Indexing Flower Patent Images using Domain Knowledge *

Madirakshi Das   R. Manmatha   Edward M. Riseman

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
E-mail : (mdas,manmatha,riseman)@cs.umass.edu

Abstract

A new approach to indexing a specialized database by utilizing the color and spatial domain knowledge available for the database is described. This approach is illustrated by using it to provide a solution to the problem of indexing images of flowers for searching a flower patents database by color. The flower region is isolated from the background by using an automatic iterative segmentation algorithm with domain knowledge-driven feedback. The color of the flower is defined by the color names present in the flower region and their relative proportions. The database can be queried by example and by color names. The system provides a perceptually correct retrieval with natural language queries by using a natural language color classification derived from the ISCC-NBS color system and the X Window color names. The effectiveness of the strategy on a test database is demonstrated.

1   Introduction

The advent of the information revolution has lead to an enormous increase in the amount of information that people and organizations have to deal with. To be able to use this information effectively, people require tools to manage the information; including tools for searching, retrieving and classifying it. A number of good search engines exist for text in ASCII form. However, there are no good tools of comparable performance for retrieving images available yet.

Traditionally image databases have been manually annotated using textual keywords. The images are then retrieved based on the manually assigned keywords. Manual annotation is slow, expensive and impractical for the large image databases that are being created today. In addition, manual annotations suffer from many limitations; annotations may be inaccurate (especially for large databases) and they cannot encode all the information present in an image. Thus there has been a great deal of interest, recently, in content-based retrieval of images where the goal is to find images in the database which are "similar" in part or whole to a query or example image.

This paper discusses how a database of flower patent images may be queried using both an example flower image as well as by using the names of colors. Flower images are submitted as a part of the process of applying for flower patents from the U.S. Patent and Trademark Office. A person who would like to check whether a new flower submitted for patenting is unique, can provide an example image from the patent application to retrieve similar flowers that already exist in the database. On the other hand, a person looking for flowers to cultivate may only be able to specify the flower type and a color name when querying the database.

The specific research contributions of the paper include methods to take advantage of the domain (flower patents) to isolate the flower region from the background. The color of the flower is then extracted. Unlike many other color based retrieval systems [1, 26], this ensures that only the color of the flower is used in the indexing process rather than colors in the entire image. A natural language color classification derived from the ISCC-NBS color system and the X Window color names is linked to the color of the flower. The database may be queried either by using natural language queries describing the color of a flower or by providing an example image of the flower.

This work is motivated by the need for formulating a methodology for using the domain knowledge available for specialized databases to provide better retrieval perfor-

mance than general-purpose retrieval strategies. We believe that this approach may be applied to other domains and databases. For example, databases of bird images or images of mammals are good candidates for such an approach.

2 Background and Motivation

The basic step towards meaningful retrieval is to ensure that the image descriptions used to index the database are related to the semantic content of the image. This requirement is difficult to meet in the context of 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 has no semantic meaning by itself. 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, many image attributes like color, texture, shape and "appearance" may often be directly correlated with the semantics of the problem. For example, logos or product packages (e.g., a box of Tide) have the same color wherever they are found. The coat of a leopard has a unique texture while Abraham Lincoln's appearance is uniquely defined. These image attributes can often be used to index and retrieve images.

These attributes must be used with care if they are to correlate with the semantics of the problem. For example, many image retrieval systems (see [1, 26]), use color to retrieve images from general collections. A picture of a red bird used as a query, may 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 was limited to those containing birds, the results retrieved would be restricted to birds and probably be much more meaningful from the viewpoint of a user.

While many image retrieval algorithms have been focused on retrieving images from general image collections, we believe that the approach of restricting image retrieval to specialized collections of images or to specific tasks will be more successful and useful. 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 query face image from a database of face images (see [28] for an early example of such a system). There are many applications of systems for retrieving faces 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, a number of systems have been devised to find human faces in a general collection of images (for example [35]) and an attempt has also been made to find horses [9] in such collections.

The nature of the task often modifies the approach taken to image retrieval. Thus, for example in the application discussed in this paper, a flower of a different color is not considered to be a match. However, in trademark retrieval, color plays no role. That is, a trademark is considered identical to another trademark even if their colors are different. Trademarks are a good example of a task in which all types of images occur but the task is very specific (i.e. 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 [34]. The use of text retrieval allows additional constraints to be used. For example, two trademark images which are visually identical are considered conflicts only if they are used for similar goods and services.

This work is motivated by the need for a better approach for indexing a specialized database by exploiting the knowledge available for the domain covered by the database. As an example, we will investigate the utility of domain knowledge in indexing a database of images which have been digitized from photographs submitted as a part of applications for flower patents to the U.S. Patents and Trademark Office. This database needs to be queried both by example images and by color name so that both persons in charge of checking new patent applications and persons buying patents for cultivation can use it.

Though 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 shown in Fig 1. There are two main problems to be addressed in this application : the problem of segmenting the flower from the background and the problem of describing the color of the flower in a form which matches human perception and allows flexible querying by example and by natural language color names.
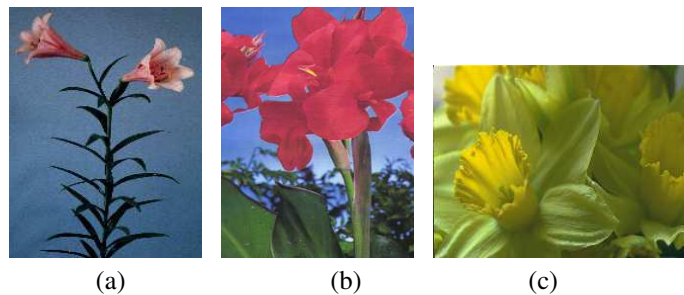

(a)   (b)   (c)

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

We would like to use the characteristics of this domain to automate the segmentation and indexing process. Most

of the domain knowledge is in the form of natural language statements; translating these into rules which can be used to build automated algorithms is non-trivial. For example, like most natural subjects, a lot of color-based domain knowledge is known for the flower domain e.g. flowers are rarely green, black, gray or brown in color. Examples of information in other domains would be facts like mammals are rarely blue, violet or green and outdoor scenes often have blue and white skies and green vegetation. However, these types of information can only be used effectively when there is a mapping from the 3D color space to natural language color names. We have constructed a mapping to a natural language color name space using color names from the ISCC-NBS [16] system and the color names defined in the $XWindow^{TM}$ system for this purpose.

We have developed an iterative segmentation algorithm which uses the available domain knowledge to provide a hypothesis marking some color(s) as background color(s) and then testing the hypothesis by eliminating those color(s). The evaluation of the remaining image provides feedback about the correctness of the hypothesis and a new hypothesis is generated when neccessary after restoring the image to its earlier state.

The next section of the paper surveys related work. Section 4 addresses the problem of segmenting the flower from the background using domain knowledge. Section 5 discusses indexing and retrieval from the database including different types of queries supported. Section 6 describes experiments carried out to test the system and the conclusions are summarized in Section 7.

3 Literature Survey and Related Work

Image retrieval has been an active area of research since the early '90s. As more application areas are encountered [8, 11], it is increasingly important to find an efficient solution to this problem. Since the end user of image retrieval systems is usually a human being, the retrieval results should aim to provide the images that a human would have selected if (s)he 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 human would use to categorize images. Human beings tend to describe images based on the objects represented in it, 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 a very hard problem and no general solutions exist. To avoid the object recognition problem, researchers have found a number of 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 attribute(s) used are related 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 at all in another domain. We will take a closer look at the attributes that have been used in image retrieval and their relevance to solving the general image retrieval problem, and to solving particular problems in different image domains.

It would appear that two-dimensional shape would be an important feature for distinguishing some objects from others. Considerable work has been done in the area of pattern recognition, on matching such shapes to each other. For example, Mehtre et al [24] provide a comparative study of various shape measures for content-based retrieval on a database of trademark images. The features used to describe shape can be classified into those that describe the boundary of the objects, like string encoding and Fourier descriptor co-efficients, and those which describe the regions in the image like polygonal approximations [23] and invariant moments [3]. However, much of this work assumes that the object can be segmented from the background before the shape features can be computed. This may not be a problem for databases where the object is depicted 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 three dimensional shape but also on the relative viewpoint of the object and the camera, its albedo as well as on how it is illuminated. It is difficult to separate out the shape of the object 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 [26] to overcome this problem.

For some objects, texture is an important distinguishing feature because these subjects (like animal skin, fur, vegetation etc.) show distinctive texture patterns. Ma and Manjunath [21] have used texture-based patterns for image retrieval. Liu and Picard [19] have proposed an image model based on the Wold decomposition of homogeneous random fields into three mutually orthogonal subfields which correspond to the most important dimensions of human texture perception - periodicity, directionality and randomness. These texture features have been shown to be effective in retrieving perceptually similar natural textures. Other image descriptions that have been used for grey-scale

images include *appearance* (proposed by Ravela and Manmatha [33, 32]) which describes the intensity surface of the images and eigen features [41].

Color is a very commonly used low-level feature when the database images are in color. It is useful for indexing objects which have very specific colors, for example, commercial products, flags, postal stamps, birds, fishes and flowers, or as a first pass for other colored images. Swain and Ballard [40] proposed the use of color histograms to index color images and described an efficient histogram intersection technique for matching. Normalized color histograms along with histogram intersection have been popular for indexing color images because of the fast speed of matching and the fact that they are generally invariant to translation, rotation and scale. However, since 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. Zabih et al [13] have proposed the color correlogram which includes information on the spatial correlation of pairs of colors in addition to the the color distribution in the image. Matas et al [22] have described a color adjacency graph which can be used to describe multi-colored objects, but the matching phase is too computationally intensive for use in large image databases. Das et al [5] have proposed a simpler spatial adjacency graph structure which is used in a filtering phase to enforce the spatial properties of the colors required by the query image. 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.

A number of studies have shown that the use of a combination of features produces better retrieval results than using each of the features alone [25, 31]. Different combinations of features have been used depending on their appropriateness for the test database. Jain and Vailaya [15] have used color histograms and shape as features to index a database of trademark images. The shape is also described as a histogram by taking counts of the different edge directions present in the image. Belongie et al [2] use color and texture features to segment an image into regions of coherent color and texture and represent the image in terms of these "blobs" for content-based retrieval.

For retrieval systems that work with general databases like generic stock photographs and mixed news pho-

tographs, it is not clear 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 which implements color, texture and shape is QBIC [26] which allows queries based on example images, sketches or selected color and texture patterns. 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 [1] is another general purpose retrieval system which provides an open framework to allow general features like color, shape and texture as well as very domain specific features to be used as plug-ins. The Photobook [29] retrieval system uses shape, texture and eigenimages as features in addition to textual annotations. 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 Candid [17] and Chabot [27]. An emerging problem in general image search is to retrieve relevant images from the World Wide Web. Smith and Chang [39] have implemented an image retrieval system for the World Wide Web (named VisualSEEk) using spatially localized color regions in the images to describe the images. Sclaroff et al [37] have developed the ImageRover system to gather images from the web and index them using color, texture, orientation and other specialized features. Traditional keyword-based search engines like 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 in which the image is embedded.

Based on the above discussion, it is clear that the trend in general image retrieval systems has been to provide a large number of low-level features as well as specialized features. However, it is the user who is expected to select the feature or combination of features that are relevant to his/her query. Appropriate feature selection is a hard problem, requiring knowledge of the features and experience in using them, neither of which should be expected of the user. An even more significant problem that arises from the use of multiple features is how the features should be combined. Surfimage by Nastar et al [25] uses normalized linear combination and voting methods to compute the ranks of images based on a combination of features. In other systems, the user needs to weight each feature selected, 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 [30] provides a standard test collection.

While a number of different systems have been implemented which try to solve the image retrieval problem in a general database, the question of what the user really needs has often been left unanswered. 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 any one of : (s)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 sub-images as queries where the user marks the area of interest [5, 32]. However, this may not be sufficient for clarifying the user's query and providing sub-image matching is usually more difficult. This has lead to the use of relevance feedback, a well-known technique used earlier for text-based information retrieval. In this approach, the user marks the relevant and irrelevant images out of the retrieved images. The system recomputes the match scores based on this user feedback, and provides a more relevant set of images. The more recent systems like Surfimage [25] provide relevance feedback as a mechanism for refining the retrieval results interactively using input from the user.

In image retrieval applications involving specialized domains, the user's needs are often well-defined. However, general purpose retrieval systems may not do as well as expected by the user on specialized, constrained domains because they do not exploit any of the special features of the domain. There is a need for automatic retrieval solutions in a number of specialized domains which are currently indexed by manual annotations and specialized codes which involve extensive, tedious human involvement. In many of these specialized domains, features specific to the domain need to be formulated to produce good retrieval results. For example, Pentland et al [28] describe the eigen-image representation which measures the similarity in appearance of faces which is used to search for similar faces in the Photobook system. Even when the domain has a wide variety of images (for example trademarks), the application may be specialized. For example, for trademark retrieval, Ravela and Manmatha [33, 34] have used a global similarity measure for images based on curvature and phase to produce superior results on a database of trademark images when compared to general-purpose shape-based approaches. Eakins et al [6] have developed a trademark retrieval system (named ARTISAN) which uses Gestalt theory to group low-level elements like lines and curves into perceptual units which describe the trademark. In addition to developing appropriate features for specialized databases, one may be able to segment and describe the objects depicted in the image using knowledge about the objects to simplify the segmentation process. Forsyth and Fleck [9] describe a representation for animals as an assembly of almost cylindrical parts. On a database of images of animals, their representation can retrieve images of horses, for example, in a variety of poses. Fleck et al [7] use knowledge about the positions of attachment of limbs and head to the human body to detect the presence of naked people in the database images. Forsyth et al illustrate some specialized applications of image retrieval in [8].

There is work in the areas of color image segmentation and modeling the appearance of colored objects which is also relevant to this work. Color histograms [40] in different color spaces [44, 45] have been used in different forms in a lot of work in the area of color image segmentation [18, 38]. Multiresolution color image segmentation is described in [20]. However, all the systems above do not identify the object of interest and cannot distinguish the background elements from the foreground elements. Automatic foreground/background disambiguation based on multiple features like color, intensity and edge information has been studied in [14], but these techniques assume relatively smooth backgrounds and objects with sufficient contrast.

Since we would like to use color domain knowledge, the color space needs to be mapped to colors as perceived by humans. There has been work on perceptual organization of the color space in the area of image indexing [42] and in color science [46] without mapping the perceptual groups obtained to natural language color names. These approaches are not very useful in the translation of natural language rules about color into computer usable information. However, they provide good indexing tools when the object of interest has been pre-segmented from the background. In the reverse approach, color domain knowledge has been mapped to the 3D color space in applications like face identification using skin tones [4] and automatic target recognition, where the part of the color space which corresponds to the object of interest is identified. Modeling the distribution of color points in objects is an important issue in this approach. The set of pixels in each natural object is modeled as a Gaussian probability density function in annotating natural scenes in [36]. Regions corresponding to a specified color model are detected in [10].

4 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 knowl-

edge to automatically extract a region from the image which has a high probability of being a flower region. The types of information available for this application can be categorized into *color-based* and *spatial* domain knowledge. Since color-based domain knowledge is available in terms of natural language color descriptions and providing color name-based retrieval is also one of our goals, the color space needs to be mapped to commonly used color names. The next task is to segment the image using both color and spatial domain knowledge. These steps which constitute the offline processing phase of indexing the database, are described in this section.

4.1 Mapping from color space to names

We need tables mapping points on a 3-D color space to color names which should agree with human perception of colors to be useful. We use two sources for names (i) the ISCC-NBS color system which produces a dense map from the Munsell color space to names and the (ii) colors defined by the X Window system which provides a sparse mapping from the RGB space to 359 names. The ISCC-NBS system uses a standard set of base hues [Fig 2] and generates 267 color names using hue modifiers [Fig 3]. This gives us a color system which can be easily decomposed into a hierarchy of colors where we may use the full color name, partial names, base hues or coarser classes [Fig 4] comprising groups of base hues.

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

Figure 2. Hue names in the ISCC-NBS system

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

Figure 3. Hue modifiers in the ISCC-NBS system

The color names in ISCC-NBS system often have simpler commonly used alternatives, for example, 'very pale

red	green	brown	orange
blue	purple	pink	yellow
violet	black	white	gray

Figure 4. Color classes derived by grouping ISCC-NBS hue names and adding three neutral colors

yellowish white' in the ISCC-NBS system is the color 'ivory' and 'light brownish yellow' is the color 'khaki'. The simpler names, like 'ivory' and 'khaki', which are often derived from commonly known objects of the same color, are obtained from the definitions in X Window system.

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 can be perceived by a human. The distances between points in this space are also not representative of the perceived distances between colors. We have used the HSV color space [12] discretized into 64x10x16 bins as an intermediate space to reduce the number of colors as well as have perceptually similar colors in the same neighborhood.

RGB (256x256x256)	(245,195,40)	(233,150,122)
HSV (64x10x16)	(7,8,15)	(2,5,14)
XColor names (359)	goldenrod2	dark salmon
ISCC-NBS colornames (267)	strong yellow	dark brownish pink
Color classes (12)	yellow	pink

Figure 5. Example of color representations used

Each point on the discretized HSV space is mapped to a color defined in X Window system. Points with no exact map are mapped to the nearest color name using the city block measure to compute distances. Each point is also mapped to the ISCC-NBS name [Fig 5]. The ISCC-NBS name is used to produce a color hierarchy so that queries can be general (for example, blue) or specific (for example, pale blue). This color structure is also used in segmentation of the flower from its background. Using color names from two sources improves the chances of finding a name which matches the user's natural language query.

4.2 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 colors of the flower. The flower regions are isolated from the background using domain knowledge about the color of flowers and also knowledge about the distribution of background regions in photographs.

4.2.1 Use of domain knowledge

Since we have constructed a mapping from 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 occuring elements of the background in flower images by deleting pixels which belong to color classes which do not represent colors of flowers. Black and gray are mostly contributed by the shadow regions in the image, brown pixels come from shadows as well as branches and soil while green pixels are from the 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 which is helpful in identifying background regions is that background colors are usually visible along the periphery of the image. If this observation was always true, the background color could be detected with certainty by analysing the colors present in the margins of the image. However, the margins of the image could be of three different types as shown in Fig 1. 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. This means that the flower itself will occupy a reasonable part of the image. Also, since the flower is the object of interest, it is unlikely that it will be present *only* near the boundaries of the image. It could, however, be present throughout the image, *including* the boundary region. The background may have other colored objects but they will not usually dominate the main subject, which is 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, though there may be many of them in the image. Due to this, a single prominent segment identified as a flower region can be selected out of multiple segments without loss of information. The goal is to isolate a region in the image from which a good description of the color of the flower can be obtained and *not* the detection of all flower regions in the image.

4.2.2 Segmentation strategy

Our approach to extracting a region which has a high probability of being a part of a flower is to use the knowledge discussed above in successively eliminating background colors till the remaining region consists solely of flower areas. This entails the generation of a hypothesis indentifying the background color(s). However, since the hypothesis may be wrong, we use a feedback mechanism from the segmentation results obtained 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 is run after *binarizing* the image, where the only two classes are pixels which have been eliminated and those that remain.
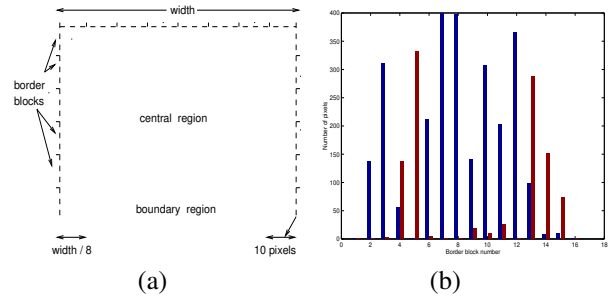

(a)   (b)

Figure 6. (a) Definitions of image regions (b) Color distribution in border blocks of canna image in Figure 1 (b)

The outline of the algorithm used to produce a segment from which the flower color is estimated is shown in Fig 7.

The image pixels are labelled by their color classes as well as their nearest X Window system color name. We use a coarse-to-fine strategy when using the color labels - the color class description is used first, finer color name distinctions are used only when necessary. In the first step, pixels belonging to the color classes *black*, *gray*, *brown* and *green* are eliminated since these are non-flower colors and the remaining image is segmented after binarization.

We use two criteria for evaluating whether a segment produced is valid; it should be of a minimum size which is based on the size of the largest segment obtained after deleting the non-flower color classes, and its centroid should fall within the 'central region' of the image as defined in Fig 6 (a). These requirements are based on the domain knowledge discussed in the previous sub-section. 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 Fig 8. Since we know that the flower is the dominant subject of the image, the largest segment has the highest probability of being a flower region.

Only the pixels comprising the largest valid segment are retained and the rest of the pixels are eliminated. In flower images taken in natural surroundings from a distance, this process is sufficient to produce a good flower segment. An example is shown in Fig 8 where the final result of segmentation is the image (c).

Further processing is required when the largest segment contains background colors in addition to the flower re-
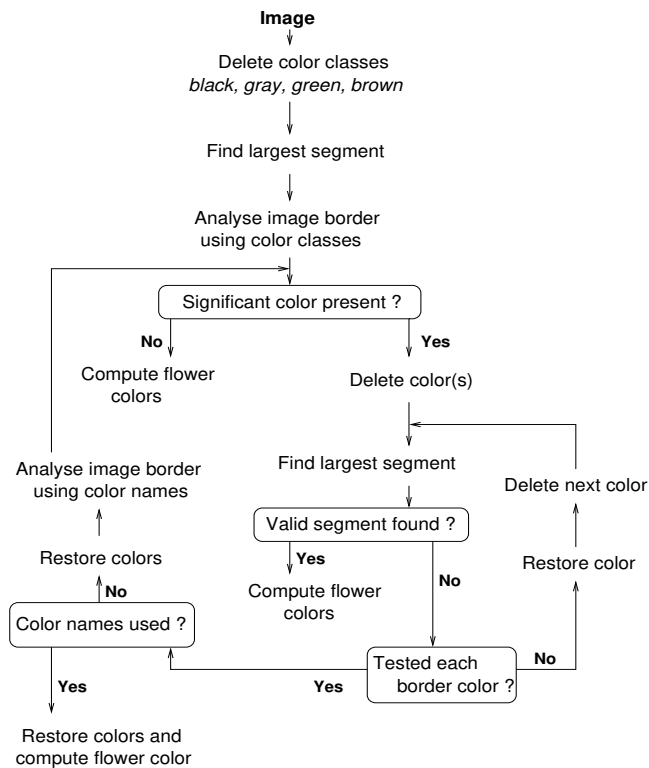
Image
↓
Delete color classes
black, gray, green, brown
↓
Find largest segment
↓
Analyse image border
using color classes
↓
Significant color present ?

No → Compute flower colors

Yes → Delete color(s)
↓
Find largest segment
↓
Valid segment found ?

Yes → Compute flower colors

No → Tested each border color ?

Delete next color

Restore color

Analyse image border
using color names
↑
Restore colors
↑ **No**
Color names used ?

Yes → Restore colors and compute flower color

Figure 7. System Overview

gions. First, the image is reduced by retaining the pixels covered by the largest segment only. The presence of background colors is then detected by analysing the color composition along the image margins. The margins of the image are divided into border blocks as shown in Fig 6(a). The distribution of color classes in these blocks is computed and colors showing substantial presence in more than half of the blocks are marked as possible background colors. For example, Fig 6(b) shows the color distributions for the two color classes present in the border of the image in Fig 1(b). From this distribution, the color *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 colors which were hypothesized to be background colors, the largest seg-
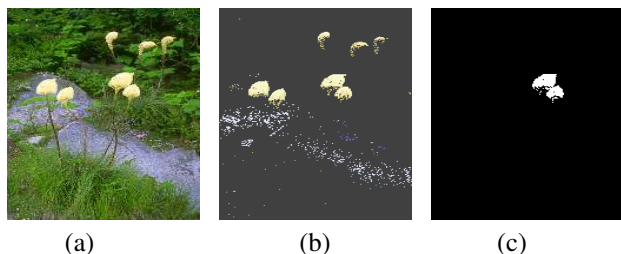
(a)　　　　　(b)　　　　　(c)

Figure 8. Detecting a reliable flower region : (a) original image (b) image left after deleting non-flower colors (c) largest valid segment

(a)　　　　　(b)　　　　　(c)

Figure 9. Background elimination : (a) original image (b) image left after deleting non-flower colors and the background color (white) (c) largest valid segment

ment in the binarized image is computed. The validity of the segment is tested to determine whether the choice of background colors was correct. Fig 9 shows an example of the final flower segment obtained when the color class *white* was deleted after being correctly identified as a background color.

This method of detecting background colors is not guaranteed to produce correct results. It will fail for images of the type shown in Fig 1(c), and may also fail for images of the type shown in (b) 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 image type (c), the hypothesis for the background color deletes the whole image. In image type (b), if the flower color is deleted instead of the background, only background pixels are left in the image. Since 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 is an indicator that the background color selection was wrong.

When feedback is obtained from the segmentation process that the background color chosen was incorrect, the color(s) 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. Fig 10 shows the intermediate steps in detail. From the analysis of the border of the segment obtained first, the color class *purple* is eliminated. 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 carried out after eliminating the new hypothesis for background color, the class *white*.

If no valid segments are found when any of the color classes present in the border are eliminated, one should be able to conclude that the image is of the type in Fig 1(c) and the flowers cover the full image. However, since we are looking at color *classes*, there is an alternative situa-
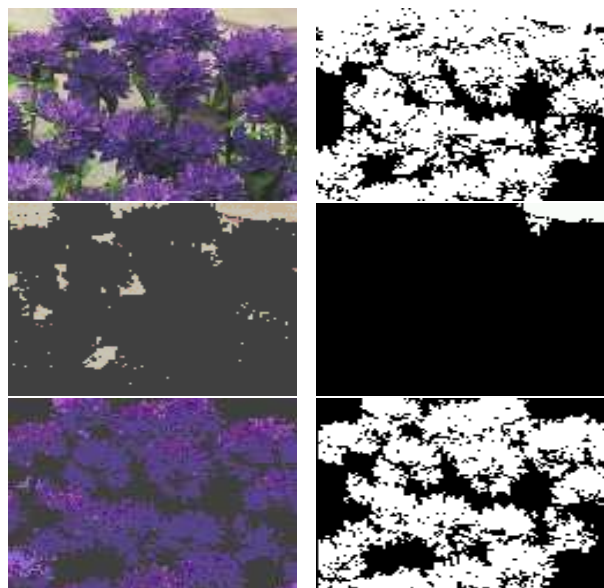
Figure 10. Recovery from erroneous background color selection : (First Row) Original image and segment found after deleting non-flower colors (Second Row) Result of deletion of the color class *purple* **which was hypothesized to be a background color and the largest segment obtained (which is not valid) (Third Row) Trying the new hypothesis that the color white is the background color and the valid segment obtained**

tion where the background is a different shade of the flower color and thus, belongs to the same class. So, we test for this situation by using color *names* to label the pixels instead of the color classes, and repeating the above procedure. An example is shown in Fig 11. 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 whole 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 the image is labelled using color names, the colors *HoneyDew* and *MintCream* (which are shades of white) are found from the border block analysis. Deleting these colors leaves the colors *LemonChiffon3* and *Ivory3* which are also shades of white. The remaining image shown in Fig 11(c) produces a valid segment which does not include any background.

When the background cannot be eliminated using any of these trials, the image is assumed to contain only the flower colors and the description is computed from the largest segment obtained after deletion of the non-flower colors.

The segmentation strategy produces erroneous results only when there are colored objects (excluding the non-
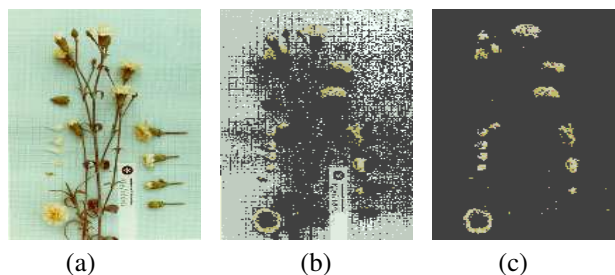

Figure 11. Using color names for labeling : (a) Original image (b) image left after deleting non-flower colors (c) result of eliminating background colors based on color *names*

flower colors) in the image which are more prominent than the flowers and when the flowers are located only along the margins of the image. Both situations have low probability in the flower patents database.

5 Indexing and Retrieval

The colors present in the segment identified as a flower region in the earlier section 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. Since there is a wide variety in the names that could be used for querying, the images are indexed by using both the X names and 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 is used to access the images by the color classes present in the images.

There is usually more than one color name 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 be considered.

5.1 Query by name

When a color name is provided as query, the X name index and the NBS color name index are searched 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 since it uses increasing numbers to indicate darker shades of the original color. For example, 'MediumPurple2', 'MediumPurple3' and 'MediumPurple4' are progressively darker shades of the original color 'MediumPurple'. Since 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,

a specific query using one of the defined X or NBS color names could also be issued which will require a knowledge of the valid names. In this case, the exact name is used from the indexes. The retrieved images are ranked by proportion - the flower with a larger proportion of the query color is ranked ahead of a flower with a smaller proportion of the query color. If more than one name is used in the query, a *join* (intersection) of the image lists retrieved for each of the query colors, is returned.

5.2 Query by example

When a flower image is used as a query, the user expects a close color match with the flower shown in the query. In this case, searching for each of the colors present separately and combining the lists often produces poor results. For example, a flower may appear to be a intermediate shade of pink because it consists of a combination of pixels of a darker shade and a lighter shade. Separate retrieval using the two shades present will retrieve a set of flowers which have both these shades, but flowers whose perceived shade does not match the query may be ranked high. This could happen since the relative proportions of the two shades was not taken into account when ranking and therefore, relative proportions of the two shades in the top retrieved flower could be quite different from the query.

In this case, we need to find a distance measure between the query flower and the retrieved flower which 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. The HSV coordinates for each X color is computed from its original RGB definition. A weighted average of the HSV coordinates of the X colors present in a color class is computed. The weights are proportional to the relative proportion of the color in the flower segment. For example, for a flower which 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 ($\frac{p_1 h_1 + p_2 h_2}{p_1 + p_2}$, $\frac{p_1 s_1 + p_2 s_2}{p_1 + p_2}$, $\frac{p_1 v_1 + p_2 v_2}{p_1 + p_2}$). The retrieved images are now ranked by the city-block distance of its average color in each of the color classes from the corresponding query color averages.

6 Experiments

The test flower database currently being used consists of 300 images. About 100 of the images are from actual flower patents from the U.S. Patent and Trademarks Office. We have added 100 images from CD-ROM collections with complex backgrounds beyond those encountered in images from patent applications to test the segmentation process. The rest are scanned from catalogs of flowering plants and

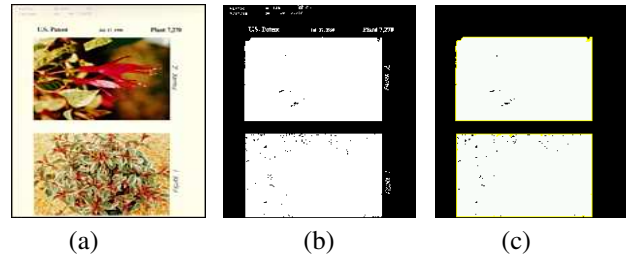photographs, including a few images of colored fruits which are treated the same way as flowers.

(a)   (b)   (c)

Figure 12. Detecting images on the patent form : (a) scanned page (b) image left after deleting background color (c) segments found

The pages from the patent forms are of the type shown in Fig 12(a), containing both text and images. Images are detected from the patent forms using the same strategy of deleting background colors and checking the remaining segments. However, in this case, there may be more than one segment found of significant size as shown in Fig 12(c). These segments are approximated by rectangles and the cropped image corresponding to each segment is added to the database.
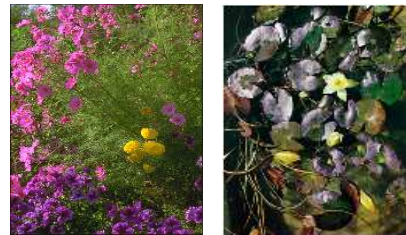

Figure 13. Images on which the segmentation algorithm produces errors

The flower segment identified by the iterative segmentation algorithm was checked for each of the database images and there were only two possibly erroneous segmentation results found for images shown in Fig 13. The segment formed by the pink flowers did not pass the centroid test and the yellow flower region was selected as the most significant segment. This is an image from the CD-ROM collection and 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 which may actually be the correct component of the patent since the flower is given very little emphasis in the image.

We tested the retrieval results obtained using 50 queries of different types. On 25 queries using color names, we
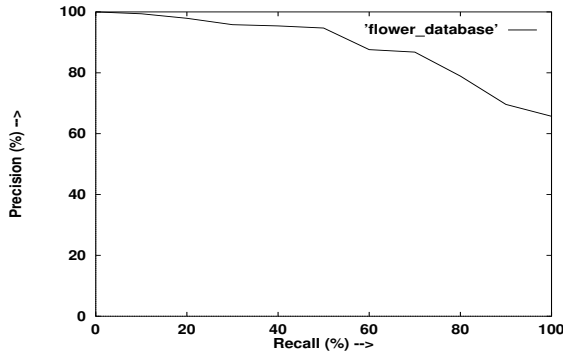
Figure 14. Recall-Precision graph for 25 queries by example on the flower patent database

checked that the retrieved flowers matched our perception of the color name used in the query. A more exhaustive evaluation was done for 25 queries using example images. The images relevant to the query were identified by scanning the database and recall and precision measures were computed. The recall-precision graph [43] obtained is shown in Fig 14. The average precision obtained was 88% and the precision at 100% recall was 66%. The latter figure is important in this application because it is important to find all relevant images even at the expense of checking a larger number of non-relevant ones.

Fig 15 shows the current user interface for querying by color. The color class can be selected from the left frame of the interface and the right frame displays the various shades of that color along with their names. A search can be performed 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 retrieved images are displayed at the bottom of the interface. Fig 16 shows the current interface for query by example. The example image can be selected by browsing through the database on the left frame or by selecting one of the retrieved images. An interface which accepts the user's images as query will be added for the real application. The example image selected is displayed in the right frame and the retrieved images are displayed at the bottom. This online interface to the system supporting both types of queries can be found at **http://cowarie.cs.umass.edu/~demo/FlowerDemo.html**

Fig 17 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 when querying using the color names 'orange' and 'ivory'. Only the top five images for each query are shown in this figure.

7 Conclusion and Future Work

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 be used to improve the retrieval performance for applications in many specialized image databases. We have proposed a methodology for using color-based and spatial domain knowledge to automatically segment and index a database of flower images using an iterative segmentation algorithm. A natural language color classification system is used to interpret color-based domain knowledge into rules for automatic segmentation of the region of interest from the background. The approach suggested here may be adapted to any database dedicated to images of known subject about which some domain knowledge is available.

Further work on the current project will include tests on a large database of the order of 10,000 flower images. Our goal is to verfiy 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. The use of color adjacency graphs [22] for distinguishing multi-colored flowers containing the same colors is also being considered. The use of this approach to other specialized databases (for example, birds) is also being investigated.

Acknowledgements

References

[1] J. R. Bach, C. Fuller, A. Gupta, and et al. The Virage image search engine: An open framework for image management. *Proceedings of SPIE*, 2670:76–87, 1996.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based segmentation using EM and its application to content-based image retrieval. *Sixth International Conference on Computer Vision(ICCV)*, pages 675–682, Jan 1998.

[3] C. C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, 1993.

[4] Q. Chen, H. Wu, and M. Yachida. Face detection by fuzzy pattern matching. *Fifth International Conference on Computer Vision(ICCV)*, pages 591–596, Jun 1995.
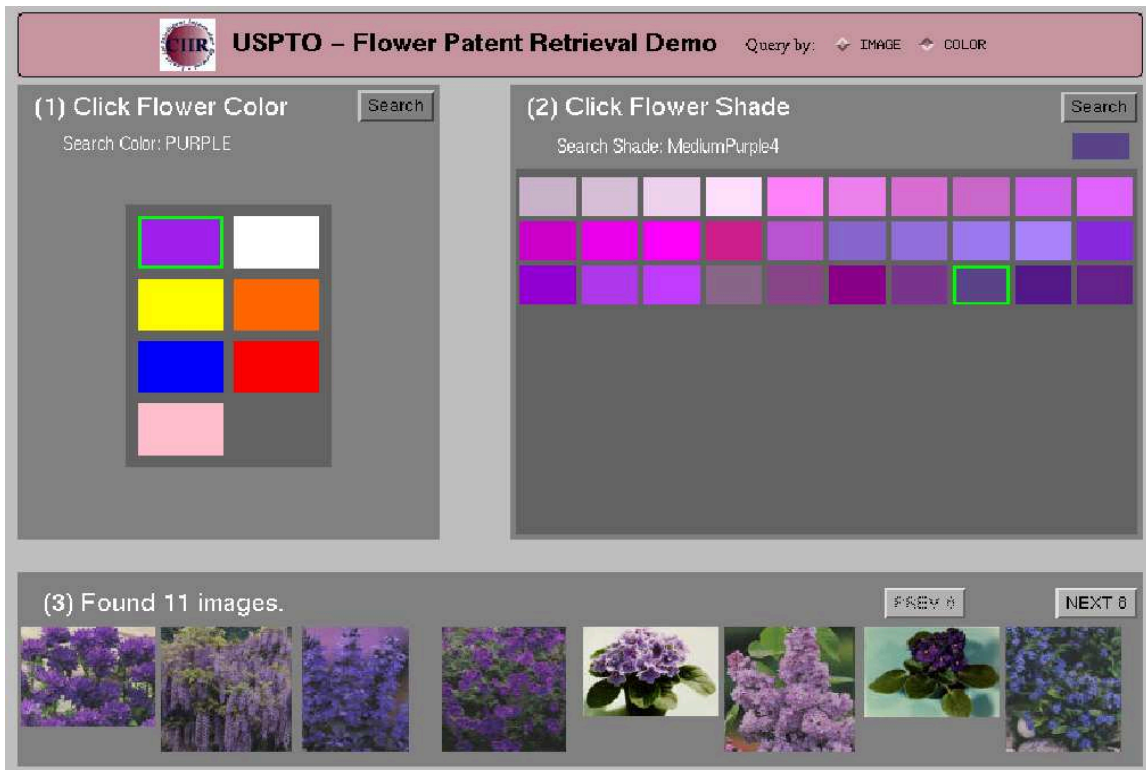
Figure 15. Retrieval by color name : The color shade selected here is 'Medium Purple' in the color class 'Purple'

[5] M. Das, E. M. Riseman, and B. A. Draper. FOCUS: Searching for multi-colored objects in a diverse image database. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 756–761, Jun 1997.

[6] J. P. Eakins, K. Shields, and J. Boardman. ARTISAN - a shape retrieval system based on boundary family indexing. *SPIE Proc. on Storage and Retrieval for Image and Video Databases*, 2670:17–28, Feb 1996.

[7] M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. *Fourth European Conference on Computer Vision(ECCV)*, pages 72–77, 1996.

[8] D. Forsyth, J. Malik, and R. Wilensky. Searching for digital pictures. *Scientific American*, 276(6):72–77, Jun 1997.

[9] D. A. Forsyth and M. M. Fleck. Body plans. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 678–683, 1997.

[10] Y. Gong and M. Sakauchi. Detection of regions matching specified chromatic features. *Computer Vision and Image Understanding*, 61(2):263–269, Mar 1995.

[11] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer Magazine*, 28(9):61–70, Sep 1995.

[12] D. Hearn and M. P. Baker. *Computer Graphics*. Prentice-Hall, 1994.

[13] J. Huang, S. Ravi Kumar, M. Mitra, W. J. Zhu, and R. Zabih. Image indexing using color correlograms. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 762–768, Jun 1997.

[14] Q. Huang, B. Dom, D. Steele, J. Ashley, and W. Niblack. Foreground/background segmentation of color images by integration of multiple cues. *International Conference on Image Processing*, 1:246–249, 1995.

[15] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.

[16] 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*, (553), Nov 1 1955.

[17] P. Kelly, M. Cannon, and D. Hush. Query by image example : the CANDID approach. *SPIE Proc. on Storage and Retrieval for Image and Video Databases*, pages 238–248, Feb 1995.

[18] A. Khotanzad and A. Bouarfa. Image segmentation by a parallel, non-parametric histogram based clustering algorithm. *Pattern Recognition*, 23(9):961–973, 1990.

[19] F. Liu and R. W. Picard. Periodicity, directionality and randomness : Wold features for image modeling and retrieval. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, Jul 1997.

[20] S. Liu and R. Yang. Multiresolution color image segmentation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 16(7):689–700, Jul 1994.
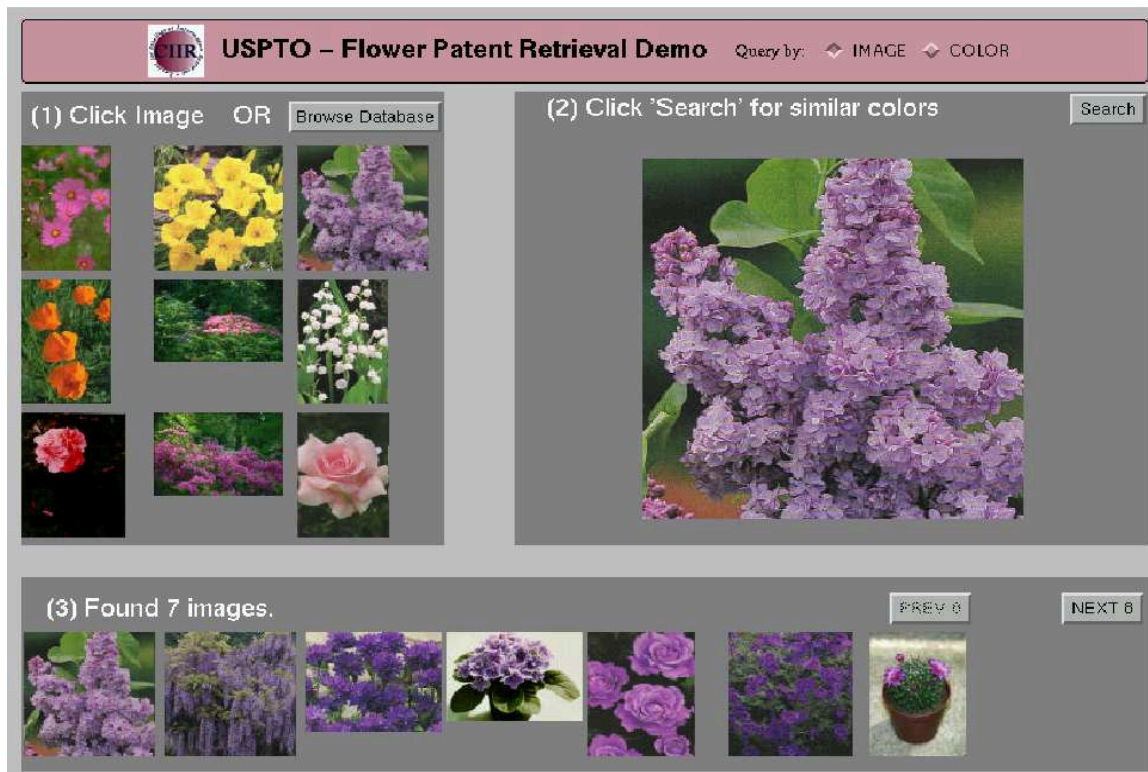
Figure 16. Retrieval by example : The query selected is shown on the right

[21] W. Y. Ma and B. S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35–51, Jan 1996.

[22] J. Matas, R. Marik, and J. Kittler. On representation and matching of multi-coloured objects. *Fifth International Conference on Computer Vision(ICCV)*, pages 726–732, Jun 1995.

[23] R. Mehrotra and J. E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, Sep 1995.

[24] B. Mehtre, M. Kankanhalli, and W. Lee. Shape measures for content-based image retrieval : A comparison. *Information Processing and Management*, 33, 1997.

[25] C. Nastar, M. Mitschke, C. Meilhac, and N. Boujemaa. Surfimage: A flexible content-based image retrieval system. *Proc. ACM Multimedia*, Sept 1998.

[26] W. Niblack, R. Barber, and et al. The QBIC project: Querying images by content using color, texture and shape. *SPIE Proc. on Storage and Retrieval for Image and Video Databases*, 1908:173–187, Feb 1993.

[27] V. Ogle and M. Stonebraker. Chabot : Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, Sep 1995.

[28] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 84–90, Jun 1994.

[29] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[30] P. J. Phillips, H. J. Moon, P. Rauss, and S. A. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 137–143, Jun 1997.

[31] R. Picard, T. Minka, and M. Szummer. Modeling subjectivity in image libraries. *International Conference on Image Processing*, Sept 1996.

[32] S. Ravela and R. Manmatha. Image retrieval by appearance. *Proc. of Intl. ACM Conf. on Research and Development in Information Retrieval(SIGIR)*, pages 278–285, Jul 1997.

[33] S. Ravela and R. Manmatha. On computing global similarity in images. *Workshop on Applications of Computer Vision(WACV)*, pages 82–87, Oct 1998.

[34] S. Ravela, R. Manmatha, and W. B. Croft. Retrieval of trademark and gray-scale images using global similarity. *CIIR Technical report*, MM-25, 1999.

[35] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, Jan 1998.

[36] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox. Annotation of natural scenes using adaptive color segmentation. *Proceedings of SPIE*, 2421:72–80, 1995.

[37] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, Jun 1997.

[38] A. F. L. Serafim. Segmentation of objects in images of natural coloured surfaces. *Proceedings of SPIE*, 2785:213–222, Jun 1996.

[39] J. R. Smith and S. F. Chang. Visually searching the web for content. *IEEE Multimedia*, 4(3):12–20, Sep 1997.

[40] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[41] D. L. Swets and J. Weng. Using discriminant eigen features for retrieval. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 18:831–836, Aug 1996.

[42] T. F. Syeda-Mahmood. Data and model-driven selection using color regions. *Second European Conference on Computer Vision(ECCV)*, pages 115–123, May 1992.

[43] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[44] A. R. Weeks and G. E. Hague. Color segmentation in the hsi color space using the K-means algorithm. *Proceedings of SPIE*, 3026:143–154, Feb 1997.

[45] W. Woelker. Image segmentation based on an adaptive 3D-analysis of the CIE-L*a*b* color space. *Proceedings of SPIE*, 2727(3):1197–1203, 1996.

[46] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, Inc., 1982.

Figure 17. First five retrieved images : Query for rows 1-3 is the first image retrieved in the row, query for row 4 is the color 'orange', query for row 5 is the color name 'ivory'