

# Indexing Flowers by Color Names using Domain Knowledge-driven Segmentation\*

Madirakshi Das

R. Manmatha

Edward M. Riseman<sup>†</sup>

Multimedia Indexing and Retrieval Group

Department of Computer Science

University of Massachusetts

Amherst, MA 01003

e-mail : mdas,manmatha,riseman@cs.umass.edu

## Abstract

*We describe a solution to the problem of indexing images of flowers for searching a flower patents database by color. We use a natural language color classification derived from the ISCC-NBS color system and the X Window color names to effectively use the domain knowledge available, provide perceptually correct retrieval and allow natural language queries. We have developed an automatic iterative segmentation algorithm with knowledge-driven feedback to isolate a flower region from the background. 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. We demonstrate the effectiveness of the strategy on a test database.*

## 1. Introduction

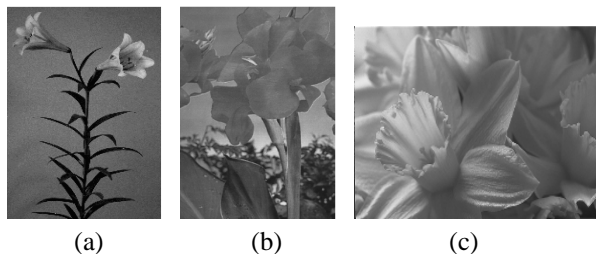
The problem of meaningful retrieval from image databases has generated a great deal of interest in recent years. Most retrieval algorithms have targeted a general image database which may contain diverse types of images [1, 5]. However, there is a growing number of large image databases which are dedicated to specific types and subjects. When using general-purpose retrieval strategies on these databases, it is easy to lose sight of characteristics of

\*This material is based on work supported in part by the National Science Foundation, Library of Congress and 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 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor(s).

<sup>†</sup>This author is with the Computer Vision Laboratory.

the domain which could be used to substantially improve the retrieval performance. There may also be special querying requirements in domains covered by the database.

This work is motivated by the need for indexing a database of flower images which were submitted as a part of applications for flower patents to the U.S. Patents and Trademark Office. The aim is to provide a tool to check whether flowers similar in color to a given flower are present in the patents database or be able to list all flowers of a given color by specifying a color name.



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

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 : segmentation of the flower from the background and the description of the flower color in a form which matches human perception and allows flexible querying by example and by natural language color names.

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

mation 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 such a mapping using color names from the ISCC-NBS [4] system and the color names defined in the X Window<sup>TM</sup> 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 necessary after restoring the image to its earlier state.

The next section surveys related work; section 3 discusses the mapping from the RGB space to natural language color names. The domain knowledge available and a segmentation strategy utilizing it are explained in section 4. Section 5 discusses indexing and retrieval from the database using different types of queries. Section 6 describes experiments carried out to test the system.

## 2. Related work

There has been work on perceptual organization of the color space in the area of image indexing [8] and in color science [11] without mapping the perceptual entities to natural language color names. In the reverse approach, color domain knowledge has been mapped to the 3D color space [2, 6] in applications like face identification using skin tones, and automatic target recognition where the part of the color space which corresponds to the object of interest is identified.

There has been a lot of work in the area of color image segmentation and indexing using histograms [7] in different color spaces [9, 10]. However, these techniques produce color segments which may not correspond to single objects in the scene and also, there is no way of discriminating foreground and background elements. Automatic foreground/background disambiguation based on multiple features like color, intensity and edge information has been studied in [3], but these techniques work well on relatively smooth backgrounds and objects with sufficient contrast.

## 3. Mapping from color space to names

We need a mapping from 3-D color space to color names which agrees with human perception of colors. We use two independent methods for the mapping :

(i) the ISCC-NBS color system which produces a dense map

from the Munsell color space to names. The ISCC-NBS system uses a standard set of 12 base hue names [Fig 2] and generates 267 color names using hue modifiers. This color system can be easily decomposed into a hierarchy of colors where we may use the full color name, partial names, base hues or coarser classes comprised of groups of base hues.

(ii) colors defined by the X Window system which provides a sparse mapping from the RGB space to 359 names. The color definitions in X provide commonly used alternative names for the colors e.g. ‘light brownish yellow’ in the ISCC-NBS system is the color ‘khaki’ in X.

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

**Figure 2. Classes derived by grouping ISCC-NBS hue names and adding neutral colors**

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 3. Color representations used**

We have used the HSV color space discretized into 64x10x16 bins as an intermediate space between the RGB space and color names to reduce the number of colors as well as have perceptually similar colors in the same neighborhood. Each point on the discretized HSV space is mapped to a color defined in X, using the nearest color name (by city block distance) for points with no exact map. Each point is also mapped to the ISCC-NBS name [Fig 3]. The ISCC-NBS name is used to produce a color hierarchy so that queries can be general (e.g. blue) or specific (e.g.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. 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.

### 4.1. Use of domain knowledge

We can utilize color-based domain knowledge of the type discussed earlier using the mapping to natural language color names. Most of the frequently occurring elements of the background can be eliminated by deleting pixels belonging to color classes which do not represent colors of flowers. Black and gray pixels are mostly from shadow regions in the image, brown pixels from shadows, branches, soil etc. and green pixels are from foliage.

Another observation helpful in identifying background regions is that background colors are usually visible along the periphery of the image. 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 also 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. The background may have other colored objects but they will not usually dominate the main subject, which is the flower.

Since the flower images were submitted as part of a patent application, there is a single type of flower, though there may be many of them in the image. Due to this, a single prominent flower segment 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. 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 is comprised solely of flower areas. This entails the generation of a hypothesis identifying 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.

The outline of the algorithm used to produce a segment from which the flower color is estimated is shown in Fig 5.

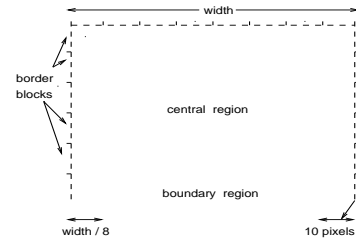


Figure 4. Definitions of image regions

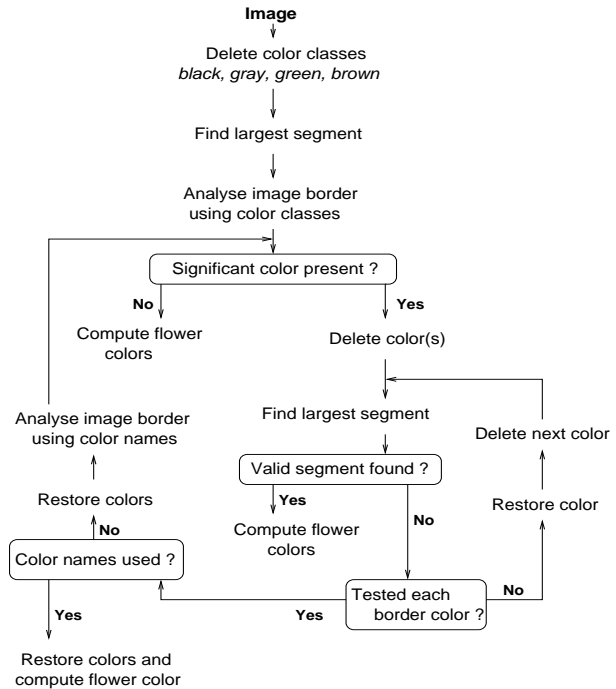
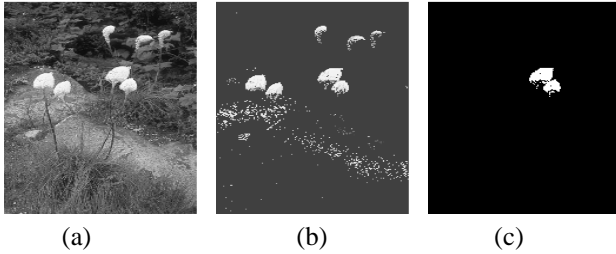


Figure 5. Overview of segmentation strategy

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

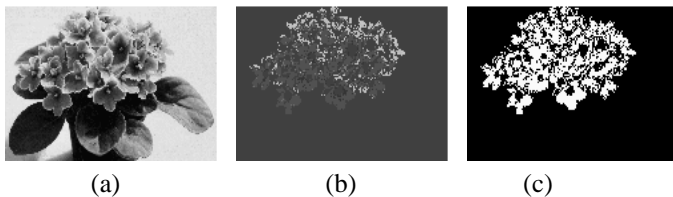
We use two criteria for evaluating whether a segment produced is valid; it should be of a minimum size which is based on the size of the largest segment obtained after deleting the non-flower color classes, and its centroid should fall within the ‘central region’ of the image as defined in Fig 4. These 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 e.g. the rock in Fig 6(a), resulting in the final flower segment shown in Fig 6(c).

Further processing is required when the largest segment contains background colors in addition to the flower regions. The presence of background colors is detected by



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

analysing the color composition of the image margins considering the pixels covered by the largest segment only. The margins of the image are divided into border blocks which are narrow vertical rectangles as shown in Fig 4. 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.

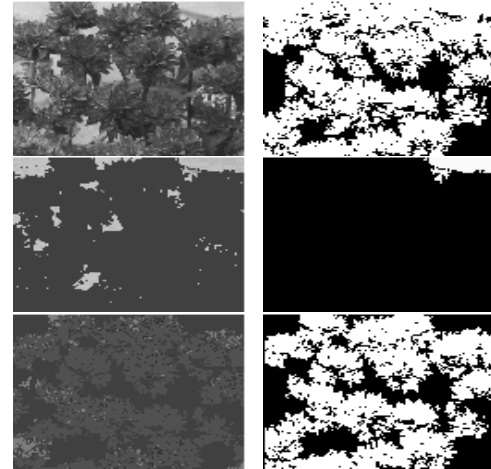


**Figure 7. Background elimination :** (a) original image (b) image left after deleting non-flower colors and the background color (white) (c) largest valid segment

After eliminating all the pixels of the hypothesized background color(s), the largest segment in the binarized image is computed. Fig 7 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 strategy 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 1(b) if there is sufficient overlap between the flower and the margin. 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 there is feedback that the background color chosen was incorrect, the color(s) is restored and the hypothesis of

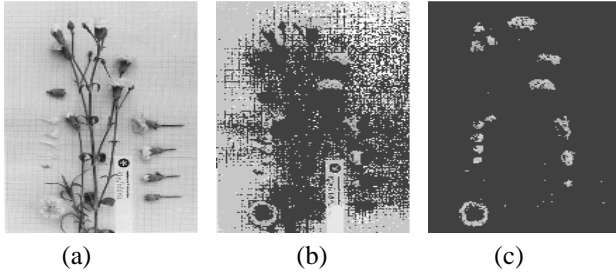


**Figure 8. Recovery from erroneous background color selection :** (First Row) Original image and segment found after deleting non-flower colors (Second Row) Result of deletion of the color class *purple* and the largest segment obtained (which is not valid) (Third Row) Trying the new hypothesis that the color *white* is the background color

each border background color is tested separately. Fig 8 shows the intermediate steps in detail. From the analysis of the border blocks, the color class *purple* is eliminated. This results in an invalid segment whose centroid falls in the boundary region. A valid segment is found when *purple* is restored and a new hypothesis for background color, the class *white*, is tested.

If no valid segments are found when any of the color classes present in the border are eliminated, we could conclude that the flowers cover the full image, the image is of the type in Fig 1(c). However, since we are considering color *classes*, an alternative situation must be examined where the background is a different shade of the flower color and thus, belongs to the same class. We test for this situation by using color *names* to label the pixels, repeating the above procedure. An example is shown in Fig 9 where the color class *white* is found to be the background color. However, no significant segments are left on deleting *white* pixels. When the image is relabelled using color names, the colors *HoneyDew* and *MiniCream* (which are shades of white) are marked as background colors from the border block analysis. The remaining image (Fig 9(c)) consists of other shades of white and 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 deleting the non-flower colors.



**Figure 9. 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**

The segmentation strategy produces erroneous results only when there are colored objects (excluding the non-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 indexing is based on the types of queries we would like to support, which includes queries using natural language color names. The flower images are indexed by X color names, ISCC-NBS color names and color classes using index tables constructed offline.

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 e.g. ‘MediumPurple2’ is a darker shade of the original color ‘MediumPurple’. However, a specific query, without using variants, can also be issued using one of the defined X or NBS color names. The retrieved images are ranked by the proportion of the query color contained. 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. Instead, we use 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 of each color class present in the query using the HSV coordinates for each X color found in the class. e.g. 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  $(p_1h_1 + p_2h_2/p_1 + p_2, p_1s_1 + p_2s_2/p_1 + p_2, p_1v_1 + p_2v_2/p_1 + p_2)$ . The retrieved images are now ranked by 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 we are currently using consists of 250 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 more complex backgrounds than those encountered in images from patent applications to test the segmentation process. The rest are scanned from catalogs of flowering plants.

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. In one of the images from the CD-ROM collection, the dominant flowers occurred only along the margins and a different type of flower in the centre of the image was picked as the flower segment. In the second case, the colored leaves of a flower constituted the most significant segment, but this 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 and found that the retrieved flowers matched our perception of the color name used in the query or the color of the example flower. Fig 10 shows some sample retrieval results obtained. An online interface to the system supporting both types of queries can be found at <http://cowarrie.cs.umass.edu/~demo/FlowerDemo.html>. The images in this paper can be viewed in color by following links from the same site.

## 7. Conclusion

We have proposed a natural language color classification and retrieval system which can be used to utilize domain



**Figure 10. Examples of image retrieval : (top two rows) Query by example where the first image retrieved was the query, (bottom row) Query by name using 'medium purple'**

knowledge encoded into rules for automatic segmentation of the region of interest from the background. We describe an iterative segmentation algorithm for identifying flower regions. However, our approach can be adapted for any database dedicated to images of known subject about which some constraining domain knowledge is available. Further work on the current project will include construction of an integrated user interface for the database and tests on a large set of flower images.

## Acknowledgements

We would like to thank Thomas Michel for building the online user interface for this system. We also thank Bruce Croft and CIIR for supporting this work.

## References

- [1] 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*, pages 756–761, Jun 1997.
- [2] Y. Gong and M. Sakauchi. Detection of regions matching specified chromatic features. *Computer Vision and Image Understanding*, 61(2):263–269, Mar 1995.
- [3] 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.
- [4] 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.
- [5] W. Niblack, R. Barber, and et al. The qbic project: Querying images by content using color, texture and shape. *Proceedings of SPIE*, 1908:173–187, Feb 1993.
- [6] 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.
- [7] A. F. L. Serafim. Segmentation of objects in images of natural coloured surfaces. *Proceedings of SPIE*, 2785:213–222, Jun 1996.
- [8] T. F. Syeda-Mahmood. Data and model-driven selection using color regions. *Second European Conference on Computer Vision*, pages 115–123, May 1992.
- [9] 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.
- [10] 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.
- [11] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, Inc., 1982.