflower colors, and the remaining image is segmented after binarization.

We use two criteria to evaluate whether a segment produced is valid: it should be of a minimum size, based on the size of the largest segment obtained after deleting the nonflower color classes, and its centroid should fall within the image's central region as defined in Figure 5a. (These requirements are based on the domain knowledge discussed in the previous section.)

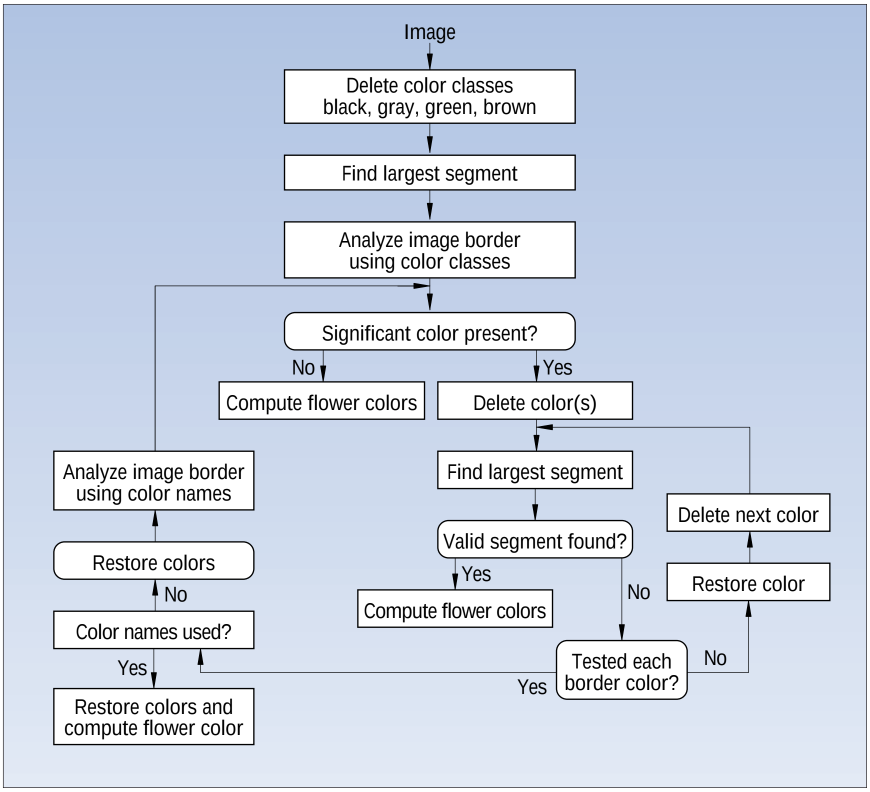


Figure 4. Overview of the indexing system, showing steps in the automatic computation of the flower color during offline processing.

If there is more than one valid segment, only the largest segment is retained. This step deletes small patches of extraneous colors from other colored objects in the image—for example, the rock in Figure 6. Because the flower is the image's dominant subject, the largest segment has the highest probability of being a flower region.

Only the pixels constituting the largest valid segment are retained; the rest are eliminated. In flower images taken in natural surroundings from a distance, this process is sufficient for producing a good flower segment (see Figure 6).

Further processing is required when the largest segment contains background colors in addition to the flower regions. First, the indexing system reduces the image by retaining the pixels covered only by the largest seg-

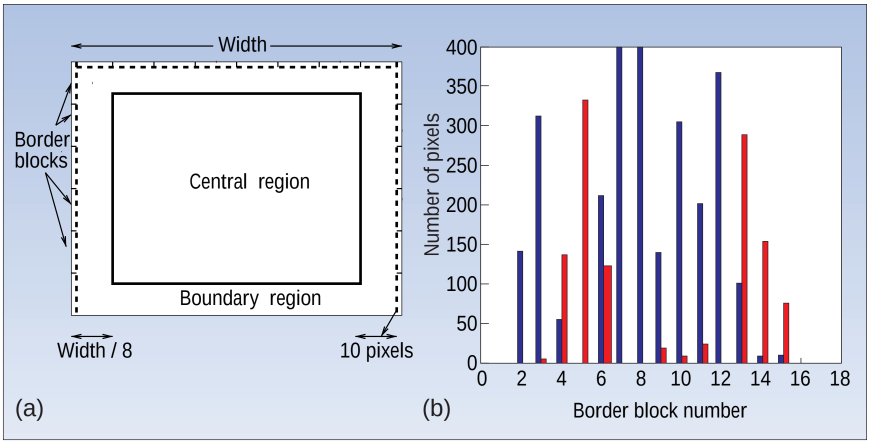
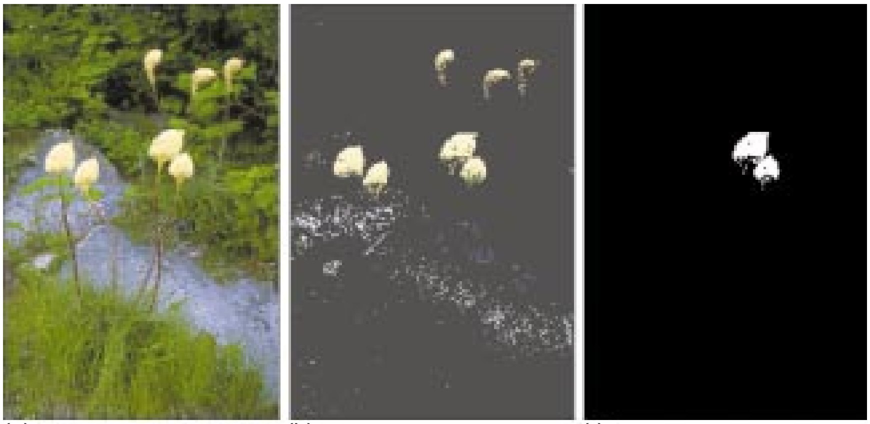
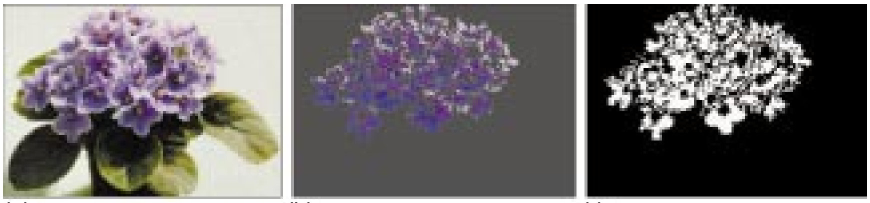


Figure 5. (a) Definitions of image regions; (b) color distribution in border blocks of the canna image in Figure 1b.



(a) (b) (c)

Figure 6. Detecting a reliable flower region: (a) original image; (b) image left after deleting nonflower colors; (c) largest valid segment.



(a) (b) (c)

Figure 7. Background elimination: (a) original image; (b) image left after deleting nonflower colors and the background color (white); (c) largest valid segment.

ment. Then, the indexing system detects background colors by analyzing the color composition along the image margins. These margins are divided into border blocks, as Figure 5a shows. The indexing system computes the distribution of color classes in these blocks and marks as background colors those colors that show substantial presence in more than half of the blocks. For example, Figure 5b shows the color distributions for the two color classes present in the border of the image in Figure 1b. From this distribution, blue is marked as a background color, since it is present in 11 out of 16 border blocks.

After eliminating all the pixels belonging to the colors hypothesized to be background colors, the indexing system computes the largest segment in the binarized image (see Figure 7). The indexing system then tests the validity of the segment to determine whether the choice of background colors was correct.

This method of detecting background colors is not guaranteed to produce correct results. It will fail for images of the type shown in Figure 1c, and may also fail for images such as the one shown in Figure 1c if there is sufficient overlap between the flower and the margin. An erroneous choice of background color can, in most cases, be detected from the segments generated after eliminating those pixels. In the case of the image type in Figure 1c, the hypothesis for the background color deletes the entire image. In the image type in Figure 1b, if the flower color is deleted instead of the background, only background pixels remain in the image. Because background tends to be scattered among the flower regions and along the margins, no connected components in the central region are usually large enough to be valid, while connected components near the boundary do not pass the centroid location test. So, the lack of valid segments indicates that the background color selection was wrong.

When the segmentation process reveals that the background color chosen was incorrect, the color is restored and the hypothesis that a color is a background color is tested separately, iterating through each of the colors present in the border region. Figure 8 shows the intermediate steps in detail. From the analysis of the segment's border obtained first, the indexing system eliminates the color class purple. This results in a segment whose centroid falls in the boundary region. A valid segment is found when purple is restored, and another segmentation is performed after the new hypothesis for background color (the class white) is eliminated.

If no valid segments are found after any of the color classes present in the border are eliminated, then the image is probably of the type in Figure 1c, and the flowers cover the full image. However, because we are looking at color classes, there is an alternative situation where the background is a different shade of the flower color and, thus, belongs to the same class. We test for this situation by labeling the pixels using color names rather than color classes and repeating the above procedure (see Figure 9). When the original image is labeled and segmented, the color class white is found to be the background color. However, deleting pixels of the color class white deletes the entire image. (The background does not appear to belong to the color class white in the figure, because the printed colors appear much more saturated than they actually are.) When we label the image using color names, the border block analysis gives the colors "honeydew" and "mint cream" (shades of white). Deleting these colors leaves the colors "lemon chiffon 3" and "ivory 3," which are also shades of white. The remaining image shown in Figure 9c produces a valid segment without any background.

When none of these trials eliminates the background, the indexing system assumes that the image contains only the flower colors and computes the description from the largest segment obtained after deleting the nonflower colors.

The segmentation strategy produces erroneous results only when the image's colored objects (excluding the nonflower colors) are more prominent than the flowers and when the flowers are located only along the image's margins. Both situations are improbable in the flower patents database.

Indexing and retrieval

The colors present in the segment identified as a flower region are used as features during retrieval from the flower database. The flower database indexing is based on the types of queries we would like to support. This includes queries using natural-language color names. Because a wide variety of names are available for querying, we index the images by using both the X names and the ISCC-NBS color names as keys to improve the likelihood of finding a name supplied by the user as the query in the database index. A third index table accesses the images by the color classes present in the images.

More than one color name is usually present in each color class contained in a flower region. The relative proportion of the different shades of the color affects the perceived color in the flower. So, the relative proportions of colors in the flower region is also an important factor to consider.

Query by name. When a color name is provided as a query, the retrieval system searches the X name index and the NBS color-name index for the query color name and its variants. The variants are produced by incompletely specified ISCC-NBS color names—and by the X naming system, because it uses increasing numbers to indicate darker shades of the original color. For example, medium purple 2, medium purple 3, and medium purple 4 are progressively darker shades of the original color medium purple. Because the user is unlikely to know the details of this nomenclature, a query of “medium purple” should consider all the shades of the color.

However, the users could also use one of the defined X or NBS color names to issue a specific query if they have knowledge of the valid names. In this case, the exact name from the indexes is used. The retrieval system ranks the retrieved images by proportion: it ranks the flower with a larger proportion of the query color ahead of a flower with a smaller proportion of the query color. If the query uses more than one name, the retrieval system returns a join (intersection) of the image lists retrieved for each of the query colors.

Query by example. When using a flower image as a query, the user expects a close color match with the flower shown in the query. In this case, separately searching for each color present and combining the lists often produces poor results. For example, a flower may appear to be an intermediate shade of pink because it has pixels of both a darker shade and a lighter shade. Separate retrieval using the two shades present will retrieve flowers that have both these shades, but flowers whose perceived shade does not match the query might be ranked high. This could happen because the retrieval system didn’t take into account the relative proportions of the two shades while ranking; therefore, relative proportions of the two shades in the top retrieved flower could be quite different from the query.

In this case, we must find a distance measure between the query flower and the retrieved flower that takes into account the relative proportions of various shades of a color class in the flower. We do this by computing an average color for each color class present in the query. We compute the HSV coordinates for each X color from its original RGB definition and store this information in the color tables. The retrieval system computes a weighted average of the HSV coordinates of the X colors present in a color class. The weights are proportional to the rel-

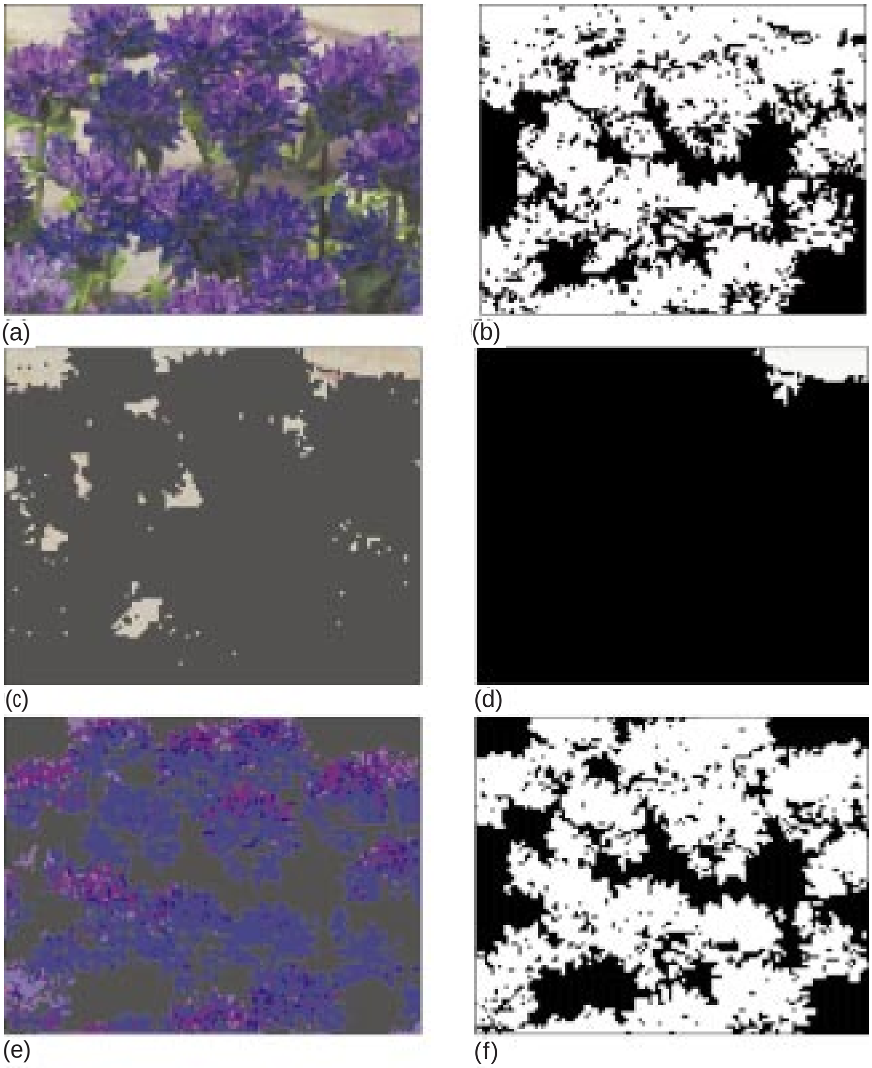


Figure 8. Recovery from erroneous background color selection: (a) original image; (b) segment found after deleting nonflower colors; (c) the result of deleting the color class purple, which was hypothesized to be a background color; (d) the largest segment obtained (which is not valid); (e) trying the new hypothesis that the color white is the background color; (f) the valid segment obtained.

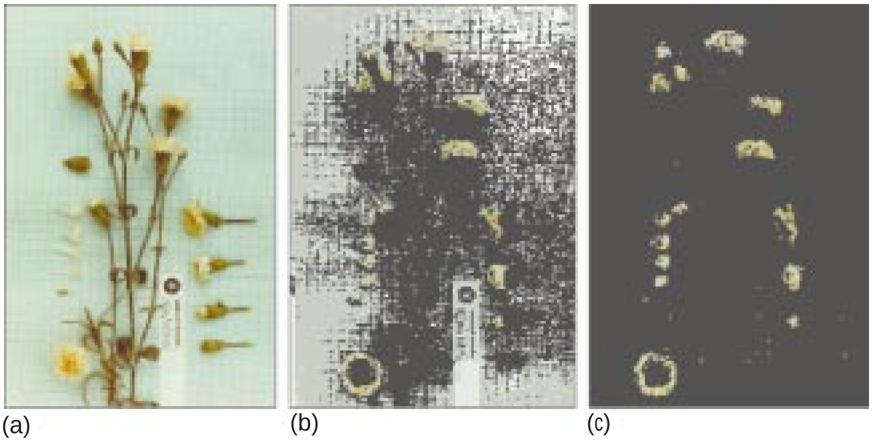


Figure 9. Using color names for labeling: (a) original image; (b) image left after deleting nonflower colors; (c) result of eliminating background colors based on color names.


(a) (b) (c)

Figure 10. Detecting images on the patent form: (a) scanned page; (b) image left after deleting background color; (c) segments found.

Figure 11. Images on which the segmentation algorithm produces errors.

ative proportion of the color in the flower segment. For example, for a flower that has color X1 (h_1, s_1, v_1) and color X2 (h_2, s_2, v_2) in proportion p_1 and p_2 in a class, the average color of the color class is

$$\left(\frac{p_1 h_1 + p_2 h_2}{p_1 + p_2}, \frac{p_1 s_1 + p_2 s_2}{p_1 + p_2}, \frac{p_1 v_1 + p_2 v_2}{p_1 + p_2} \right).$$

The retrieval system ranks each retrieved image on the basis of the city-block distance of its average color in each color class from the corresponding query color averages.

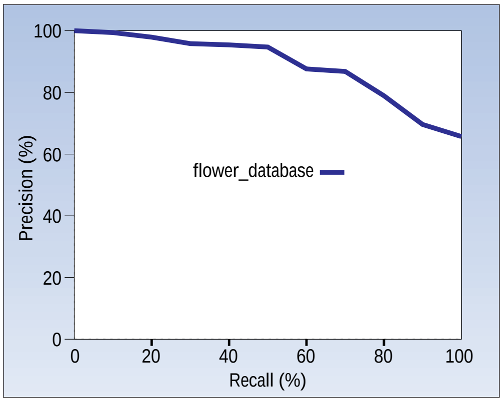


Figure 12. Recall-precision graph for 25 queries by example on the flower-patent database.

Experiments

The test flower database we are using contains 300 images. About 100 are from actual flower patents from the US Patent and Trademarks Office. To test the segmentation process, we have added 100 images from CD-ROM collections with complex backgrounds beyond those encountered in images from patent applications. We scanned the rest from catalogs of flowering plants and photographs—including a few images of colored fruits, which the retrieval system treats the same way as flowers.

The pages from the patent forms are of the type shown in Figure 10a; they contain both text and images. The indexing system detects the images from the patent forms, using the same strategy of deleting background colors and checking the remaining segments. However, in this case, one might find more than one segment of significant size, as shown in Figure 10c. We approximate these segments by rectangles and add to the database the cropped image corresponding to each segment.

For each database image, we checked the flower segment identified by the iterative segmentation algorithm and found only two possibly erroneous segmentation results for the images shown in Figure 11. The segment formed by the pink flowers did not pass the centroid test, and the indexing system selected the yellow flower region as the most significant segment. This is an image from the CD-ROM collection and is unlikely to be a part of a patent application. In the second image, the pale violet leaves of the water lily constituted the most significant segment; it may actually be the correct component of the patent, because the flower is given very little emphasis in the image.

We tested the retrieval results using 50 queries of different types. On 25 queries using color names, we verified that the retrieved flowers matched our perception of the color name used in the query. We did a more exhaustive evaluation for 25 queries, using example images. We identified the images relevant to the query by scanning the database, and we computed recall and precision measures. Figure 12 shows the recall-precision graph.[15] The average precision obtained was 88%, and the precision at 100% recall was 66%. The latter figure is important in this application because finding all relevant images is essential, even at the expense of checking a larger number of irrelevant ones.

Figure 13a shows the user interface for querying by color. Users can select the color class from the left frame of the interface, and the right frame displays that color's various shades along with their names. Users can search the database by color class or by selecting a particular shade of the color. In the snapshot of the interface shown in the figure, the color medium purple is selected. The bottom of the interface displays the retrieved images. Figure 13b shows the current interface for query by example. Users can select the example image by browsing through the database

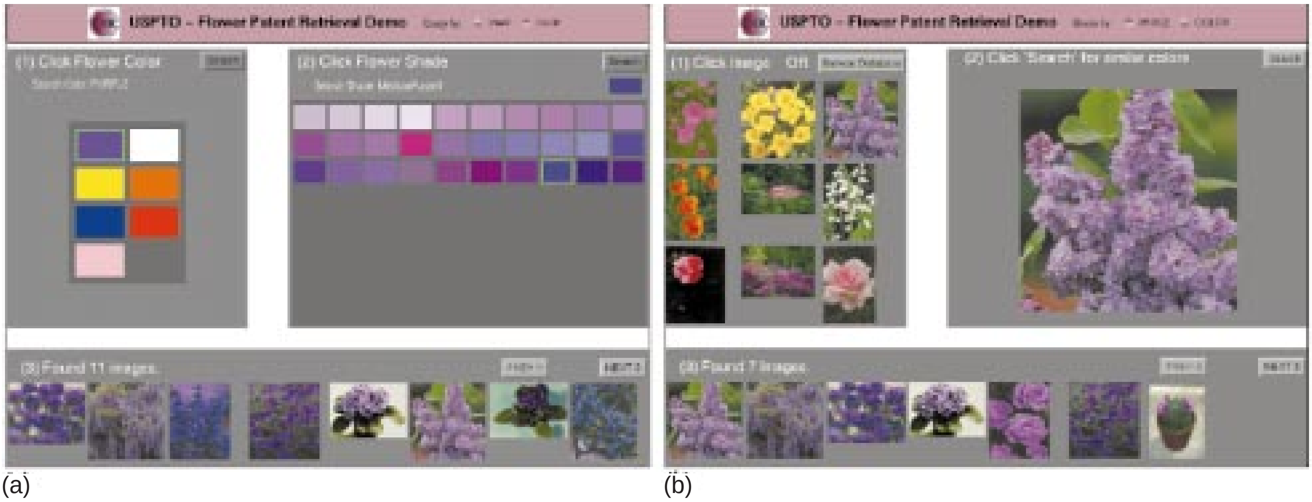


Figure 13. Retrieval by (a) color name: the color shade selected here is medium purple in the color class purple; (b) by example: the query selected is shown on the right.

on the left frame or by selecting one of the retrieved images. For the real application, we will add an interface that accepts users' images as queries. The right frame displays the example image selected, and the bottom shows the retrieved images. (This online interface to the retrieval system, which supports both types of queries, can be found at http://cowarie.cs.umass.edu/~demo/FlowerDemo.html.)

Figure 14 shows some sample retrieval results obtained using different types of queries. The first three rows demonstrate the query-by-example approach, where the first retrieved image was the query image. The last two rows show the results obtained from querying, using the color names orange and ivory. This figure shows only the top five images for each query.

Figure 14. First five retrieved images: the query for rows a–c is the first image retrieved in the row; the query for row d is the color orange; the query for row e is the color ivory.

WE HAVE FOCUSED ON THE importance of using domain knowledge to improve the retrieval performance for specialized applications in constrained image domains. The number of such applications is growing, and general-purpose image-retrieval strategies do not provide the level of performance required. Domain knowledge may improve the retrieval performance for applications in many specialized image databases. Our methodology uses color-based and spatial domain knowledge to automatically segment and index a database of flower images using an iterative segmentation algorithm. The indexing system uses a natural-language color-classification system to interpret color-based domain knowledge into rules to automatically segment the region of interest from the background. The approach suggested here can be adapted to any database dedicated to images of known subject about which some domain knowledge is available.

Further work on our current project will include tests on a large database of the order of 10,000 flower images. Our goal is to verify the retrieval results using feedback from actual users. We would also like to investigate the use of shape and texture features to broadly distinguish between flowers—for example, whether the flower is tubular or round, and whether it has one row of petals or multiple layers—to improve the precision of retrieval. We are also considering using color adjacency graphs to distinguish multicolored flowers containing the same colors.[16] We are also investigating how to use this approach for other specialized databases (for example, birds).

Acknowledgments

We thank Thomas Michel for building the online user interface for this retrieval system. We also thank Bruce Croft and the Center for Intelligent Information Retrieval for supporting this work.

This material is based on work supported in part by the National Science Foundation, the Library of Congress, and the Department of Commerce, under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235; in part by the National Science Foundation under grant number IRI-9619117; in part by the United States Air Force Office of Scientific Research under grant number F49620-99-1-0138; and in part by NSF Multimedia CDA-9502639. Any opinions, findings, and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

1. J.R. Bach et al., "The Virage Image Search Engine: An Open Framework for Image Management," *Proc. SPIE Conf. Storage and Retrieval for Still Image and Video Databases IV,* Vol. 2670, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1996, pp. 76–87.
2. W. Niblack et al., "The QBIC Project: Querying Images by Content Using Color, Texture and Shape," *Proc. SPIE Conf. Storage and Retrieval for Still Image and Video Databases*, Vol. 1908, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1993, pp. 173–187.
3. A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* IEEE Computer Soc. Press, Los Alamitos, Calif., 1994, pp. 84–90.
4. S. Ravela, R. Manmatha, and W.B. Croft, *Retrieval of Trademark and Gray-Scale Images Using Global Similarity*, Tech. Report MM-25, CIIR, Computer Science Dept., Univ. of Massachusetts, Amherst, Mass., 1999.
5. T.F. Syeda-Mahmood, "Data and Model-Driven Selection Using Color Regions," *Proc. Second European Conf. Computer Vision*, Springer-Verlag, New York, 1992, pp. 115–123.
6. Q. Chen, H. Wu, and M. Yachida, "Face Detection by Fuzzy Pattern Matching," *Proc. Fifth IEEE Int'l Conf. Computer Vision*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 591–596.
7. E. Saber et al., "Annotation of Natural Scenes Using Adaptive Color Segmentation," *Proc. SPIE Conf. Image and Video Processing III*, Vol. 2421, 1995, pp. 72–80.
8. Y. Gong and M. Sakauchi, "Detection of Regions Matching Specified Chromatic Features," *Computer Vision and Image Understanding*, Vol. 61, No. 2, Mar. 1995, pp. 263–269.
9. K.L. Kelly and D.B. Judd, "The ISCC-NBS Method of Designating Colors and A Dictionary of Color Names," *National Bureau of Standards Circular*, No. 553, Nov. 1955.
10. M.J. Swain and D.H. Ballard, "Color Indexing," *Int'l J. Computer Vision*, Vol. 7, No. 1, Jan. 1991, pp. 11–32.
11. A.R. Weeks and G.E. Hague, "Color Seg-

mentation in the HSI Color Space Using the K-Means Algorithm," *Proc. SPIE Conf. Nonlinear Image Processing VIII*, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., Vol. 3026, 1997, pp. 143–154.

12. W. Woelker, "Image Segmentation Based on an Adaptive 3D-Analysis of the CIE-L*a*b* Color Space," *Proc. SPIE Visual Communications and Image Processing*, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., Vol. 2727, No. 3, 1996, pp. 1197–1203.

13. Q. Huang et al., "Foreground/Background Segmentation of Color Images by Integration of Multiple Cues," *Proc. Int'l Conf. Image Processing*, IEEE Signal Processing Soc., Piscataway, N.J., 1995, pp. 246–249.

14. D. Hearn and M.P. Baker, *Computer Graphics*, Prentice-Hall, Upper Saddle River, N.J., 1994.

15. C.J. van Rijsbergen, *Information Retrieval*, Butterworths, Australia, 1979.

16. J. Matas, R. Marik, and J. Kittler, "On Representation and Matching of Multi-Coloured Objects," *Proc. Fifth IEEE Int'l Conf. Computer Vision*, IEEE CS Press, Los Alamitos, Calif., 1995, pp. 726–732.

Madirakshi Das is completing her doctoral thesis on the use of color in content-based image retrieval in the Computer Science Department at the University of Massachusetts, Amherst. Her research interests include multimedia indexing and retrieval, computer vision, and object recognition. She received her BS in electrical engineering and an MS in electronics and electrical communication engineering, both at the Indian Institute of Technology, Kharagpur, India. She also received an MS in computer science from the University of Massachusetts, Amherst. Contact her at mdas@cs.umass.edu.

R. Manmatha is a research assistant professor in the Computer Science Department at the University of Massachusetts, Amherst. He is the lead researcher in the Multimedia Indexing and Retrieval Group at the Center for Intelligent Information Retrieval. His current research is in the areas of image retrieval and document recognition and analysis. He has worked on image retrieval, finding text in images, and methods to retrieve handwritten manuscripts. He is involved in a project from the US Patent and Trademark Office to index and retrieve trademarks using both content-based retrieval and text. He earned his B.Tech. in electrical engineering from the Indian Institute of Technology, Kanpur, India, and his PhD in computer science from the University of Massachusetts, Amherst. Contact him at manmatha@cs.umass.edu.

Edward M. Riseman is a full professor in the Computer Science Department at the University of Massachusetts, Amherst. He has conducted research in computer vision, artificial intelligence, learning, pattern recognition, and information retrieval. He has codirected the University of Massachusetts Computer Vision Laboratory since its inception, with recent research projects in knowledge-based scene interpretation, motion analysis, indoor and outdoor mobile robot navigation, 3D terrain reconstruction and 3D site model construction for aerial photo interpretation, and aerial-image analysis for environmental monitoring. He is a senior member of the IEEE and a fellow of the AAAI. He received his BS in electrical engineering from Clarkson College of Technology, Potsdam, New York; he received his MS and PhD, both in electrical engineering, from Cornell University. Contact him at riseman@cs.umass.edu.

Literature survey and related work

Image retrieval has been an active area of research since the early 1990s. As more application areas are encountered,[1,2] finding an efficient solution to this problem becomes increasingly important. Because the end user of image-retrieval systems is usually a human being, the retrieval results should provide the images that a human would have selected if she could manually browse through the full database. This is an ill-defined problem, because a human's idea of image semantics is hard to encode in an automatic algorithm. The best a system can do is to appear to be intelligent by using some of the attributes a person would use to categorize images. Human beings tend to describe images based on the objects represented in them, so an image description in terms of objects found in the image is more likely to produce results matching the human perception of the image content. However, object recognition in a general image domain is very difficult, and no general solutions exist.

To avoid the object-recognition problem, researchers have found several low-level features that are well correlated with image content. An image is described in terms of these low-level features or attributes. It is assumed that images with matching low-level features will have related semantic content. The quality of retrieval obtained will depend on the extent to which the attributes used relate to image content. For example, machine parts can be distinguished on the basis of their shape; commercial products can be identified by their color; and texture could be used to distinguish animals with different types of fur. These examples also illustrate the point that the attributes that work are domain-specific; an attribute that works well in one domain may not be relevant in another domain.

Using shape and texture

It would appear that a 2D shape would be an important feature for distinguishing some objects from others. Researchers have done considerable work in the area of pattern recognition to match such shapes to each other. For example, Babu Mehtre and others compare various shape measures for content-based retrieval on a database of trademark images.[3] The features used to describe shape can be classified into those that describe the objects' boundaries (such as string encoding and Fourier descriptor coefficients) and those that describe the regions in the image (such as polygonal approximations[4] and invariant moments).[5]

However, much of this work assumes the object can be segmented from the background before the shape features are computed. This may not be a problem for databases that depict the object against a plain background, but this is a serious problem for general image databases. In general, an object's appearance in an image depends not only on its 3D shape but also on the relative viewpoint of the object and the camera, its albedo, and how it is illuminated. It is difficult to separate the object's shape from these other factors. Thus, image segmentation (especially when the segments need to correspond to objects in the image) is a hard problem for which no general solution exists. Some systems have used manual segmentation to overcome this problem.[6]

For some objects (such as animal skin, fur, and vegetation), texture is an important distinguishing feature because they show distinctive texture patterns. Wei-Ying Ma and B.S. Manjunath have used texture-based patterns for image retrieval.[7] Fang Liu and Rosalind Picard have proposed an image model based on the Wold decomposition of homogeneous random fields into three mutually orthogonal subfields that correspond to the most important dimensions of human texture perception: periodicity, directionality, and randomness.[8] These texture features have been shown to be effective in retrieving perceptually similar natural textures. Another image description that has been used for gray-scale images includes appearance (proposed by Srinivas Ravela and R. Manmatha),[9,10] which describes the intensity surface of the images and Eigen features.[11]

Using color features

Color is a very commonly used low-level feature (when the database images are in color). It is useful for indexing objects having very specific colors—for example, commercial products, flags, postal stamps, birds, fishes, and flowers—or as a first pass for other colored images. Michael Swain and Dana Ballard proposed using color histograms to index color images and described an efficient histogram-intersection technique for matching.[12] Normalized color histograms along with histogram intersection have been popular for indexing color images because of the fast speed of matching and because they are generally invariant to translation, rotation, and scale. However, because color histograms do not incorporate information on the spatial configuration of the color pixels, there are usually many false matches where the image contains similar colors in different configurations.

A few researchers have attempted to include this information in the representation to improve the retrieval results. Ramin Zabih and others have proposed the color correlogram, which includes information on the spatial correlation of pairs of colors in addition to the color distribution in the image.[13] Jiri Matas and others have described a color-adjacency graph that can be used to describe multicolored objects, but the matching phase is too computationally intensive for use in large image databases.[14] Madirakshi Das and others have proposed a simpler spatial-adjacency graph structure, which is used in a filtering phase to enforce the spatial properties of the colors the query image requires.[15] The main problem with color-based image retrieval is that color as a feature is not well correlated with image content in a general image database. For example, a query with a red ball may retrieve red cars, flowers, a person wearing a red shirt, or a fire truck. In addition, using color alone is not sufficient to produce enough discrimination between database images with only a few colors—for example, images of apes, tigers, and forests. However, in domains where color is an important attribute, it can be very useful.

Several studies have shown that using a combination of features produces better retrieval results than using each feature alone.[16,17] Researchers have used different combinations of features, depending on their appropriateness for the test database. Anil Jain and Aditya Vailaya have used color histograms and shape as features to index a database of trademark images.[18] They also describe shape as a histogram by taking counts of the different edge directions present in the image. Serge Belongie and others use color and texture features to segment an image into regions of coherent color and texture and represent the image in terms of these regions for content-based retrieval.[19]

General-purpose retrieval systems

For retrieval systems that work with general databases such as generic stock photographs and mixed news photographs, we do not know a priori which feature (or combination of features) would produce better retrieval performance. This depends on the type of object or scene depicted in the query. Many such systems implement a wide variety of features and let the user choose the important aspects of the query at query time. An example of a system that implements color, texture, and shape is Query by Image Content (QBIC), which allows queries based on example images, sketches, or selected color and texture patterns.[6] The user can select the features to be used as well as the relative importance to be attached to each feature in the final ranking. Virage is another general-purpose retrieval system that provides an open framework for general features such as color, shape, and texture, as well as very domain-specific features for plug-ins.[20] The Photobook retrieval system uses shape, texture, and Eigen images as features, in addition to textual annotations.[21] The system can be trained to work on specific classes of images.

Other examples of existing systems using multiple features and multiple query modes are the Candid[22] and Chabot systems.[23] An emerging problem in general image search is retrieving relevant images from the World Wide Web. John Smith and Shih-Fu Chang have implemented an image retrieval system for the Web named VisualSeek, which uses spatially localized color regions in the images to describe them.[24] Stan Sclaroff and others have developed the ImageRover system to gather images from the Web and index them using color, texture, orientation, and other specialized features.[25] Traditional keyword-based search engines such as Yahoo and Lycos have also implemented image search engines, but these are actually text-based search engines, which extract keywords from the image captions and the URL where the image is embedded.

On the basis of the above discussion, we see that the trend in general image-retrieval systems has been to provide many low-level features as well as specialized features. However, the users are the ones expected to select the feature or combination of features relevant to their queries. Appropriate feature selection is difficult, requiring knowledge of the features and experience in using them, neither of which should be expected of users.

An even more significant problem that arises from the use of multiple features is how the features should be combined. Surfimage by Chahab Nastar and others uses normalized linear combination and voting methods to rank images on the basis of a combination of features.[16] In other systems, the user needs to prioritize each selected feature, by its importance, which may be very hard to do.

One of the weaknesses of image retrieval techniques has been in their evaluation. Most researchers have evaluated their techniques on their own individual databases. It is not always clear, especially for techniques focused on general image collections, what the evaluation criteria are. For example, in such databases similarity is sometimes hard to define. For some applications of face retrieval, the Feret database provides a standard test collection.[26]

Although many systems have tried to solve the image retrieval problem in a general database, the question of what the user really needs has rarely been answered. The most common query format is to provide an example image, but this may not be sufficient to fathom the user's intent. For example, the user may provide a picture with a car parked in front of a building on a sunny day, which could mean he wants other pictures of the same building, pictures of similar cars, pictures of buildings with cars parked in front, or even other sunlit scenes! One approach to specifying the object of interest has been to allow subimages as queries where the user marks the area of interest.[10,15] However, this might not be sufficient for clarifying the user's query, and providing subimage matching is usually more difficult. This has led to the use of relevance feedback, a well-known technique used several years ago for text-based information retrieval. In this approach, the user marks the relevant and irrelevant images out of the retrieved images. The retrieval system recomputes the match scores based on this user feedback and provides a more relevant set of images. The more recent systems—for example, Surfimage[16]—provide relevance feedback as a mechanism for refining the retrieval results interactively, using input from the user.

Image retrieval from specialized databases

In image retrieval applications involving specialized domains, the user's needs are often well-defined. However, general-purpose retrieval systems might not do as well as the user expects on specialized, constrained domains, because they do not exploit any of the domain's special features. Several specialized domains need automatic retrieval solutions; these domains are currently indexed by manual annotations and specialized codes involving extensive, tedious human involvement. Many of these specialized domains require formulating features specific to the domain to produce good retrieval results. For example, Alex Pentland and others describe the Eigen image representation that measures the similarity in appearance of faces and then searches for similar faces in the Photobook system.[27] There are many applications of systems for retrieving faces,[27] including identity verification for financial transactions and law enforcement. To a limited extent, specialized approaches have also been used for images of specific objects in general collections of images. For example, several systems can find human faces in a general collection of images,[28] and some researchers have tried to find horses in such collections.[29]

Even when the domain has various images (for example, trademarks), the application may be specialized. For example, for trademark retrieval, Srinivas Ravela and R. Manmatha have used a global similarity measure for images based on curvature and phase to produce results on a database of trademark images; they were able to produce results superior to those obtained from general-purpose shape-based approaches.[9,30]

John Eakins and others have developed a trademark retrieval system named Artisan, which uses Gestalt theory to group low-level elements such as lines and curves into perceptual units that describe the trademark.[31] Besides developing appropriate features for specialized databases, one might be able to segment and describe the objects depicted in the image using knowledge about the objects to simplify the segmentation process. David Forsyth and Margaret Fleck describe a representation for animals as an assembly of almost cylindrical parts.[29] On a database of images of animals, their representation can retrieve images of horses, for example, in a variety of poses. Fleck and others use knowledge about the positions of attachment of the limbs and head to the human body to detect the presence of naked people in the database images.[32] Forsyth and others illustrate some specialized applications of image retrieval.[1]

References

1. D. Forsyth, J. Malik, and R. Wilensky, "Searching for Digital Pictures," *Scientific American*, Vol. 276, No. 6, June 1997, pp. 72–77.

2. V.N. Gudivada and V.V. Raghavan, "Content-Based Image Retrieval Systems," *Computer*, Vol. 28, No. 9, Sept. 1995, pp. 61–70.

3. B. Mehtre, M. Kankanhalli, and W. Lee, "Shape Measures for Content-Based Image Retrieval: A Comparison," *Information Processing and Management*, Vol. 33, No. 3, Mar. 1997, pp. 319–337.

4. R. Mehrotra and J.E. Gary, "Similar-Shape Retrieval in Shape Data Management," *Computer*, Vol. 28, No. 9, Sept. 1995, pp. 57–62.

5. C.C. Chen, "Improved Moment Invariants for Shape Discrimination," *Pattern Recognition*, Vol. 26, No. 5, May 1993, pp. 683–686.

6. W. Niblack et al., "The QBIC Project: Querying Images by Content Using Color, Texture and Shape," *Proc. SPIE Conf. Storage and Retrieval for Still Image and Video Databases*, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1993, pp. 173–187.

7. W.Y. Ma and B.S. Manjunath, "Texture-Based Pattern Retrieval from Image Databases," *Multimedia Tools and Applications*, Vol. 2, No. 1, Jan. 1996, pp. 35–51.

8. F. Liu and R.W. Picard, "Periodicity, Directionality and Randomness: Wold Features for Image Modeling and Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, July 1997, pp. 722–733.

9. S. Ravela and R. Manmatha, "On Computing Global Similarity in Images," *Proc. Workshop Applications of Computer Vision*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 82–87.

10. S. Ravela and R. Manmatha, "Image Retrieval by Appearance," *Proc. SIGIR '97*, ACM Press, New York, 1997, pp. 278–285.

11. D.L. Swets and J. Weng, "Using Discriminant Eigen Features for Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, Aug. 1996, pp. 831–836.

12. M.J. Swain and D.H. Ballard, "Color Indexing," *Int'l J. Computer Vision*, Vol. 7, No. 1, Jan. 1991, pp. 11–32.

13. J. Huang et al., "Image Indexing Using Color Correlograms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp.762–768.

14. J. Matas, R. Marik, and J. Kittler, "On Representation and Matching of Multi-Coloured Objects," *Proc. Fifth IEEE Int'l Conf. Computer Vision*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 726–732.

15. M. Das, E.M. Riseman, and B.A. Draper, "FOCUS: Searching for Multi-Colored Objects in a Diverse Image Database," *Proc. Sixth IEEE Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp. 756–761.

16. C. Nastar et al., "Surfimage: A Flexible Content-Based Image Retrieval System," *Proc. ACM Multimedia '98*, ACM Press, New York, 1998, pp. 339–344.

17. R. Picard, T. Minka, and M. Szummer, "Modeling Subjectivity in Image Libraries," *Proc. Int'l Conf. Image Processing*, IEEE Signal Processing Soc., Piscataway, N.J., 1996, pp. 18–24.

18. A.K. Jain and A. Vailaya, "Image Retrieval Using Color and Shape," *Pattern Recognition*, Vol. 29, No. 8, Aug. 1996, pp. 1233–1244.

19. S. Belongie et al., "Color- and Texture-Based Segmentation Using EM and Its Application to Content-Based Image Retrieval," *Proc. Sixth IEEE Int'l Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 675–682.

20. J.R. Bach et al., "The Virage Image Search Engine: An Open Framework for Image Management," *Proc. SPIE Conf. Storage and Retrieval for Still Images and Video Databases IV*, Vol. 2670, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1996, pp. 76–87.

21. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases," *Int'l J. Computer Vision*, Vol. 18, No. 3, Mar. 1996, pp. 233–254.

22. P. Kelly, M. Cannon, and D. Hush, "Query by Image Example: The CANDID Approach," *Proc. SPIE Conf. Storage and Retrieval for Still Image and Video Databases*, Vol. 2420, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1995, pp. 238–248.

23. V. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images," *Computer*, Vol. 28, No. 9, Sept. 1995, pp. 40–48.

24. J.R. Smith and S.F. Chang, "Visually Searching the Web for Content," *IEEE Multimedia*, Vol. 4, No. 3, Sept. 1997, pp. 12–20.

25. S. Sclaroff, L. Taycher, and M. La Cascia, "ImageRover: A Content-Based Image Browser for the World Wide Web," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp. 2–9.

26. P.J. Phillips et al., "The FERET Evaluation Methodology for Face Recognition Algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp. 137–143.

27. A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., June 1994, pp. 84–90.

28. H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, Jan. 1998, pp. 23–38.

29. D.A. Forsyth and M.M. Fleck, "Body Plans," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp. 678–683.

30. S. Ravela, R. Manmatha, and W.B. Croft, *Retrieval of Trademark and Gray-Scale Images Using Global Similarity*, Tech. Report MM-25, Center for Intelligent Information Retrieval, Computer Science Dept., Univ. of Massachusetts, Amherst, Mass., 1998.

31. J.P. Eakins, K. Shields, and J. Boardman, "ARTISAN: A Shape Retrieval System Based on Boundary Family Indexing," *Proc. SPIE: Storage and Retrieval for Still Image and Video Databases*, Vol. 2670, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1996, pp. 17–28.

32. M.M. Fleck, D.A. Forsyth, and C. Bregler, "Finding Naked People," *Proc. Fourth European Conf. Computer Vision*, Springer-Verlag, New York, 1996, pp. 72–77.