STUDIES IN THE USE OF COLOR FOR IMAGE INDEXING AND RETRIEVAL IN SPECIALIZED DATABASES

A Dissertation Presented

by

MADIRAKSHI DAS

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2001

Department of Computer Science

STUDIES IN THE USE OF COLOR FOR IMAGE INDEXING AND RETRIEVAL IN SPECIALIZED DATABASES

A Dissertation Presented

by

MADIRAKSHI DAS

Approved as to style and content by:

Edward M. Riseman, Chair

R. Manmatha, Member

W. Bruce Croft, Member

Patrick A. Kelly, Member

W. Bruce Croft, Department Chair
Department of Computer Science

To my little girl, Promiti

ACKNOWLEDGMENTS

I would like to thank my advisor, Edward Riseman, for his constant support of my work. At all of our meetings, Ed emphasized the positive aspects of my work. He let me know that he considered my work to be new and significant; which motivated me to carry on when the progress was slow. Despite his busy schedule, he took the time to suggest numerous improvements to the initial draft of my dissertation; and the dissertation is much more readable, thanks to his suggestions.

I am very grateful to Manmatha, who has been a mentor to me in many ways. I joined the Multimedia Indexing and Retrieval group just after the start of my doctoral research, and he has been a source of support and guidance throughout. After I left Amherst, he made sure that my thesis work was on track by encouraging me to keep at it even when pressures from work and home apparently left me with no time to devote to my thesis work. He checked on my progress regularly, and was very diligent in reading and correcting my work. My thesis would probably still be unfinished, if it was not for his commitment to making my graduation a reality.

I would also like to thank my committee members, Bruce Croft and Patrick Kelly, for patiently reading and commenting on my thesis and for their positive remarks on my research work. I would like to thank Bruce Draper for introducing me to the world of research by pointing out the open problems in a particular paper which finally lead to the start of my thesis work.

I wish to extend a special thank you to my husband, Suprakash Datta, who went through the ups and downs of doctoral research along with me. He provided me the initial motivation for enrolling in graduate studies and also the motivation to continue till the end. He was the system administrator of my home office, who set up and maintained our personal computers so that I could work at home in the same environment as at school. I could ask him for any technical help I needed, knowing that he would either know the information or know where to find it.

I would like to thank my parents who flew thousands of miles to visit me so that I could concentrate on finishing my thesis, leaving housework and baby-sitting duties to them. My apologies to my little daughter who had to spend time playing on her own because "Mama working" as she put it. Though, as an infant, she practically brought my research work to a grinding halt by keeping me awake at night, her arrival focussed my efforts and made me more efficient. She has also been instrumental in countering the stresses involved in doctoral research.

Thanks are due to Tom Michel for creating the web-based demo for the flower database retrieval system, and to Jonathan Lim for the demo of advertisement database retrieval. Thanks to all the people, staff and students, of the Center for Intelligent Information Retrieval for making it a wonderful place to interact and work together.

Finally, thanks to Kate and Laurie for taking care of all the paperwork, signatures, contracts etc that are part of graduate school.

This material is based on work supported in part by the Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by 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 grant number 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.

ABSTRACT

STUDIES IN THE USE OF COLOR FOR IMAGE INDEXING AND RETRIEVAL IN SPECIALIZED DATABASES

SEPTEMBER 2001

MADIRAKSHI DAS

B.Tech., INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR, INDIA

M.Tech., INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR, INDIA

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Edward M. Riseman

The content of an image is often associated with the main object(s) present in an image. Therefore, for effective content-based retrieval, the database images need to be indexed by features extracted from the object of interest, ignoring any irrelevant image background. In this work, we propose content-based retrieval strategies focusing on the use of color-based features for specialized image domains where the performance of general-purpose color image retrieval techniques is poor. The retrieval performance is improved by taking the special characteristics of the domain into account to extract the object of interest when possible, or capture the properties of the important objects present in an image when it is not possible to extract an object of interest a priori.

Three test domains are selected which have very different characteristics requiring different retrieval strategies. These domains are representative of a larger class of specialized image databases which have similar characteristics. A two-phase image retrieval engine which is robust in the presence of interfering backgrounds and large variations in the size of the query object in the target images, is proposed for an advertisement images domain where there are extreme variations in backgrounds and the size of the object of interest. An iterative segmentation algorithm for extracting the object of interest is proposed when there is useful domain knowledge available about the subject of the database images, as in the flower images domain tested in this dissertation work. Automatic segmentation of the object of interest is extended to a database of bird images where there is no subject-specific domain knowledge available, using general observations true for any image where the object of interest is prominent in the image.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 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. The medium of information has also shifted from text-based to multimedia information which includes images, audio and video, in addition to text. To be able to use this information effectively, people require tools to manage the information; including tools for searching, retrieving and classifying it.

Image retrieval has been an active area of research since the early '90s. Recently, the proliferation of digital cameras and scanners has lead to an explosion of digital images, creating personal image databases large enough to require efficient image retrieval techniques for easy access and organization. The number of commercial and informational image databases which need to support image search and browsing to be used effectively, have also multiplied. As more application areas are encountered [16, 24], it is increasingly important to find an efficient solution to the problem of extracting images relevant to a user from a large image database. Since the end user of image retrieval systems is usually a human, 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 impossible to define accurately, much less 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 *content* of the image, so an image description which captures the image content is more likely to produce results matching the expectations of a user.

The determination of image content can be subjective, but if there are specific objects present in the image, a human user usually associates content with the main object(s) present in the image. For example, we may identify a picture as that of a "flower", "bird", "house" etc. Therefore, to provide true content-based retrieval, we need to be able to focus on the object of interest in the image. In this thesis, we formulate methods for extracting the object of interest when possible, and methods for capturing the properties of the important objects present in an image when it is not possible to extract an object of interest a priori. This enables indexing the database using features extracted from the object of interest alone, improving upon existing image retrieval systems which are influenced by irrelevant features from the background. We propose different retrieval strategies for domains where the object of interest varies widely in size and is embedded in a lot of background clutter; and where the object of interest occupies a very prominent place in the image. Our goal is to develop content-based retrieval techniques for some commonly encountered classes of specialized image databases, where the performance of general-purpose image retrieval techniques are poor and could be improved by taking the special characteristics of the domain into account.

Traditionally, image databases have been manually annotated using textual keywords. Most commercial databases of stock photographs currently available on the world wide web still employ captioning by the photographer for indexing the collection. This enables the use of techniques developed for text-based information retrieval in the image domain. However, manual annotation is slow and expensive 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. The main difficulty in replacing a text-based image description with one extracted from the image lies in the lack of available semantic units in an image, 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 over a varying and complex spatial context. Therefore, segmenting images into such meaningful units (objects) is in general an unsolved problem in computer vision. In this thesis, we propose a framework for automatically segmenting the foreground object from the background elements when domain knowledge is available about the object or about the image in general.

In the absence of semantic-level image descriptors, an image is usually described in terms of low-level features or attributes which can be directly computed from the image pixels. 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. Image retrieval systems [3, 49, 48] using an array of low-level image features such as color, texture, edge description and composition have been developed, focussing on retrieving images from general image collections. While these work well when the low-level features correlate well with the subject of the images and the subject dominates the image, they can produce completely useless results when the subject of the image is small in proportion to the image size or when there is significant background present in the images. Most of the failures of retrieval systems targeted at general image collections can be attributed to three assumptions :

(a) that the object of interest occupies most of the image,

(b) that the background present in the database images is insignificant,

(c) that low-level features are correlated to the object of interest.

For example, a query showing a flying bird may produce a list of images of the sky, many of which do not even contain a bird. A query showing a red car may produce images of red flowers and birds, a case where the low-level feature (color) is not unique to the object of interest.

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! It is not always clear, especially for techniques focused on general image collections, what the evalution criteria are, since similarity between images is sometimes hard to define.

In image retrieval applications involving specialized domains, however, the user's needs are often well-defined. Specialized databases contain images dedicated to specific types and subjects of images. Examples include databases of birds, flowers, sports photographs, scenery, commercial products, cars, family pictures etc. In some of the above examples, all images in the database represent a particular type of object (like flowers and birds), while in other examples there is an abstract theme (scenery, sports photographs). In each case, there is some unifying element which links the images in the database. 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. Though a general-purpose image retrieval engine suggests that the database may contain any type of image, such systems may not do as well as expected by the user (or fail altogether) on specialized, constrained domains because of their built-in assumptions and the fact that they do not take any of the special features of the domain into account. For example, in a database of family pictures, a user may want to find pictures of a particular family member, an application in which only the faces in the picture are important. So global image matching based on any low-level feature(s) would produce very poor results. In this thesis, we avoid the pitfalls of the existing image retrieval algorithms by targeting the growing number of specialized image databases. In these scenarios, we are able to make valid assumptions about the database characteristics and exploit these characteristics to provide superior retrieval performance for targeted applications.

We believe that restricting image retrieval to specialized collections of images or to specific tasks is more likely to be successful and useful, because of many factors. Many image attributes like color, texture, shape and "appearance" may often be directly correlated with the semantics of the problem. For example, machine parts can be distinguished on the basis of their shape, commercial products can be identified by their color, texture could be used to distinguish animals with different types of fur, and a person's appearance is uniquely defined. These examples 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. In general image 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.

The restriction to specific domains does not make the task any less interesting, since the goal now is to provide better retrieval than what is possible using general-purpose algorithms. Specialized databases can be categorized into different classes based on the common characteristics of the database images in terms of knowledge about object size, color or location, presence or absence of background and the type of background. A retrieval strategy designed for a particular database should work well on other databases with similar characteristics.

1.1 Use of color in content-based retrieval

A variety of low-level image characteristics, including color [73], texture [37], shape [43] and filter response [59], have been used as features for indexing and matching images for image retrieval. When the target image database contains color images, color is an obvious choice for indexing because of its perceptual significance. There is a lot of interest in using color as a recognition cue because as a feature, it is largely independent of view, size and image resolution. In general, color-based retrieval works much faster and is computationally simpler than methods based on other low-level features. Existing image retrieval systems have used other low-level features in addition to color [48, 57], but examples where color has not been used even when the target images are in color are rare. Even in applications of image retrieval like face recognition, which is very specialized, color is often used as a pre-filter to detect skin regions in an image [8].

However, low-level attributes like color must be used with care if they are expected to be correlated with the object of interest. Some of the main problems associated with the use of color for content-based retrieval are discussed below.

1.1.1 Weak correlation between color and object of interest

Since general image databases contain a wide variety of images, in many cases, color is not a relevant feature for a particular image subject. For example, a query showing a red car would return red cars and other red objects like fruits and flowers, and ignore images of the same make and model of car in other colors, which are semantically relevant to the query. Clearly, in this case, color is not an important descriptor for the object of interest. However, since the database images are not classified by subject, the same feature set is used for all subjects, regardless of relevance to the subject.

1.1.2 Background colors

Since most systems do not attempt to make a distinction between colors from the main subject of the image and irrelevant background, retrieval results often do not match a human's expectation (which is based on the main subject of the image) when there is a lot of background in the image. For example, color is clearly an important attribute for flowers. However, with the naive use of color to describe the database images, a query image of a flower against a background of green leaves may not be able to retrieve images of the same flower against a background of soil or in a close-up without any background. This is because the query contains green areas which are given equal importance as the flower regions. The presence of backgrounds is a major problem which needs to be handled intelligently before retrieval can be effective.

1.1.3 Large variations in size

Since most retrieval strategies equate the area occupied by a color in the image with the importance of the color, poor retrieval results are obtained when the object of interest is small with respect to the target image or the query image. For example, a user will not be able to retrieve images of close-up views of an object by providing a small image of the object embedded in background. Conversely, a close-up view of an object posed as a query will not be able to retrieve database images of the object embedded in different backgrounds. In most cases, only images showing the object at nearly the same size as that in the query will be retrieved, which would exclude many true matches if the database images depict an object at widely varying sizes.

1.1.4 Color constancy

Color constancy is the ability of humans to perceive the same apparent color in the presence of variations in illumination, even though the physical spectrum of the perceived light changes with illumination [84]. In the absence of such an adaptation, digital images record colors differently under varying lighting conditions. This problem affects all color-based retrieval methods. The illumination model required to solve this problem is rarely available for any database image. The choice of the three-dimensional color space used to represent image color affects the sensitivity of the color representation to differences in illumination. Different axes of the color space may have different sensitivities to lighting variation. Specular highlights and shadows produce drastic changes in the measured color. These issues need to be handled if a wide variety of illumination conditions is expected in the database images.

1.1.5 Other variations in color

There are often variations in color between images due to factors such as image quality, poor color balance and mode of acquisition. This is more common in images taken by amateurs and in images from non-commercial databases. There may be obvious color casts in the image, making objects appear more greenish or reddish. There may be poor color quality due to the use of a low bit-depth in color images (e.g. 8-bit color description instead of 24 bit). Even hardware configurations like the digital camera and the scanner settings could affect the color depicted by the image.

1.1.6 Too few or non-unique colors

Even when the color of an object is relevant, there may not be enough discriminatory power in the color signature of the object of interest. For example, in a database of apes, the range of colors may be very limited, resulting in poor discrimination between different species of apes. In a database of landscapes, there may be a limited number of colors corresponding to the sky, vegetation and rocks which occur in a large proportion of images. Color-based retrieval would not be very effective for these types of databases.

1.1.7 Subjective aspects of color perception

It is usually difficult to get different human users to agree on the correctness of color-based retrieval results. This happens when the system uses a finer or coarser color discrimination than the human does. However, since the granularity of discrimination between colors may vary from person to person, it may not be possible to provide perceptually correct retrieval across all people.

We have chosen to study the use of color in a variety of databases where the color signature of the object of interest is consistent across images. We propose solutions to some of the problems listed above for specialized image databases. In particular, we investigate ways of eliminating the effect of background colors and variations in the size of the object of interest.

1.2 Goals of the thesis

The overall goal of this work is to develop content-based retrieval techniques for some commonly encountered classes of image databases, concentrating on the effective use of color and color-based features for indexing images. Our aim is to find database images which depict objects similar to that featured in the query by matching only the features extracted from the object of interest, and not the whole image. We propose to meet this goal while maintaining database search speeds fast enough to work with an online user interface i.e. where the user waits for the results.

We select three different test domains and propose effective color-based image retrieval strategies for each domain. The choice of test databases for this thesis has been made on the basis of three criteria :

- There is scope for improvement over general-purpose image retrieval strategies.

- There are significant differences between the test databases which require different strategies.
- The databases are examples of a larger set of databases with similar characteristics.

1.2.1 Domain I: Advertisement images

The first category of databases we study may have the object of interest in a wide variety of sizes and locations in the target images, embedded in a large amount of complex, interfering backgrounds. Background includes other objects which are present in the image, and these could be more prominent than the object of interest; so that the object of interest cannot be extracted a priori (i.e. before a query is posed). General purpose retrieval strategies are not effective in this case, because the object of interest may not be prominent in the image and there may be a lot of background clutter. For example, the image in Figure 1.1 (a) shows an advertisement of the product “Breathe Right”. The product, which is the subject of this advertisement, occupies a very small portion of the image. The image is dominated by the “background”, the picture of Mona Lisa. The closest match found by a general-purpose retrieval strategy that indexes images by their global color histograms is shown in Figure 1.1 (b). It is clear that the match was obtained primarily by matching colors from the background. Figure 1.1 (c) shows a true match which contains the object of interest, but has a completely different color distribution from the original image. In this scenario, our goal is to propose techniques which describe all the colors in the image accurately; with methods for fast and efficient filtering of the description to remove background elements once the query image identifying the object of interest is provided.

We use a database of advertisement images scanned from magazines to test our retrieval strategy in the presence of large variations in size and background. The task

(a) (b) (c)

Figure 1.1. (a) Example of an image with a small object of interest and a lot of background (b) closest match obtained by global histogram matching (c) a true match

of retrieving all advertisements featuring a given product is particularly complex, since the queried object may appear in candidate images in various sizes and orientations with a wide variety of background colors and forms. In most of the advertisement images, the products do not spatially dominate the image, nor are they necessarily in the center. There is no concept of foreground and backgound - what is backgound clutter for this application may actually be the foreground of the image. Consequently, no focus-of-attention pre-segmentation is possible.

1.2.2 Domain II: Flower images

In the second category of databases studied, there is a lot of domain knowledge available about the object of interest and the backgrounds are known to be simple. However, the object size and location are still variable. Figure 1.2 shows an example of this scenario. Both images in the figure depict white water lillies, but while the first image is dominated by the green leaves, the second image is a close-up of the flower. Though these two images should match based on semantics, they do not match when using general-purpose color histogram-based matching because they have very different color distributions. These images would match based on color if only the object of interest (the flower) was considered when indexing. In this case, our goal is to formulate a method for automatically segmenting the object of interest from the background, exploiting the domain knowledge available. Features from the segmented object can then be used to index the database, providing retrieval based on an accurate description of the objects of interest even on images with significant background.

Figure 1.2. Example of images depicting the same object of interest (water lilly) with very different global color distributions

The test database for this domain consists of images of flowers (and some fruits) from various sources. These include images submitted as part of flower patents, scanned photographs of flowers, images from CDROM collections and images downloaded from the world wide web.

1.2.3 Domain III: Bird images

The final scenario under investigation in this thesis is the case where the object of interest is known to be prominent and the focus of the images in the database, but no subject-specific domain knowledge is available about the object of interest. In this case, general purpose retrieval strategies will produce reasonable results since the object of interest is prominent, but the results will be affected by the background since background elements are part of the image description. For example, Figure 1.3 shows the top five images retrieved by a global color-based retrieval engine when a user provides the first image in the panel as query, expecting to find other images of the black bird. It is clear from the results that the background played an important part in the retrieval since all the images have a backdrop of water, and some of the retrieved birds are not black. In this scenario, our goal is to automatically extract the object of interest from the background, so that the image description is based on the object of interest only. We use commonly observed facts about photographs of objects in general, no useful domain knowledge about the specific subject is assumed.

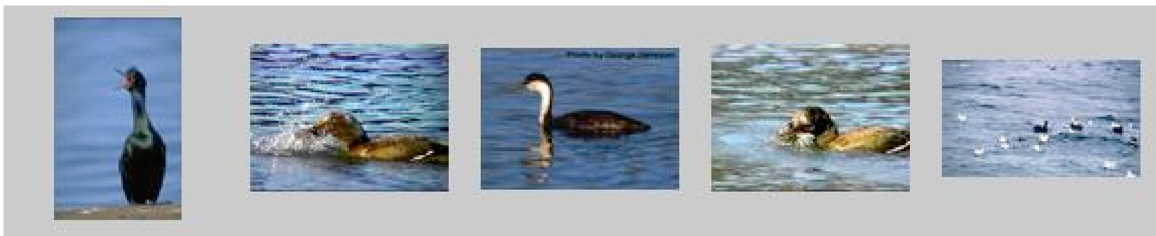


Figure 1.3. Example of image retrieval based on global color distribution in a database of bird images

The test database used for this domain consists of bird images downloaded from the world wide web.

Table 1.1 provides an overview of the features of the test domains. The test domains cover important segments in the space of all image categories. The first domain (advertisements) is highly unconstrained, with wide variations in background and scale of object. The second domain (flowers) has some limitations on the variations in background and scale, in addition to having useful domain knowledge available about the subject of the images. The third domain (birds) assumes a prominent object of interest with constraints developed from aesthetic considerations in photography.

1.3 Contributions of the thesis

In terms of the object of interest, the three domains that are being studied can be described as :

Database / Features	Advertisements	Flowers	Birds
Object color	Multi–colored (>=3 colors) Saturated colors	Few prominent colors (1–2 colors) Some colors unlikely	Few colors (1–3) Includes neutral colors
Object size	Very large variations	Moderate variations May occupy whole image (close–up shot)	Large Object usually most prominent
Background	Diverse Interfering colors Often more prominent than object	Simple backgrounds Does not overshadow object	Natural backgrounds
Object location	Unknown Could be anywhere in the image	Centrally located	Centrally located

Table 1.1. Characteristics of databases used in this work

1. *Domain I (Advertisement images) :* The object of interest is distinctively colored, but is present in a wide variety of sizes and locations in the target images, embedded in a large amount of complex, interfering backgrounds.

2. *Domain II (Flower images) :* The object of interest has some known properties and the background is of limited complexity, but there is variation in the size and location of the object of interest.

3. *Domain III (Bird images) :* The object of interest occupies a prominent area in the target images and is clearly the focus of the image, but the color of the object is not always distinctive enough to distinguish the object from the background.

In the first case, our proposed solution concentrates on filtering out the information from the irrelevant background elements after a query is posed, since the object of interest cannot be segmented a priori. In the second case, we propose methods for automatic segmentation of the object of interest (flowers) using the specific domain knowledge available about flowers. The automatic segmentation algorithm is generalized to the case where specific information is not available about the subject of the images for the third domain.

1.3.1 Domain I : Advertisement images

The first problem that this work investigates involves retrieval of multi-colored objects in the presence of large variations in the scale of the object in the target image and the presence of large amounts of interfering backgrounds in the target images.

We describe a new multi-phase, color-based image retrieval system (FOCUS) which is capable of identifying multi-colored query objects under adverse conditions (extreme variation in scale and background). The color features used to describe an image have been developed based on the need for speed in matching and ease of computation on complex images while maintaining the scale and rotation invariance properties. The first phase matches the color content of an image computed as the peaks in the color histogram of the image, with the query object colors. The second phase matches the spatial relationships between color regions in the image with the query using a spatial proximity graph (SPG) structure designed for the purpose. Generating histograms in local cells, combined with the use only of peak locations provides a reliable color description of complex images. The spatial proximity graph structure proposed is simple enough to be easily generated for complex images and yet captures color ajacency information that can be used to reduce false positives. The paper also proposes an effective two-phase strategy for matching where information computed after the first phase is exploited during the second phase computations to make the process computationally feasible.

The novel aspects of this system include:

- Two scale and rotation invariant color features have been developed which provide a good description of the color and color relationships present in a multi-colored object, even when the object is embedded in interfering backgrounds.

- A method is proposed for filtering the elements of the color description which were generated by background objects, once a query determining the object of interest is posed. This makes the matching process robust in the presence of background colors, reduces computation and makes the matching phase tractable.

- The matching process is split into two phases based on the requirement for speed. The first phase produces a very fast listing of possible candidate images, while the second phase can be used to eliminate some false matches if additional time is available. The overall retrieval is fast enough for an online interface, even when the images are very complex.

1.3.2 Domain II: Flower images

In the second problem, we investigate the complementary scenario where there is limited background complexity and domain knowledge is available about the object of interest. In this scenario, we develop a framework for automatic segmentation of the object of interest (flowers) from the background, before indexing the image by the color of the object.

We have developed an iterative segmentation algorithm which uses the available color and spatial 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 main contributions made while solving this problem include:

- Color and spatial domain knowledge are used for eliminating potential background elements.
- A natural language color classification is used to provide perceptually correct retrieval and for interpreting natural language domain knowledge.
- An automatic iterative segmentation algorithm with domain knowledge-driven feedback is proposed for isolating the object of interest from the background.

1.3.3 Domain III: Bird images

In the third problem, the framework for automatic segmentation of the object of interest developed for the flower database is extended to the case where no significant domain knowledge is available except for some commonly true non-domain-specific facts about many photographs of objects.

The aim of this work is to index images in domain specific databases using colors computed from the object of interest only, instead of the whole image. The main problem in this task is the segmentation of the region of interest from the background. Viewing segmentation as a figure/ground segregation problem leads to a new approach - eliminating the background leaves the figure or object of interest. To find possible object colors, we first find background colors and eliminate them. We then use an edge image at an appropriate scale to eliminate those parts of the image which are not in focus and do not contain contain significant structures. The edge information is combined with the color-based background elimination to produce object (figure) regions.

The solution to the problem of automatic detection of the main object in an image incorporates:

- An automatic figure/ground segregation algorithm based on elimination of potential background regions.
- Fusion of color and edge information to produce a final region of interest.

1.4 Organization of thesis

The thesis is organized such that each domain is covered in a different chapter. In addition, chapter 2 gives an overview of related work in the area of image retrieval, and the last chapter is devoted to conclusions and possible future extensions to the current thesis.

This thesis proposes novel ways to improve the performance of image retrieval based on the characteristics of the image database for which the retrieval strategy is designed. Chapter 3 proposes a solution to the problem of the presence of interfering backgrounds and large scale variations, which plagues most existing image retrieval systems. In chapter 4, improvements over existing image retrieval systems are proposed by exploiting the special features of the database to extract the region of interest. Chapter 5 shows possible improvements by using an automatic figure/background segmentation before extracting features from the image. Finally, chapter 6 concludes with a summary of the dissertation work and potential future work.

CHAPTER 2

LITERATURE SURVEY

This thesis aims to develop color-based image retrieval engines for specialized databases, where it may be possible to segment or distinguish the object of interest from the background. In this context, work in the areas of image retrieval, color representation and image segmentation are relevant to this work. We will discuss work in these three areas separately in the next sections.

2.1 Image retrieval

Image retrieval has been an active area of research since the early '90s. The initial focus in this area was to develop suitable low-level features to describe the semantic content of images, analogous to words in language. Color, texture, shape and filter response-based features have been used as attributes for indexing images for content-based retrieval. A recent survey of techniques used in content-based image retrieval [69] provides a good overview of the approaches that have been investigated over the last ten years. Recent papers have focussed on building image retrieval systems which use combinations of features and address actual applications like searching for images on the world wide web. A survey of content-based image retrieval systems [62] lists many such end-to-end systems, many of which are available for trial online. 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.

2.1.1 Color features

Color is a commonly used low-level feature when the database images are in color. It is useful for indexing objects which have distinctive colors signatures, for example, commercial products, flags, postal stamps, birds, fishes and flowers, or as a first pass for other colored images. Swain and Ballard [73] 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 [27] 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. Matas et al [41] 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. An efficient indexing strategy using a hybrid graph representation of color adjacencies in an image is proposed by Park et al [53]. Spatial color distribution information has been used for indexing trademark images in [31]. The Photobook [56] system uses principal component analysis on color which maintains spatial adjacencies. Gevers and Smeulders [20] describe a color-based retrieval strategy where the colors inside and outside curvature maximums in color edges are used to identify objects.

The color recorded in a digital image varies considerably with the orientation of the object surfaces with respect to the camera, the position and spectrum of the illuminant and other factors discussed in the introductory chapter. Human perception of color is also an important factor since the end user of most image retrieval systems is a human. The color representation selected often has an important impact on the ability of the retrieval system to deal with variability of color in the database images and providing the user with results which appear to be perceptually correct. There has been work on perceptual organization of the color space in the area of image indexing [75] and in color science [84]. Funt and Finlayson [19] have proposed an illumination invariant color representation for color image indexing. In specialized domains, color domain knowledge has been mapped to the 3D color space in applications like face identification using skin tones [8] 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 [63]. Regions corresponding to a specified color model are detected in [23]. Models of the appearance of colors under known viewing conditions has been studied in [6].

2.1.2 Other features

There are many examples of subjects where color is not a relevant feature e.g. industrial parts, cars, buildings, people etc. Also, the database images may not be in color, requiring features independent of color for indexing.

Two-dimensional shape is an important feature for distinguishing objects in some domains like cars, houses and machine parts. Considerable work has been done in the area of pattern recognition, on matching such shapes to each other. For example, Mehtre et al [43] 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 [42] and invariant moments [7]. 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. Since 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 using shape features have used manual segmentation [49] 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 [37] have used texture-based patterns for image retrieval. Liu and Picard [35] have proposed an image model based on the Wold decomposition of homogeneous random fields into three mutually orthogonal sub-fields 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 [59, 58]) which describes the intensity surface, eigen features [74] and signatures extracted from the Fourier power spectrum of images [46].

2.1.3 Systems using a combination of features

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 [48, 57]. Different combinations of features have been used depending on their appropriateness for the test database. Jain and Vailaya [29] have used color histograms and shape as features to index a database of trademark images. The shape is described as a histogram by taking counts of the different edge directions present in the image. Belongie et al [4] use color and texture features for content-based retrieval.

For retrieval systems that work with general databases like generic stock photographs and mixed news photographs, 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* [49] 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* [3] 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* [56] 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. The *SIMPLIcity* system described in [79] uses a wavelet-based approach for feature extraction, combined with integrated region matching. The regions in the image are characterized by their color, texture, shape and location. Other examples of existing systems using multiple features and multiple query modes are *Candid* [32] and *Chabot* [50].

An emerging problem in general image search is to retrieve relevant images from the World Wide Web. The *PicToSeek* [21] image search engine for the web uses a unified high-dimensional feature set combining color and shape information for indexing images. Smith and Chang [70] 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 [65] have developed the *ImageRover* system to gather images from the web and index them using color, texture, orientation and other specialized features. The Viper system [85] can search World Wide Web images using text, colors, wavelet features or shape. The user needs to provide the weightage given to each kind of feature. 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. Nastar et al [48] 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 evalution criteria are.

2.1.4 User interaction

Since most image retrieval systems are aimed at human users, the retrieval results can only be evaluated by extensive user studies. Since user judgements are often subjective, it is often hard to design an automatic system which a user will find satisfactory. The alternative is to keep the user in the loop during the retrieval process.

The most common query format used in image retrieval systems 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! The PicHunter [9] system models the uncertainty about the user's goal by a probability distribution over possible goals. Assuming that the user has a desired goal, *PicHunter* uses Bayes's rule to predict the goal image, by maintaining an explicit model of a user's actions. Another approach to specifying the object of interest has been to allow sub-images as queries where the user marks the area of interest [13, 58]. However, this may not still 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. More recent systems like *Surfimage* [48] provide relevance feedback as a mechanism for refining the retrieval results interactively using input from the user.

2.1.5 Retrieval in specialized domains

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 [55] describe the eigenimage 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 [59, 60] 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 [14] 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 [17] 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 [15] 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 [16].

2.2 Image segmentation

Though the focus of this thesis is on image retrieval, some work in the area of color image segmentation is relevant since the thesis proposes automatic segmentation of the object of interest for specialized domains. Image segmentation is a relatively old field of research in image processing and encompasses a vast body of literature. A review of image segmentation techniques are available in a number of survey papers on the topic [87, 86, 52].

Color histograms have been used in different forms in a lot of work in the area of color image segmentation. Beveridge et al [5] have used localized histograms followed by region merging to segment an image. Clustering entries in the 3D color histogram in different color spaces followed by back-projection to the image has been used to generate image segments in [81, 83]. A non-parametric clustering of the histogram for image segmentation has been proposed in [33]. Pauwels and Frederix [54] have identified image pixels originating from one uniformly colored object in an image using a nonparametric clustering algorithm in RGB-space. Multiresolution color image segmentation is described in [36]. Recent work has focused on combination of different cues like color, texture and edges for segmentation [40, 39]. Belongie et al [4] 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". Relational graph matching has been used for segmenting natural images in [67]. However, all the techniques mentioned above segment an image into regions satisfying some similarity and smoothness criteria. They do not identify objects of interest in the image and cannot distinguish the background elements from the foreground elements.

There have been some attempts to provide higher level grouping of segments into objects. Fuh et al [18] describe a hierarchical relationship between regions to describe objects in an image. Region adjacency graphs have been used in [76] to enhance image segmentation for pattern images with distinct boundaries. Serafim [66] describes the extraction of colored object surfaces in constrained situations. Shi and Malik [68] have posed image segmentation as a graph partitioning problem which generates perceptual groups in the image as output. These are likely to correspond to objects of interest in the image. Pauwels and Frederix [54] have identified image pixels originating from one uniformly colored object in an image using a nonparametric clustering algorithm in RGB-space. Automatic foreground/background disambiguation of image segments based on multiple features like color, intensity and edge information has been described in [28], assumimg relatively smooth backgrounds and objects with sufficient contrast. Recently proposed techniques for detecting natural shapes in real images [38, 82] also work best with simple backgrounds. The QBIC image retrieval system [49] uses some semi-automatic techniques for segmentation of the object of interest [1].

CHAPTER 3

INDEXING IMAGES WITH MULTI-COLORED OBJECTS

3.1 Introduction

When the database has images of multi-colored objects which can be recognized on the basis of their distinctive color signatures alone, the color of the object is an obvious choice for indexing. Most of the existing retrieval systems which use color [3, 49] assume that the target images to be retrieved are those in which the query object occupies the most prominent part. As a result, large variations in the scale of the object and the presence of signicant background in the database images cause problems for these systems. Some methods use features in addition to color histograms [49], but these require manual annotation by the user during offline processing. The histogram cluster-based matching described by Kankanhalli *et al* [30] is scale invariant but does not handle the presence of interfering backgound colors in an image. Thus, even though color has been recognized as an important tool in content-based retrieval, fast color-based retrieval strategies which can handle interfering backgrounds and large variations in scale are not yet available.

The problem of finding global similarity between a query image and candidate images based on color has been addressed by a number of existing retrieval systems [72, 49, 30, 22]. However, global similarity-based retrieval ignores the presence of background and the possibility that the object occupies a small portion of the image. Since multi-colored objects usually do not occur by themselves in images, we make no assumptions in our solution about the presence or absence of background objects, or the prominence of the query object in the database images in which it is present.

The queried object may be embedded in images which have nothing else in common apart from the presence of the queried object (Figure 3.1).

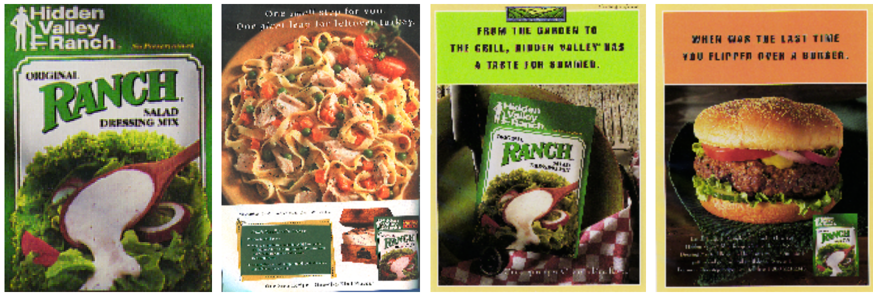


Figure 3.1. Example of a query image (left) and correctly retrieved images

There are many image domains where object color can be a basis for retrieval - flags, logos, consumer products, textile patterns and postal stamps among man-made objects and flowers, birds, fish and butterflies as example image databases in the natural domain. To demonstrate the techniques of this chapter, we have selected a test database of commercial advertisements on which we evaluate our retrieval engine. The task of retrieving all advertisements featuring a given product is particularly complex, since the queried object may appear in candidate images in various sizes and orientations with a wide variety of background colors and forms as shown in Figure 3.1. Unlike other databases on which color-based retrieval has been tried (flags, logos, products), in most of the advertisement images the products do not spatially dominate the image, nor are they necessarily in the center. There is no concept of foreground and backgound - what is backgound clutter for this application may actually be the foreground of the image. Consequently, no focus-of-attention pre-segmentation is possible. In the rest of the chapter, background refers to all objects and context in the image which are not a part of the queried object.

Our choice of domain gives us some advantages in offsetting the difficulty of the problem. Advertisers want consumers to see their product clearly, so occlusion of the product is rare, and typically the same aspect of the product is presented in all advertisements. There may be small out-of-plane rotations causing some occlusion, but all major colors of the object remain visible. Also, the advertisers take care that their products are printed with their true colors so there is little color distortion across different advertisements of the same product. However, variations in color still arise because light and shadow effects in images create differences in the apparent color of objects. Fortunately, the color variations are not severe and can be handled by the selection of a robust color representation and by allowing some tolerance in the matching strategy.

3.2 Our Approach

There are many different philosophies on what an ideal image retrieval system should offer. Some researchers believe in allowing the user to visualize and manipulate the low-level feature space to get the desired retrieval results. At the other end of the spectrum, a very naive user is assumed and effort is directed at hiding the internal working of the retrieval engine from the user. While some systems concentrate on speed of response, others strive to produce accurate results, often at the expense of speed. The amount of processing deemed acceptable for the offline index generation phase also differ. Some systems are extensively manually annotated, while others may compute low-level features from low resolution images to quickly index large image databases.

Our goal is to provide a retrieval engine suitable for an online user interface i.e. the user waits while the results are being generated. This requires a fast response from the system, of the order of magnitude of a few seconds. We balance the trade-off between speed and accuracy such that the initial response is very fast, but there is an option to require higher accuracy at the cost of additional processing, which is also fast enough for online interfaces. We do not expose the user to low-level features, but we expect them to be able to provide a focus on their object of interest by marking it in an image. Our offline processing is fully automated, and is computationally lightweight.

The speed and accuracy of a retrieval system will depend on the features used to describe the images and the matching strategy used. The main requirement for the color characteristics selected for matching is to provide discrimination between images which contain objects similar to the query object and those which do not. Here again, systems make different assumptions about the database images. Some of the more restrictive conditions that are frequently assumed are that the query object occupies most of the area in target database images, there is no significant background in the images or the query and target object are of the same size. Our goal is to provide retrieval in the scenario where there is large variation in scale between query and target objects and there is a lot of interfering background clutter in the target images. Thus, the feature(s) matched needs to be invariant to differences of the query object in scale, location and orientation in the candidate image and the presence of background colors in the candidate image. In order to provide fast matching, it is also desirable for the color characteristics to be indexable.

In this chapter, we describe the FOCUS (Fast Object-Color based qUery System) retrieval engine which meets these goals. We develop two scale- and orientation-invariant color features, combining them in a two-phase matching strategy to achieve fast and accurate retrieval. The emphasis in the first phase of matching is on speed of retrieval, and the second phase aims at removal of false matches from the image list produced by the first phase.

During the first phase, database images which have all the colors of the query image present in them are extracted using an index structure computed offline. During the second phase, evidence supporting the hypothesis that a candidate image contains the query object is generated by detecting the specific local spatial color relationships of the query object in the image. This is achieved without involving any slow pixel

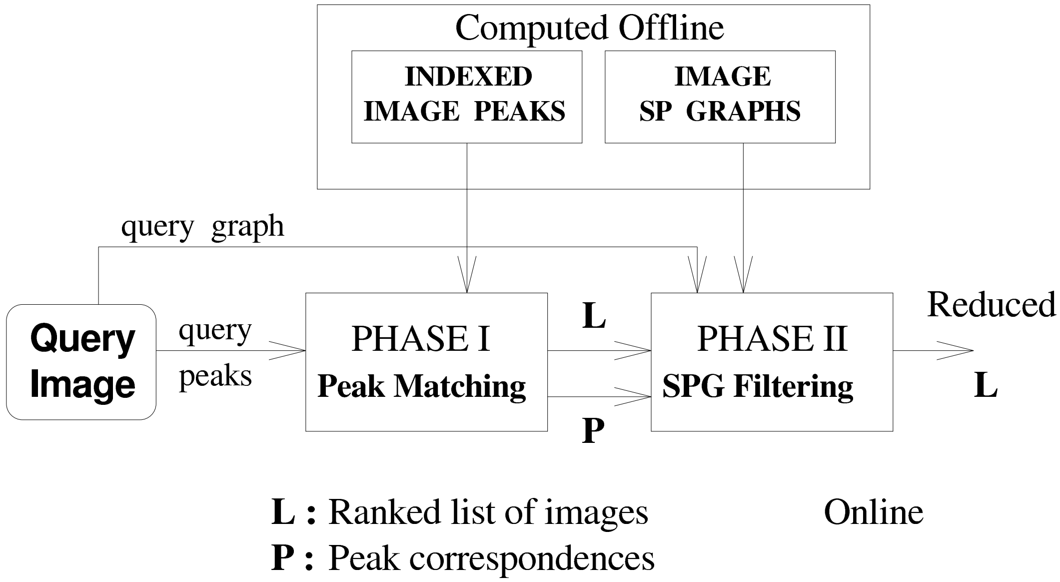


Figure 3.2. System overview of FOCUS

level processing of the image by using a graph description of the image described in a later section. The second phase acts as a filter, deleting images from the list obtained by the first phase if no evidence of the query color relationships is found in them. Figure 3.2 shows an overview of the FOCUS system. The retrieval results can be examined by the user after the first phase and a decision made on whether the second phase of processing is needed depending on user estimation of the number of false matches produced.

This chapter is arranged such that each section describes a component of the system. Sections 3.3 and 3.4 describe the first and second phase of matching, and section 3.5 describes query processing. Section 3.6 presents experimental results, followed by the conclusion.

3.3 Phase I: Peak Matching

The first phase of matching is intended to produce a candidate image list as quickly as possible. However, histogram-based matching, which is very fast and is the most commonly used method for matching color, is unsuitable for our problem. This is due to the fact that differences between the query size and the candidate object size and the presence of background colors will cause mismatches in histograms even when the query object is present in the candidate image. Instead, we propose *histogram peak* matching for retrieval of images from the database. This method is based on the observation that *each prominent color in an image corresponds to a distinct peak in the color histogram* of the image. Using only the peaks in the histogram is sufficient when the queried objects are multi-colored and have distinct, prominent colors. This is true of many artificial objects like consumer products, fabrics, company logos etc. In the natural domain, this is true of birds, insects, flowers, minerals etc. Matching histogram peaks ensures that all the major colors in a a multi-colored query object are present in the candidate image. This is the simplest requirement for a database image to be a potential match, and is therefore, suitable for the fast generation of an initial candidate image list.

Histogram peak-based matching addresses the problems encountered with the full histogram-based color matching. Since only the *location* of the peak in color space is used and not its height (i.e. the *number of pixels* of that color), this method is unaffected by the presence of other objects in the image even if they are of the same color as the object. The peaks detected are also independent of the size of the object in the image. Justifications for the proposed method are also available from other standpoints. Stricker [71] shows that indexing by color histograms works well only if the histograms are sparse, i.e. most of the database image histograms have only a few non-zero bins. Since representing an image by its histogram peaks is equivalent to the case where very few bins in the histogram have significantly large counts, this measure is a good choice for indexing. The storage requirement is reduced from the full histogram for each image to just the peak locations which is two orders of magnitude less than the full histogram. The locations of peaks in a histogram are stable under viewpoint change and scale transformation, unlike histogram bin counts.

This section covers the construction of color histograms, histogram peak detection and its use in the first phase of the FOCUS retrieval system.

3.3.1 Color histogram construction

The two important parameters which govern the construction of a color histogram are the choice of the *color space* and the histogram *bin size*. We need color peaks to be stable over some variation in lighting in the image and to be localized enough to distinguish nearby colors. The *color constancy* problem, which is the difference in perceived color under varying lighting conditions, affects all color-based retrieval methods. The illumination model required to solve this problem is rarely available for any database image. The choice of the three-dimensional color space affects the sensitivity of the peak to differences in illumination. Since the three axes of the color space may have different sensitivities to lighting variation, different bin sizes may need to be selected along each axis. The choice of bin sizes should also be such that fine discrimination between perceptually different colors is possible. Experiments were run on test patches of perceptually single colors taken from the test database which show variations in illumination to determine a suitable color space and the level of discretization required for this application. Figure 3.3 shows a representative set of 20 color patches taken from single color regions (where white is not considered to be a color) in the images.

Color images are generally stored in the *RGB* (Red, Green, Blue) space with 256 levels (8-bit) in each color. Since there is no distinction between the properties of the R, G and B axes of this color space, the same bin size is used along all three axes. With a 16-bin discretization along each axis, the RGB values vary widely for the same color when there is variation in the lighting across the color patch as seen from the large errorbars in Figure 3.4.

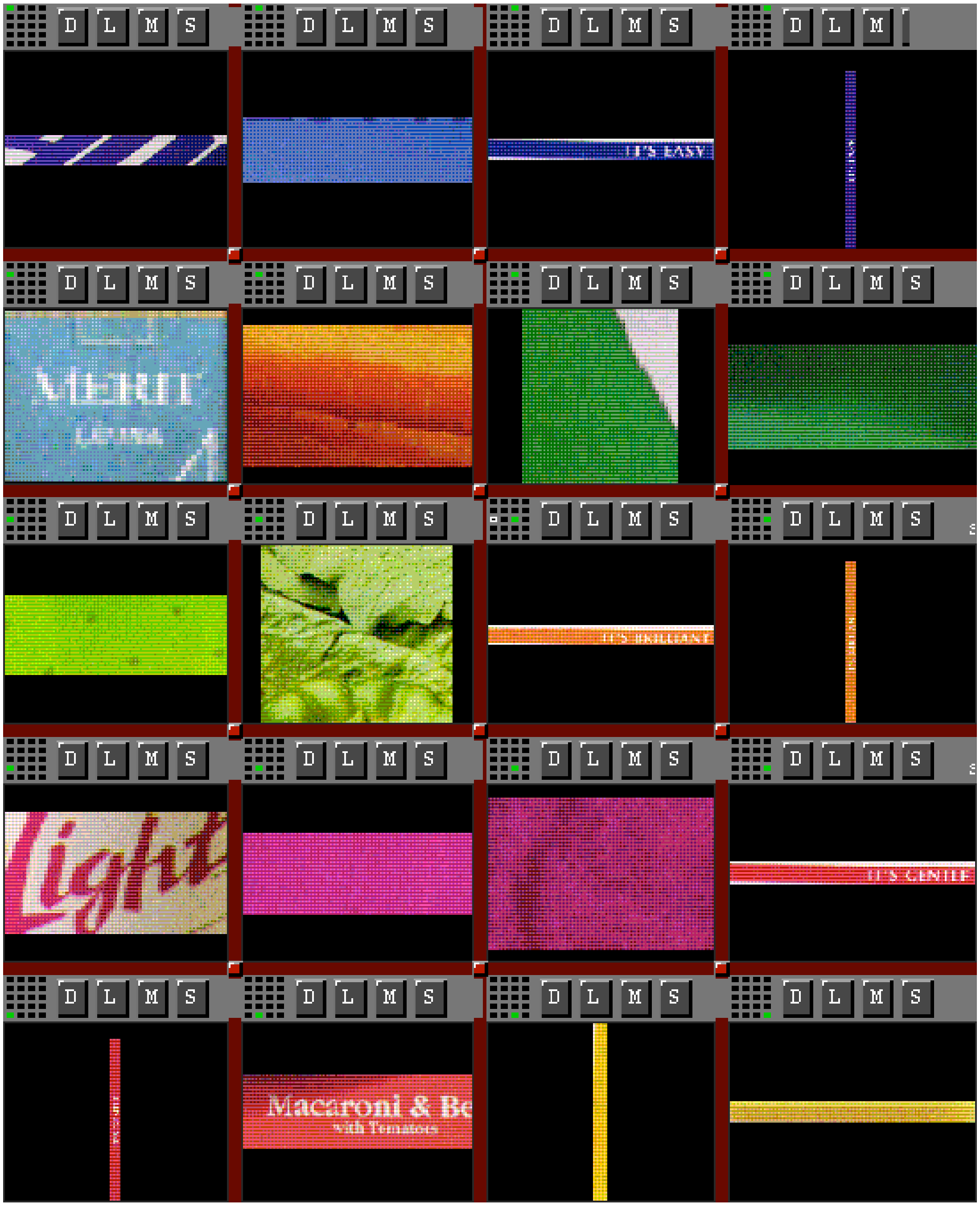


Figure 3.3. Test patches of single colors taken from images in the advertisement database

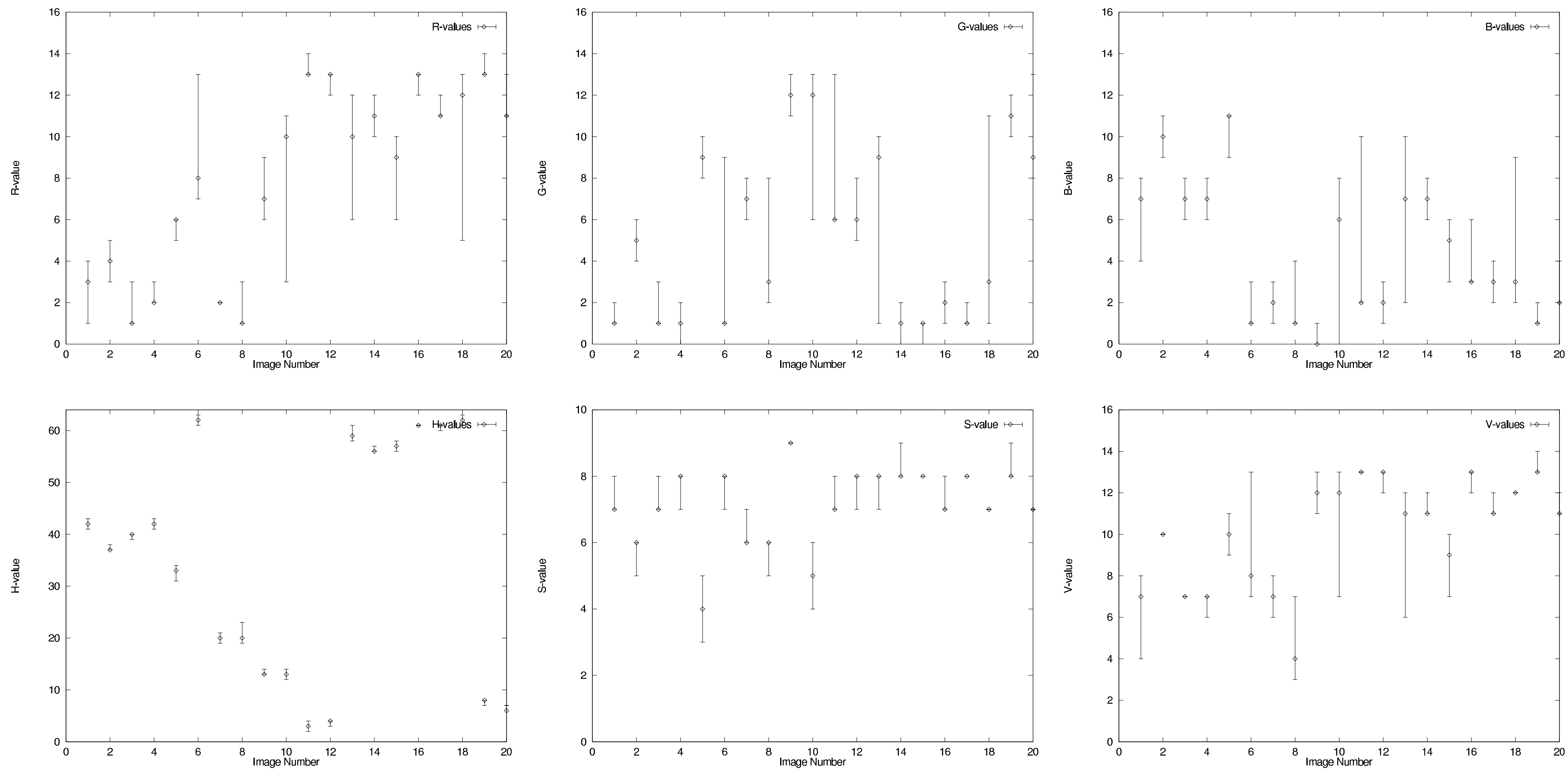


Figure 3.4. Errorbars about peak location for the test patches (top) in RGB space (bottom) in HSV space

There are various other three-dimensional color spaces which can be used. We selected the *HSV* (Hue, Saturation, Value) color space for further investigation because the axes of the color space are meaningful from the point of view of perception of color. The H-value corresponds to the hue of the color, which we loosely call the color itself. The S-value or saturation corresponds to how deep the color is e.g. a particular hue of red could appear to be anywhere between pink and deep red depending on the saturation. The V-value corresponds to the intensity or brightness of the color. Theoretically, only the V-value should be affected by changes in illumination levels. However, in practice, saturation may also be affected by changes in illumination due to the formation of shadows and highlights. Both these components of color are also affected by differences in the quality of printing in the advertisements and the scanning process used. Since hue is expected to be most stable, we can use more bins along the H-axis for fine discrimination between colors than the other two axes. Figure 3.4 shows the peaks obtained with the test patches using the *HSV* color space. The hue axis has 64 bins, the saturation axis has 10 bins and the value axis has 16 bins. The hue component of the histogram peak is seen to be very stable and sharp even at fine resolution. The saturation component is stable to within one bin. The value component is most affected by the changes in illumination as expected. These observations are used in the matching phase by allowing more variation in value than in the hue and saturation components when matching peaks.

The *HSV* space with a discretization of 64x10x16 bins along the H,S, and V axis respectively is selected for use in this work based on its characteristics observed by experimentation. Figure 3.5 shows the hue histogram constructed from the query template shown on the left. There are four peaks in the histogram corresponding to the four colors present in the template. Only a one dimensional histogram is shown for ease in visualization, yet the peak structure is apparent with just the hue component.

It should be noted that the selection of color space is not unique i.e. it is likely that other perceptually based color spaces (e.g. CIE L*a*b*, L*u*v*) [84] will produce relatively stable peaks as well. Evidence for this exists in the interchangeable use of the L*a*b*, L*u*v* and HSV color spaces in literature on color-based image retrieval. There is also an one-to-one correspondence between the perceptual meanings of the axes in these color spaces.

An issue that needs to be addressed when using the HSV color space is the problem with classification of "grey" pixels in the image. These are pixels with nearly equal red, blue and green components and appear as shades of grey from black to pure white. In the HSV space, these pixels map to arbitrary hue locations depending on the component that is slightly larger than the others. These pixels add noise to the color histogram and can obscure the peak structure since they are present in large numbers in most advertisement images. However, they can be easily identified by their very low saturation component and are not counted during histogram construction. We do not lose any valuable information due to this because white and black pixels are present in most images and therefore, these colors do not provide any discriminatory power between images.

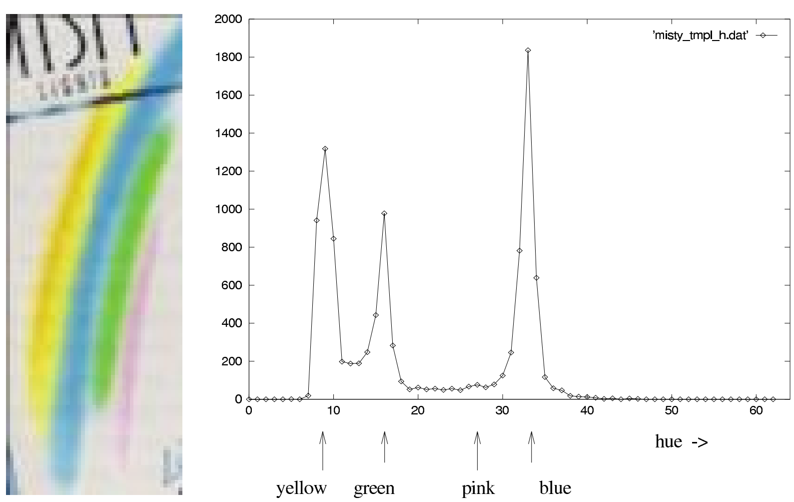


Figure 3.5. Query template and its histogram along hue axis with peak locations labeled

3.3.2 Detection of histogram peaks

For images with distinct regions of single, chromatically pure colors e.g. flags, commercial products etc., there is a sharp peak corresponding to each color region in the color histogram. However, images containing natural scenes and people, produce wide peaks in the histogram. So even when the query object has distinct colors locally in the database image, the peaks corresponding to the query colors may be masked or shifted by the wide peaks from background colors. For example, Figure 3.6(b) shows the global hue histogram of an image of a "Ziploc" bag along with various vegetables which are different shades of green and yellow. The histogram in Figure 3.6(c) of an area which covers the "Ziploc" package only, shows the actual peak locations of the colors present on the package. These peaks are lost in the global histogram, being subsumed by the colors in the background. This example also suggests a solution to the problem. The color peaks present in an image can be determined more accurately when the histogram covers a small area of the image, reducing the effect of the presence of interfering colors.

We use a *split and merge* strategy for peak detection for handling interfering background colors. Since we do not know the size or the location of the object of interest in the image a priori, the image is divided uniformly into m x n non-intersecting cells as shown in Figure 3.6(a). Local histograms are constructed in the image cells and peaks are detected in the local histograms. A combined list of peaks is produced by merging multiple copies of the same peak, and a *peak color label* is assigned to each peak which is unique for that image. Splitting the image localizes the color peaks in a cell of the image and this information is used during the next phase of matching. The *split* step also reduces the area of the image covered by the histogram to a small locality and thus reduces the number of colors present in each cell and the chance of interference between colors. The peaks obtained by this method describe the various colors present in the image more accurately than global histogram peaks. Localized processing of images has been shown to improve image segmentation in earlier work by Beveridge [5], Nagin [47] and Ohlander [51].

During offline processing, peaks are detected for all the images in the database. The histogram peaks are detected by finding *local maxima* in a 3-D neighborhood window. Using local maxima results in a larger number of peaks when compared to global methods like histogram clustering described in the literature [30]. However, in that case, a cluster mean may not be representative of the whole cluster distribution for accurate matching, since smaller peaks close to larger ones (e.g. yellow and green in Figure 3.6(b)) may be merged into a single cluster. Since the query object colors are not known a priori, we argue that it is necessary to have peaks representing every distinct color present in the database images.

3.3.3 Indexing color peaks

The aim of indexing is to narrow the search to include only the images which could match the given query peaks. When looking for color similarity, it is not sufficient to look for exact matches since there is color variation with illumination and other factors like mode of acquisition. Approximate matching is necessary so that the matching process is robust to small variations in color and degrades gracefully. Thus, given a query peak P_q (h_q, s_q, v_q), we need to find all images which contain a peak in the *neighborhood* of P_q. This requires an order-preserving indexing structure which supports range queries of the form $(h_q \pm d_h, s_q \pm d_s, v_q \pm d_v)$ where d_h, d_s and d_v specify the spread around the hue, saturation and value components of the query. We have used the standard $B+$ tree described in most database systems textbooks [34] to store the peaks in the database. The peaks are sorted with hue as the primary key followed by saturation and then value.

A $B+$ tree is the most common implementation of B-trees which is used by most commercial relational database systems. It features fast random and sequential access

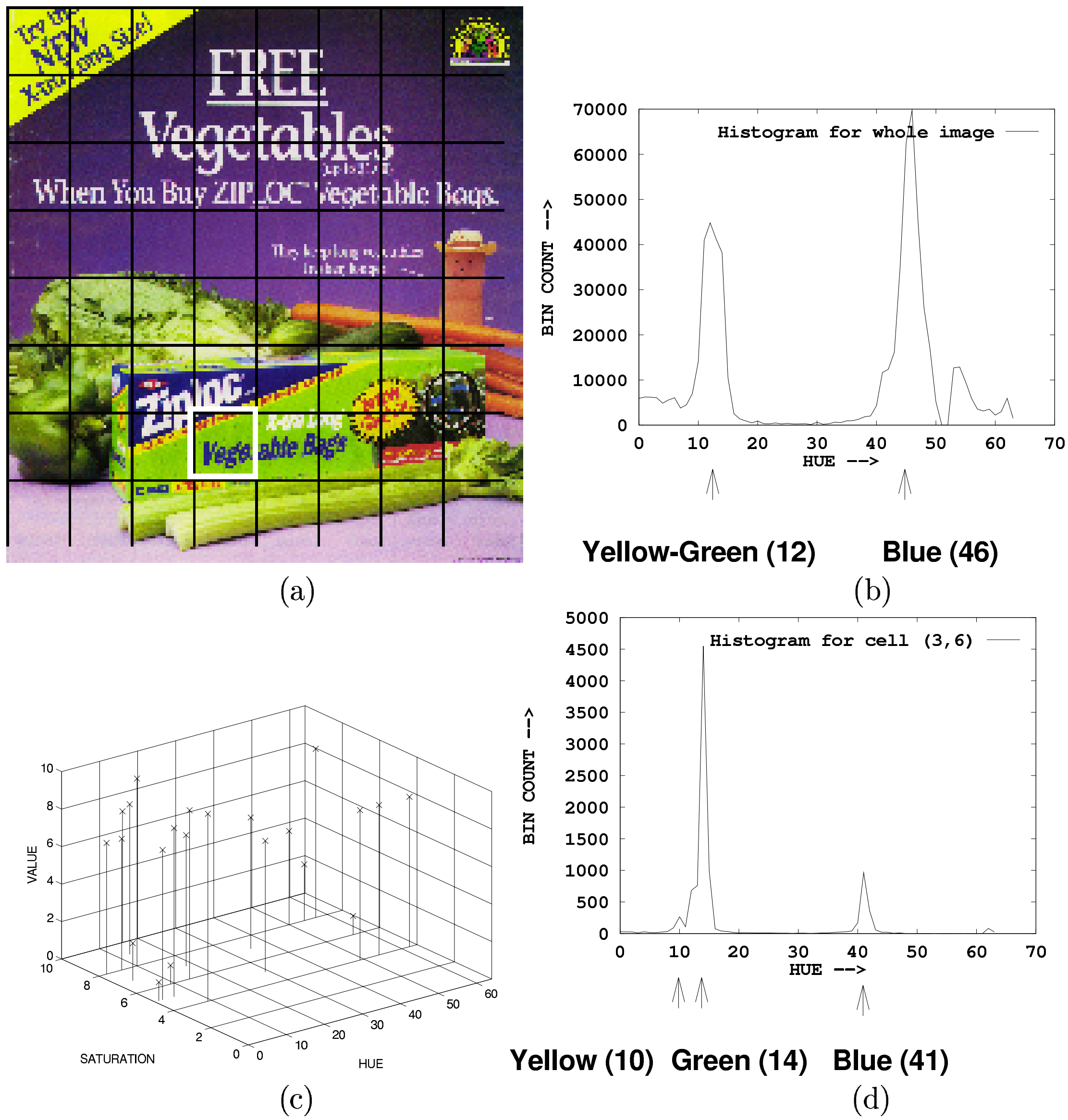
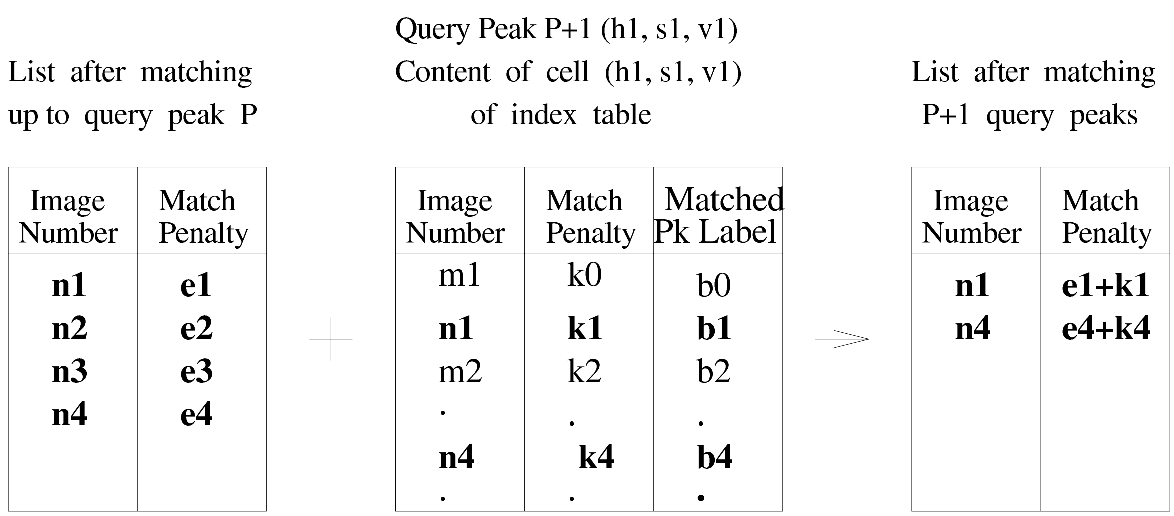


Figure 3.6. Effect of interfering background on histogram peak location: (a) Original image (b) global hue histogram (c) hue histogram of cell marked in white (d) final peaks in discretized HSV space

and dynamically maintains a balanced structure. Each non-leaf node in the tree consists of pointers to sub-trees and key values which indicate which sub-trees to search (smaller key values are obtained by taking the tree pointer to the left of the key, and higher values to the right of the key). So, range queries are easily supported. The R-tree indexing structure described in [2] could also be used in this case and has similar properties. In addition to the peak index, a *frequency table* is also constructed which gives the number of images which will be retrieved for each point in the discretized HSV space.

3.3.4 Matching query peaks

During online processing, query peaks are computed in a process described in a later section. We assume that the query image selected by the user has no background, so all query peaks need to be matched. The query peaks are ordered by increasing frequency of occurrance in the database by consulting the *frequency table* computed and stored during offline processing. The justification for this ordering is based on the well known fact in text-based information retrieval that the frequency of occurrance of terms is inversely proportional to their discriminatory power. So the rarest query peak is used to produce the first list of images.



Correspondences noted : (P+1, b1) for image n1, (P+1, b4) for image n4

Figure 3.7. Actions performed during the *join* step of retrieval

For each peak in the query, a range query of $(h_q \pm 3, s_q \pm 4, v_q \pm 5)$ is executed starting with the peak which retrieves the minimum number of images onwards. The larger range along the saturation and value axes are needed to take into account variations in these components of the color description, and to include images with similar colors (inexact matches) in the retrieved list. The standard technique of a *join* [34] is taken to combine the lists of image identifiers to find the images which have peaks matching *all* query peaks (an example is shown in Figure 3.7). The *join* step performs three tasks :

- Finds the image identifiers common to both lists and retains only these identifiers in the joined list. In Figure 3.7, image numbers **n1** and **n4** are common to lists for both query peaks P and $P+1$. This process is fast because both lists are sorted by the image identifiers.

- Updates the mismatch scores for the retained images by *adding* the new mismatch scores with the existing scores. In Figure 3.7, the mismatch scores for **n1** and **n4** are updated to $e1 + k1$ and $e4 + k4$ respectively.

- Notes the correspondence between query peak and image peak label. In Figure 3.7, query peak $P+1$ matched the peak labelled $b1$ in image **n1** and $b4$ in image **n4**.

The utility of the second and third operations above will be explained in subsequent sections.

The time complexity of the retrieval process is given by $O(qlog(kN))$, where q is the number of query peaks, N is the total number of images in the database and k is the average number of peaks per image (which is 12 for the test database used in this study). The join process is linear in the size of the lists retrieved.

3.3.5 Producing a ranked image list

It is standard practice in information retrieval to order the retrieved list using some match criteria before being presented to the user so that more relevant documents (images, in this case) appear first. Ordering is imposed by computing a numerical score for each image retrieved, based on the degree of match with the query. In this case, a perfect match is detected when there is a candidate peak very close to (within a small tolerance window) around each of the query peaks. A tolerance window of $(h \pm 1, s \pm 2, v \pm 3)$ is empirically found to be suitable for the test database (Figure 3.4). The tolerance window is smaller along hue and larger along value, reflecting the different sensitivity of each component to variations in illumination. Any peak beyond the tolerance window produces an *inexact match* with some *mismatch score*. The mismatch score is computed as the *city block* distance between the candidate peak and the nearest perfect match. All mismatch scores greater than four are treated as *mismatches*. Therefore, for a candidate peak (H, S, V) and a query peak (h, s, v) the mismatch score is computed as $h' + s' + v'$ where

$h' = |H - h - 1|$ if $|H - h| > 1$, 0 otherwise,

$s' = |S - s - 2|$ if $|S - s| > 2$, 0 otherwise,

$v' = |V - v - 4|$ if $|V - v| > 4$ and 0 otherwise.

The final mismatch score is computed during the *join* phase of retrieval. The match penalty is cumulative and reflects the degree of match for all the peaks in the query. The final list of images is sorted in increasing order of match penalties. This ranked list is the output from the first phase.

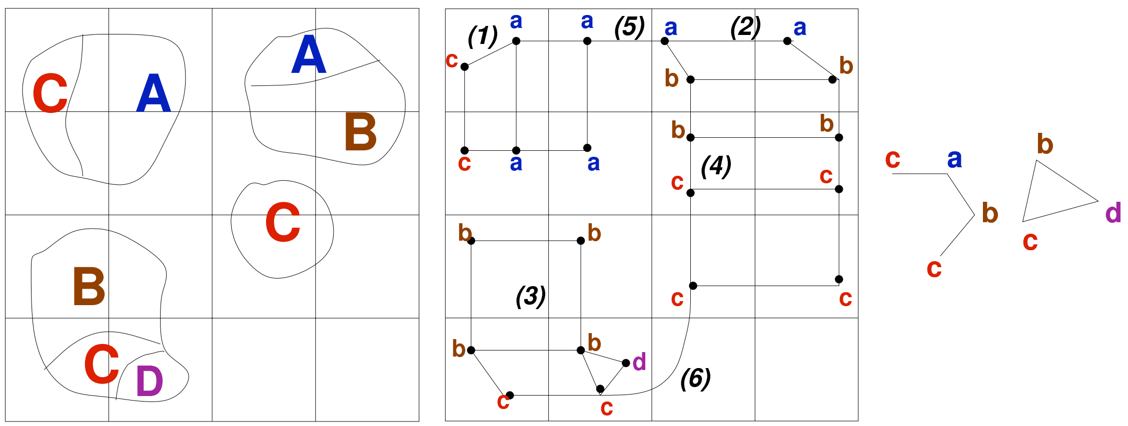
3.4 Phase II: Spatial Proximity Graph (SPG) matching

A ranked list of candidate images is obtained at the end of the first phase of matching. There will be a number of false matches in the image list retrieved by the first phase of matching in which all the colors of the query object are present, but not in the same spatial configuration as in the query object. In the extreme case, the matched colors could be scattered across the image and not form any connected cluster which could represent a single object. In other cases, some color adjacency relationship present in the query object may be violated in candidate images. For example, in the query in Figure 3.10(a), the red (peak color label 0) and blue (peak color label 3) regions are adjacent, whereas in the false match (Figure 3.10(b)) they are not adjacent. These false matches could be eliminated if information on spatial distribution of colors in the image was available.

The color adjacency graph (CAG) formulation used by Kittler et al [41] is a good descriptor of the color relationships in a multi-colored object, where the color regions are nodes in a graph with edges connecting color regions which are adjacent at the pixel level. However, a CAG description of the database images is not feasible for retrieval due to the complexity of the images. Most of the images contain natural objects and color regions in which there are no distinct boundaries between colors. For example, the images shown in Figure 3.1 are quite typical of the images in the database. An attempt to construct a CAG for these images would produce very large, complex graphs making the matching phase intractable. Even with simpler images, the estimated time of 20 seconds per match computation [41] is too slow for retrieval with online user interfaces, though it may be acceptable for object recognition applications. A matching strategy which is more computationally intensive than the first phase of retrieval can be applied at this stage since these operations need to be carried out only on the candidate images from the first phase and not the whole database. Even with this reduction, pixel level processing of the images (histogram backprojection, for example) would make this phase of matching too slow for online user interfaces. In response to this problem, we have developed a new graph description of the spatial relationship between color regions which is efficient to compute and match.

3.4.1 Construction of the SPG

Our aim is to produce a description of the spatial relationships between color regions in the image while avoiding the pitfalls of earlier graph-based color matching strategies i.e. creation of graphs which are computationally expensive to construct and match. We propose a new graph which is constructed from information created during the peak detection process, without additional pixel-level processing. Efficiency during matching is achieved by the effective use of information generated while matching peaks during the first phase of retrieval. The graph captures all possible pixel-level adjacencies present in an image, but is not exact, including some false edges as well.



(a) (b) (c)

Figure 3.8. Example of spatial proximity graph (SPG) construction (a) Synthetic image divided into cells (b) Cells marked with nodes (peaks) contained in them. The intermediate graph is shown in broken lines (c) SPG constructed from (b)

We start by constructing an intermediate graph representation directly from the peak description of the image based on whether pixel level adjacency is *possible* between two color regions. Figure 3.8 is used to explain how the spatial relationships between color regions can be inferred from the color peak description and condensed into a compact graph - the *spatial proximity graph* (SPG). Figure 3.8(a) shows a synthetic image of objects with four color regions (A,B,C,D) producing peaks which are labelled (a,b,c,d) respectively. The peaks detected in each cell are shown in Figure 3.8(b) by including the peak color label within the cell. These peaks form the nodes in the intermediate SPG. The edges in the intermediate SPG indicate that the two peaks could be from adjacent color regions in the original image. The edges marked are generated from the following observations.

- When two nodes occur in the same cell, they could be from adjacent color regions in the original image, so they are connected by an edge, e.g. edge labelled (1). Some nodes may be connected which are not actually adjacent, e.g. edge (4), but we cannot determine the exact adjacency relation within a cell without pixel level processing which is avoided here.

- Identically labelled nodes in neighboring cells could be a part of the same color region in the image and therefore are connected by an edge. An example of such an edge where the two nodes connected are from the same region is labelled (2). The edge labelled (5) shows a case where two nodes are connected but are actually not from the same color region. Most of these type of edges can be removed by checking for the presence of the color along the cell boundary, e.g. the color **a** is not present along the cell boundary.

  If two regions of different colors are adjacent, there will be at least one cell where peaks from both the regions will be present *together* and therefore will be connected by an edge in *that* cell. This is untrue only if the region boundary and cell boundary coincide exactly, which is a very low probability event. So it is not necessary to connect nodes of different color labels across cell boundaries which may, in fact, add some adjacencies where none exist.

- Diagonal edges are not considered because they would be redundant e.g. (3), and may add some edges where no adjacency is possible e.g. line (6). If two color regions are adjacent, they cannot have peaks only along a diagonal since there is just a single pixel of contact between two cells along the diagonal.

Putting the above discussion concisely, let nodes of the intermediate SPG be of the form c^i_m, where m is the peak color label of the node and i is the cell in which it is located. There is an edge E between two nodes of the graph if the following condition is met.

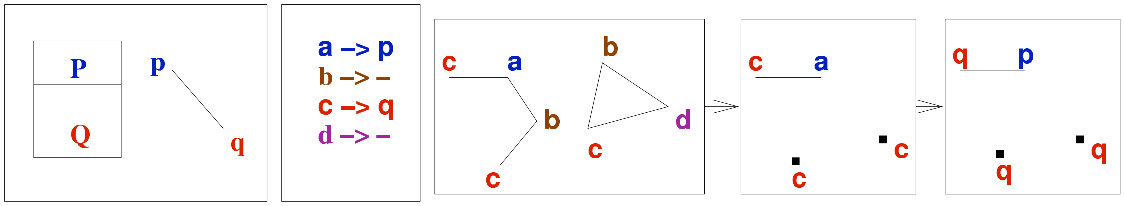
- $E(c^i_m, c^j_n)$ if $i = j$ OR $\{m = n$ and (i, j) are 4-neighbors$\}$.

The intermediate graph obtained is not scale invariant, since a larger region would produce more nodes in the graph. The smaller, scale invariant SPG which still captures the spatial relationships between colors is obtained by *collapsing* connected nodes of the same color label into a single node of that color label. The graph may still have multiple nodes of the same color label, but only if these peaks were spatially disconnected in the image. Figure 3.8(c) shows the SPG obtained by collapsing the intermediate graph in Figure 3.8(b). The SPG is computed offline for all database images and stored using an adjacency matrix representation.

The spatial proximity graph (SPG) description has a number of very useful properties. Apart from being scale and orientation invariant, it can be computed easily for all types of images, with or without prominent color boundaries. The SPG shows all possible pixel-level adjacencies that could appear in an image, without going through pixel-level processing. So any color adjacency relationship present in the image is still captured in this simplified graph. On the other hand, the graph is approximate since it may indicate some possible adjacency relationships for which there is actually no pixel-level adjacency in the image (for example, edge **b-d** in Figure 3.8(c)).

3.4.2 Matching SPGs

The problem tackled during the online second phase is to detect if the query color graph occurs as a sub-graph of the candidate image SPG. However, the whole image SPG need not be used. At the end of the first phase of retrieval, the correspondence between color labels in the image and the query peaks are available for each image in the retrieved list. The color label of the nodes in the image SPG are replaced by the query peak number they matched using the available correspondence. Any node



(a) (b) (c)

Figure 3.9. SPG filtering on the synthetic example in Figure 3.8 (a) Query image and graph (b) Correspondence between query and candidate peaks obtained from first phase of matching (c) Construction of reduced SPG from the SPG shown in Figure 3.8(c) by deleting unmatched peaks and relabelling nodes

from the image SPG which does not match a query peak are removed. Figure 3.9 shows the process of constructing the reduced SPG from the SPG computed offline.

(a) (b)

Figure 3.10. Example of SPG filtering (a) “Blueberry Morning” query image with SPG superimposed (b) A false match with reduced SPG superimposed

Figure 3.10 shows an example query graph and the reduced image SPG where a false match is detected. For ease of understanding, the labeled nodes in the graph have been placed on the region from which the color peak was obtained. The “Blueberry Morning” query image on the left has four peaks labeled 0-3 which correspond to the colors red, light brown, dark brown and blue in the image. The graph indicates that red and blue should be adjacent and the two shades of brown should be adjacent. The reduced SPG of the false match has colors 1 and 2 (browns) in close proximity, but 0 and 3 are not connected, leading to a mismatch with the query graph.

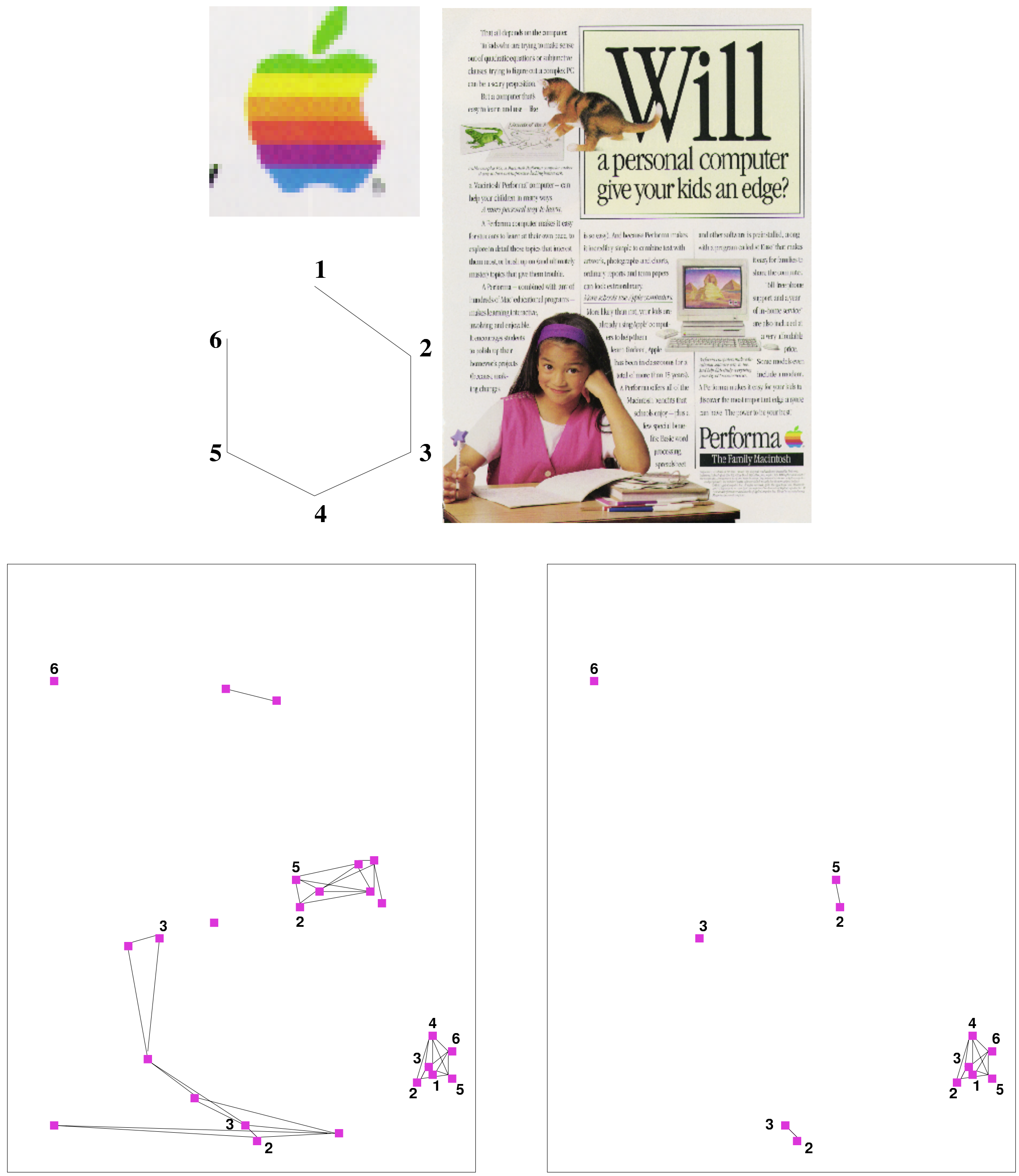


Figure 3.11. Example of reduction of SPGs after phase 1 in a true match: (top left) query and query graph; (top right) a correctly retrieved image; (bottom left) SPG stored offline; (bottom right) reduced SPG in which a match was detected.

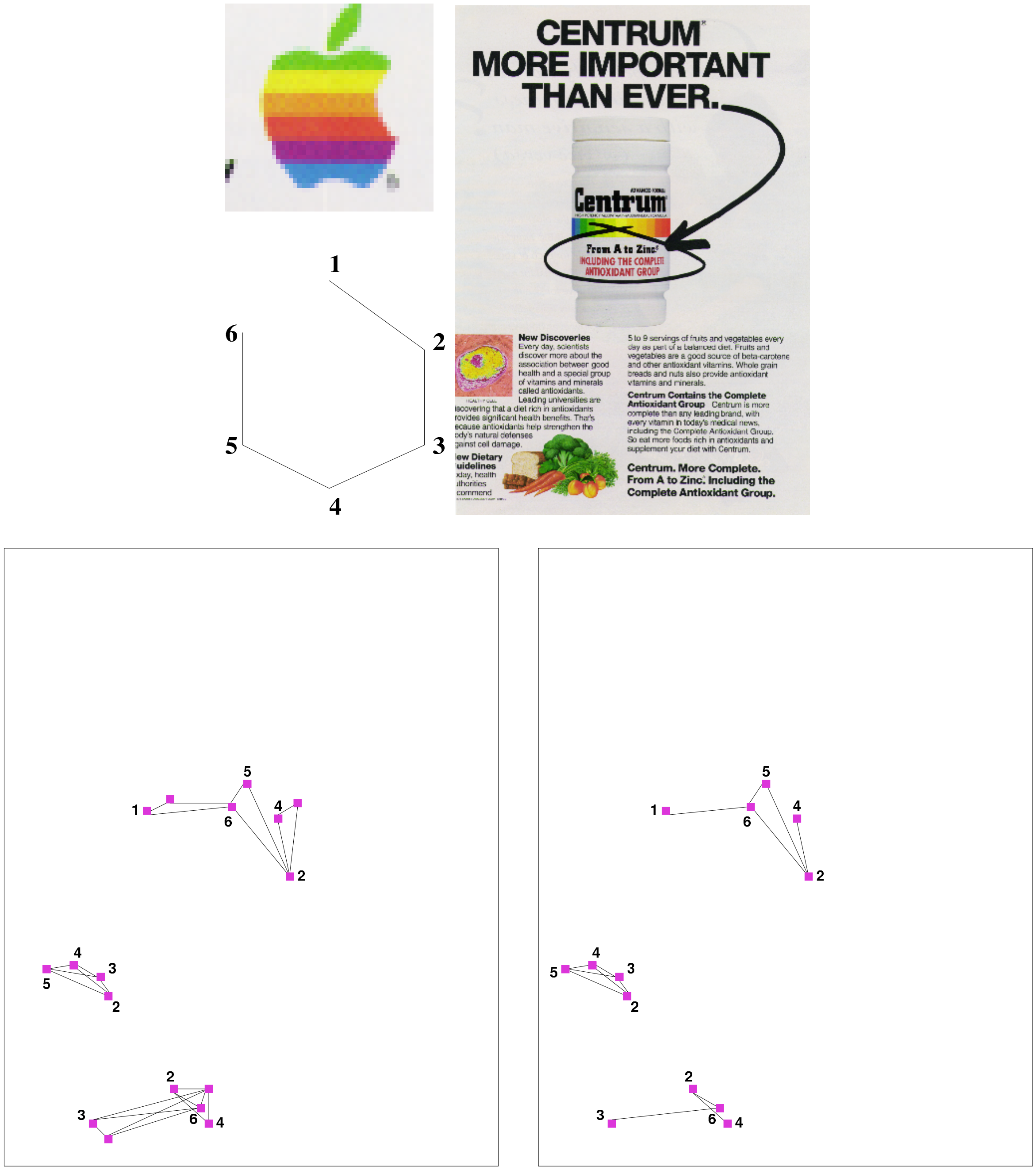


Figure 3.12. Example of reduction of SPGs after phase 1 in a false match: (top left) query and query graph; (top right) false match retrieved after phase 1; (bottom left) SPG stored offline; (bottom right) reduced SPG which did not match the query graph (hence this image is deleted by phase 2)

Figure 3.11 and 3.12 shows the drastic reduction in the SPG of real images when only nodes which matched a query peak are considered. The SPG of the "Macintosh" advertisement image containing 25 nodes and 40 edges is reduced to a graph containing 12 nodes and 17 edges. The query graph needs to be found in the reduced SPG. In this case, the query graph matches the bottom right cluster in the "Macintosh" advertisement. The query graph is not present as a sub-graph in the reduced SPG of the false match. Checking for this condition is an instance of the subgraph isomorphism problem which is known to be NP-complete. However, due to the restricted nature of this problem where the reduced SPG nodes are labelled with the same labels as the query graph, the matching computation is feasible. The running time is of the order of $O(n^m)$ where n is the size of the query adjacency matrix and m is the maximum number of instances of a color label in the reduced SPG, typically 3 or less.

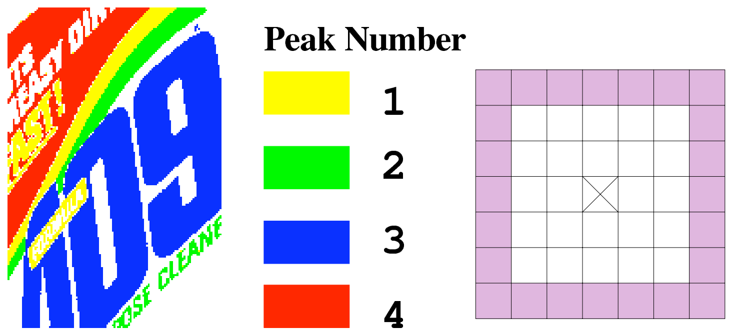
The average search time is further reduced by starting the matching process with the query peak which has the minimum number of instances in the reduced SPG of the image. As an example, the reduced SPG in Figure 3.12 contains 13 nodes; however, a mismatch is detected by checking just a single node when the above ordering is used during matching. The node labelled **1** has the least number of instances (1), and so is selected as the starting node. In the query graph, there is an edge between label **1** and label **2**, but no edge connecting the node labelled **1** to any node with label **2** is found in the reduced SPG of the false match.

3.5 Query construction and processing

FOCUS has an interactive user interface where the user can select a query from a variety of images in the database by marking a sub-image which covers the object of interest. The query image should not contain any background colors but it is not necessary to include the whole object exactly; including the salient colors of the object in a sub-image embodying their spatial relationships is sufficient. An example of query selection from a "Macintosh" advertisement is shown in Figure 3.18(a). A part of the advertisement is selected as query using a box enclosing all the significant colors in the object.

The processing required on the query image includes the computation of query colors and the construction of a graph to represent the color adjacency relationships between the query colors. Both these processes are different from the peak and graph extraction used for offline processing on the database images. The differences are warranted since query processing is done online and the amount of processing is directly related to the image size. Small query images are, thus, desirable (larger query images, when provided, can be sub-sampled to follow this guideline). In a small query image, there is usually an insufficient number of pixels to support division into fine grain cells for histogram construction and peak detection. However, this is not a problem for accurate peak detection since there is no background included in the query. The query color peaks can be computed directly from the global histogram in the absence of interfering background colors. The histogram construction and peak detection processes are the same as described for offline database image processing.

SPG construction as described for offline processing, uses image peaks localized in cells which are relatively small compared to the size of the image. Since this fine subdivision of the image is not feasible in the query image due to size constraints, at best a coarse division is possible. However, it is not possible to maintain scale invariance with SPGs constructed from coarse division of the query image as explained below. In a coarsely divided query image, two peaks could be in the same cell and thus be connected by an edge in its SPG. However, when a bigger copy of the query object appears in a database image, these two peaks could now be in different cells with no edge between them. This would create a mismatch with the query graph. On the other hand, if there was adjacency at the pixel level between two colors in the query image, this would be reflected in the SPG of the database image even if the query object was of a much larger size in the database image. So if the query graph represents pixel level adjacencies in the query object, it will match database images containing the query object at any scale.



(a) (b)

	1	2	3	4
1	9292	381	1	150
2	1730	10436	42	0
3	2	82	10042	0
4	59	0	0	27301

(c)

	1	2	3	4
1	1	1	0	1
2	1	1	1	0
3	0	1	1	0
4	1	0	0	1

(d)

Figure 3.13. Steps in query processing: (a) Query image labelled with the peak color labels (b) Mask defining neighbors - the cross marks the center pixel and the shaded pixels are its neighbors (c) Pixel pairs counted supporting each adjacency (d) Query color adjacency matrix obtained by thresholding (c)

The graph describing the query color relationships is a true color adjacency graph where edges in the graph represent actual pixel-level adjacency between the two connected color regions. The steps in the construction of the query adjacency graph are listed below :

- The query image pixels are first labeled with the peak color label they belong to, as shown in Figure 3.13(a).

- An empty table with the peak labels as row and column is initialized. For each pair of neighboring pixels in the query image with color labels i and j, the table entry (i, j) is incremented. This yields a table of the form shown in Figure 3.13(c). Entry (i, j) of the table gives the number of neighboring pixel pairs which had i and j as their color labels. There are regions of intermediate or mixed color at the boundaries between two color regions. To take care of boundary transition effects, a mask (shown in Figure 3.13(b)) is used to define neighbors of the center pixel.

- The actual numbers in this table are not a reliable guide of the extent of the common edge between two color labels because of the presence of unlabeled pixels produced by intermediate colors. Thresholding this table using a small threshold (0.025 of maximum off-diagonal term) to remove entries very close to zero produces the query adjacency matrix shown in Figure 3.13(d) which is much more stable. A non-zero entry at position (i, j) in this matrix indicates that a region with color label i and a region with color label j are adjacent in the image.

3.6 Experimental results

The FOCUS system was tested on a diverse image database to judge its retrieval performance and speed of retrieval. An online user interface was developed for testing and demonstration of the system's capabilities. A snapshot of the interface is shown in Figure 3.14. The figure shows a query selected by the user marked by a green box and stored in the query space on the right. The query includes all the significant colors of the "Ziploc" box while excluding any colors from the background. The bottom panel of the interface shows the retrieved images after the first phase of matching. The user can activate phase 2 matching by clicking on the "Refine Results" button.

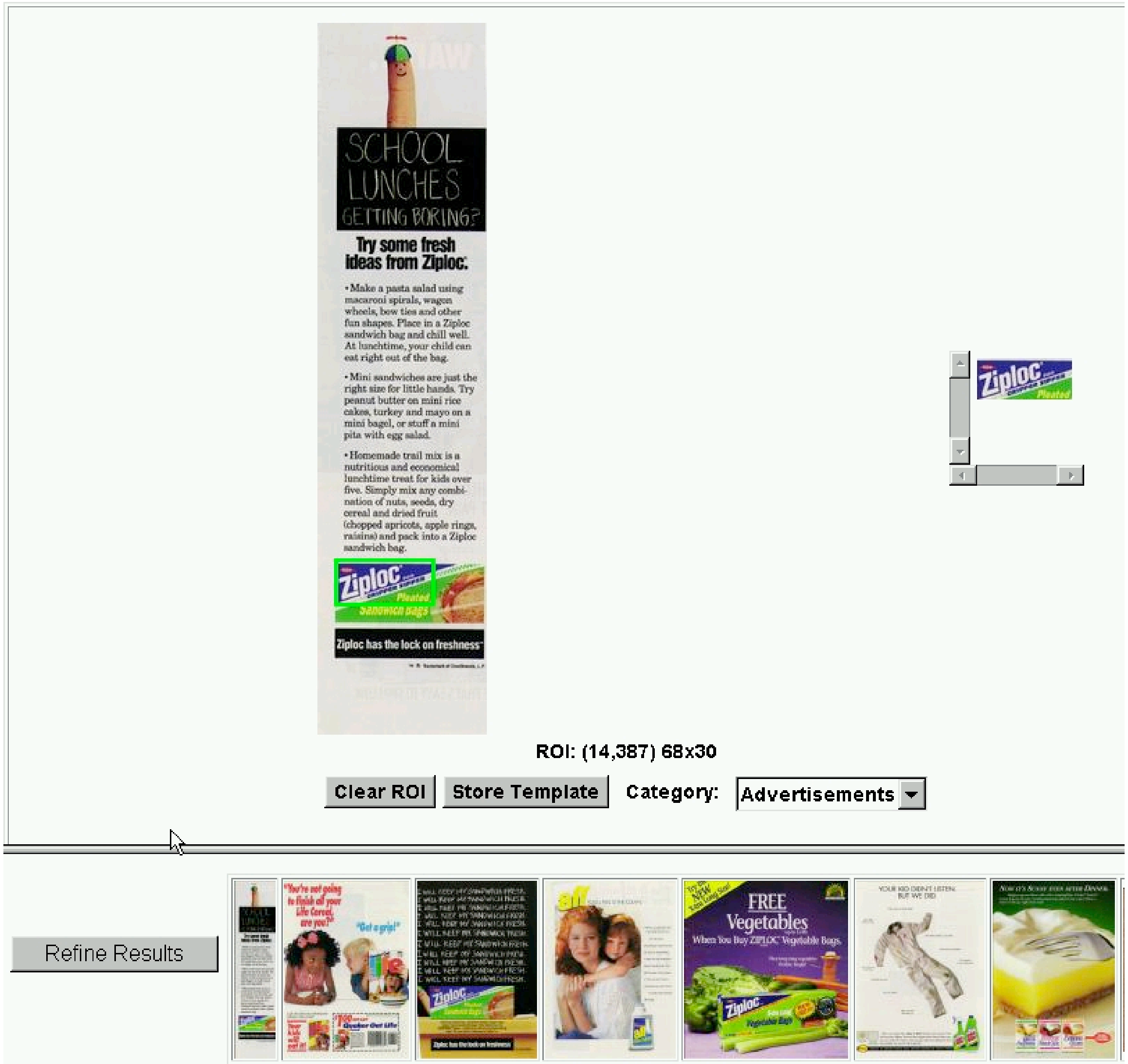


Figure 3.14. Online user interface to FOCUS showing a query box being selected and the results after first phase of processing (where the first, third and fifth images contain the query object)

3.6.1 Test database

A test database of 1200 images was created for this work using various images of multi-colored objects. There are 400 advertisement images scanned from magazines and 800 color images of natural objects including birds, fish, flowers, animals and vegetables obtained from commercially available CDROM image libraries. Most of the retrieval results will be provided with queries targeting the advertisement segment of the database, since this segment provides the most complex backgrounds and extreme variations in scale. Since the claim of being able to handle these two problems distinguishes our proposed method from other color image retrieval methods, this segment generates the most appropriate test results.

Effort was made to obtain different advertisements of the same product to provide objective ground truth, instead of relying on subjective judgements of similarity between different products. All reported recall-precision figures are based on the actual ground truth - similar objects retrieved are not counted as correct retrievals.

3.6.2 Retrieval performance

The retrieval performance was tested on two different query sets. The query sets were constructed from patches from products which have at least two different advertisements in the test database. The average number of target images for queries in the query sets is 3.3. Figure 3.15 shows some examples of query images included in the tests. The first query set (set I) consists of 25 randomly picked queries. The second query set (set II) of 15 query images constrains the queries to those which have more than three different colors. There is some overlap between the two sets i.e. some queries are present in both sets.

The retrieval results obtained by this system can be judged by the criteria used in text retrieval, *precision* and *recall*. Precision is the proportion of correct retrievals out of the images retrieved. Recall is the proportion of correct retrievals out of all the images in the database that should have been retrieved for the given query. Table 3.1 gives the precision at high recall (90%) and average precision obtained in tests with query sets I and II. There was significant improvement in precision due to the deletion of false matches in phase II, especially when there are more than three colors in the query. At three colors or less, the number of spatial color relationships that can be captured in the SPG is much fewer, and therefore, the false match filtering in

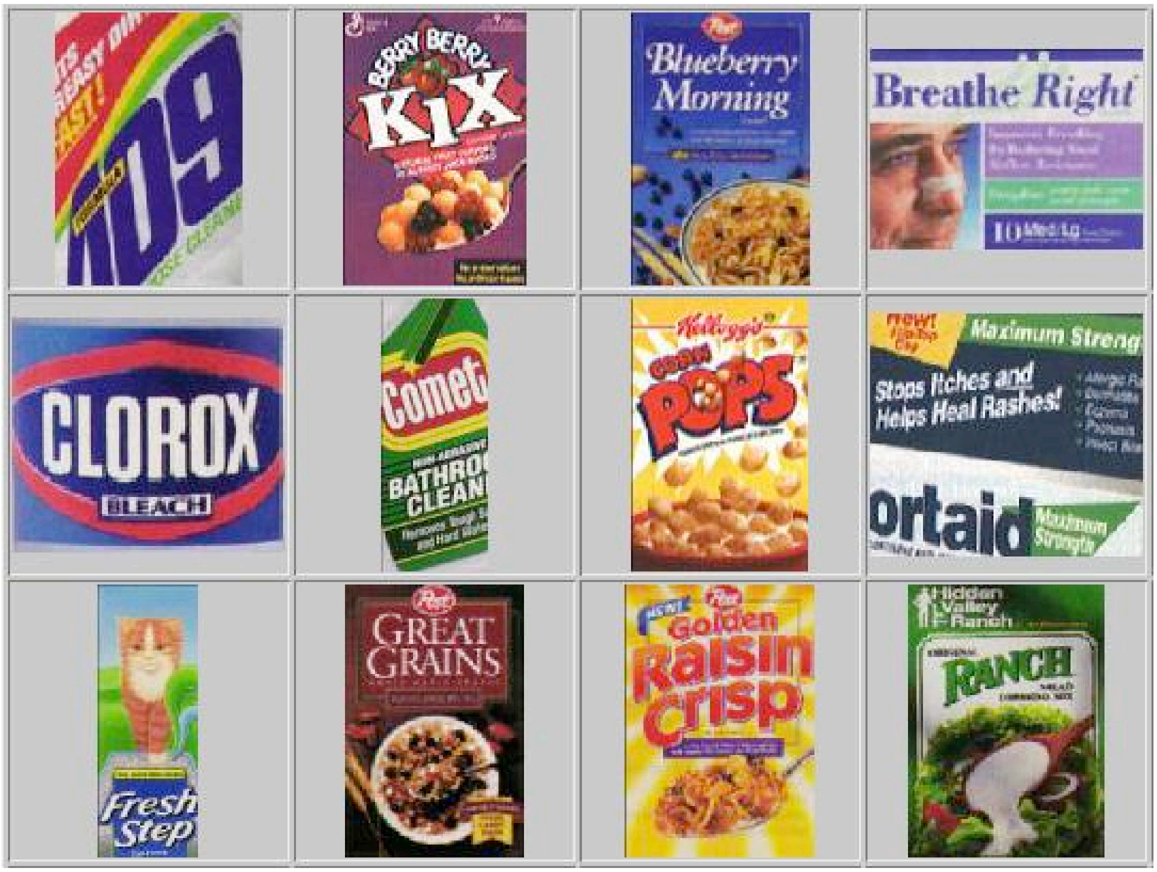


Figure 3.15. Some query images used in testing retrieval performance

phase II is also less effective. Figure 3.16 shows the recall-precision trade-off for these two query sets.

Figure 3.17 shows the first 10 retrieved images after the first phase of retrieval and the top five images after completion of the second phase of processing for a typical retrieval. Some of the false matches in the original retrieved sequence have been eliminated by the second phase, so that a correct match which was earlier ranked 8 in the sequence has been moved to the top five bracket. An example of retrieved images with a query with six colors is shown in Figure 3.18 where the “Macintosh” apple, an 80x80 sub-image of the 1200x1000 pixel original image, is selected as the query. Only the first three images (which are correct retrievals) remain after the second phase of processing. Table 3.2 shows numerical results for some of the other queries on advertisements.

	Precision at 90% recall (%)	Average precision (%)
Set I		
After phase I	38	44
After phase II	54	60
Set II		
After phase I	45	50
After phase II	70	75

Table 3.1. Retrieval performance on query sets I and II

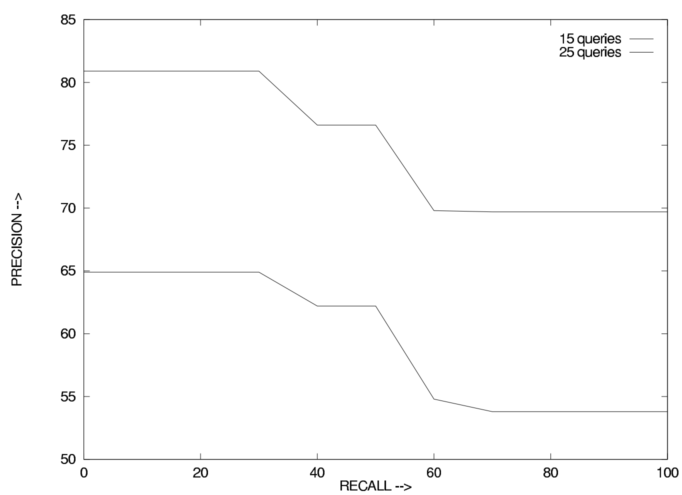


Figure 3.16. Recall-Precision graph after Phase 2 for a set of 25 randomly selected queries (set I) and 15 queries with more than three colors each (set II)

Name	Recall	Prec 1	Prec 2
Breathe Right	3/3	3/3	3/3
L'oreal Casting	4/4	4/7	4/5
Comet	2/2	2/23	2/9
Dannon	3/3	3/5	3/5
Fresh Step	2/2	2/3	2/2
Hidden Valley	7/7	7/20	7/15
Macintosh	3/3	3/3	3/3
Merit	6/6	6/18	6/12
Reynolds	4/4	4/13	4/6
Sun Crunchers	2/3	2/13	2/8
Total	36/37	36/108	36/68

Table 3.2. Retrieval results for 10 queries : (Recall) Images retrieved/No. of correct images in database (Prec 1) Precision after Phase 1 (Prec 2) Precision after Phase 2

Figure 3.17. Refinement of retrieval by second phase of processing : The query is marked by a white box (Top two rows) Results after the first phase of retrieval (Last row) Results after completion of second phase

Figure 3.18. Example of query selection and result : (Top) Portion of image (from original image shown in Figure 3.11) with query marked by a box and the query image generated. (Bottom) Retrieved images - the first three images have the query object embedded in the lower right corner

The time taken for a complete cycle of retrieval consists of the query processing time, phase 1 matching and phase 2 matching. All times mentioned are on a 400 MHz Pentium III PC and are averaged over many trials. Query processing takes about 0.05 sec on a query of size 100x200, which is the average size of queries tried. Phase 1 matching takes 0.05-0.1 sec and phase 2 matching takes about 0.005 sec for each image in the list produced by phase 1. Since this list has 30 images on an average, the second phase takes about 0.15 sec. The retrieval process is fast enough to be scalable to very large databases. For example, if the database is scaled to 10^6 from the current 10^3 images, the query processing time is unchanged, the phase I matching time is doubled (since it increases logarithmically with size of database) and the phase 2 matching time *per image* is unchanged. The total phase 2 matching time depends on the number of images returned by phase 1. Since the images are ranked, the top n images may be processed by phase 2, selecting n depending on the time available.

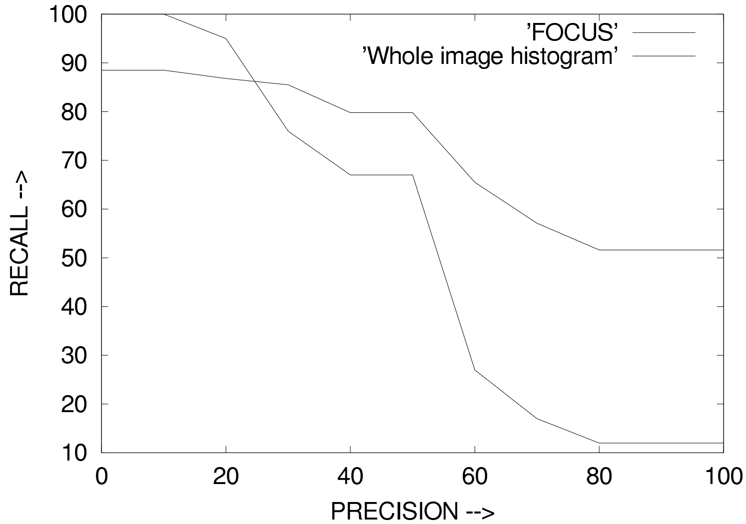


Figure 3.19. Comparison of recall-precision graphs obtained with FOCUS and whole image color histogram-based retrieval on a set of 20 queries (set III)

Six sample retrieval results are shown in Figures 3.20 and 3.21 with the query marked by a white box. Figure 3.20 shows some results from the advertisement domain which has been discussed throughout this chapter. Figure 3.21 shows that FOCUS works well when queried with natural objects when they have multiple colors, though there is no objective criteria to judge the accuracy of retrieval in this case. The system shows good retrieval performance even when the query object is present in different sizes, orientations and with different backgrounds in the candidate images.

We compared the retrieval results obtained using FOCUS with color histogram-based retrieval. Figure 3.19 provides a comparison between the two methods. The performance of FOCUS is clearly better. It is to be noted that the initial (at low recall) high precision obtained by the histogram-based method is an artifact of the fact that whole image matching ensures that the first retrieved image is the same as the query, and therefore, correct. However, this is not very useful, since the user already has access to the query image, and is actually searching for other images in the database which match its content.

Figure 3.20. First five retrieved images for three different queries (in the advertisement images domain) in order of rank. The query is marked by a white box. (First row) First, second and fourth images are correct matches (Second row) First, second and fifth images are correct matches (First row) First, second, third and fifth images are correct matches

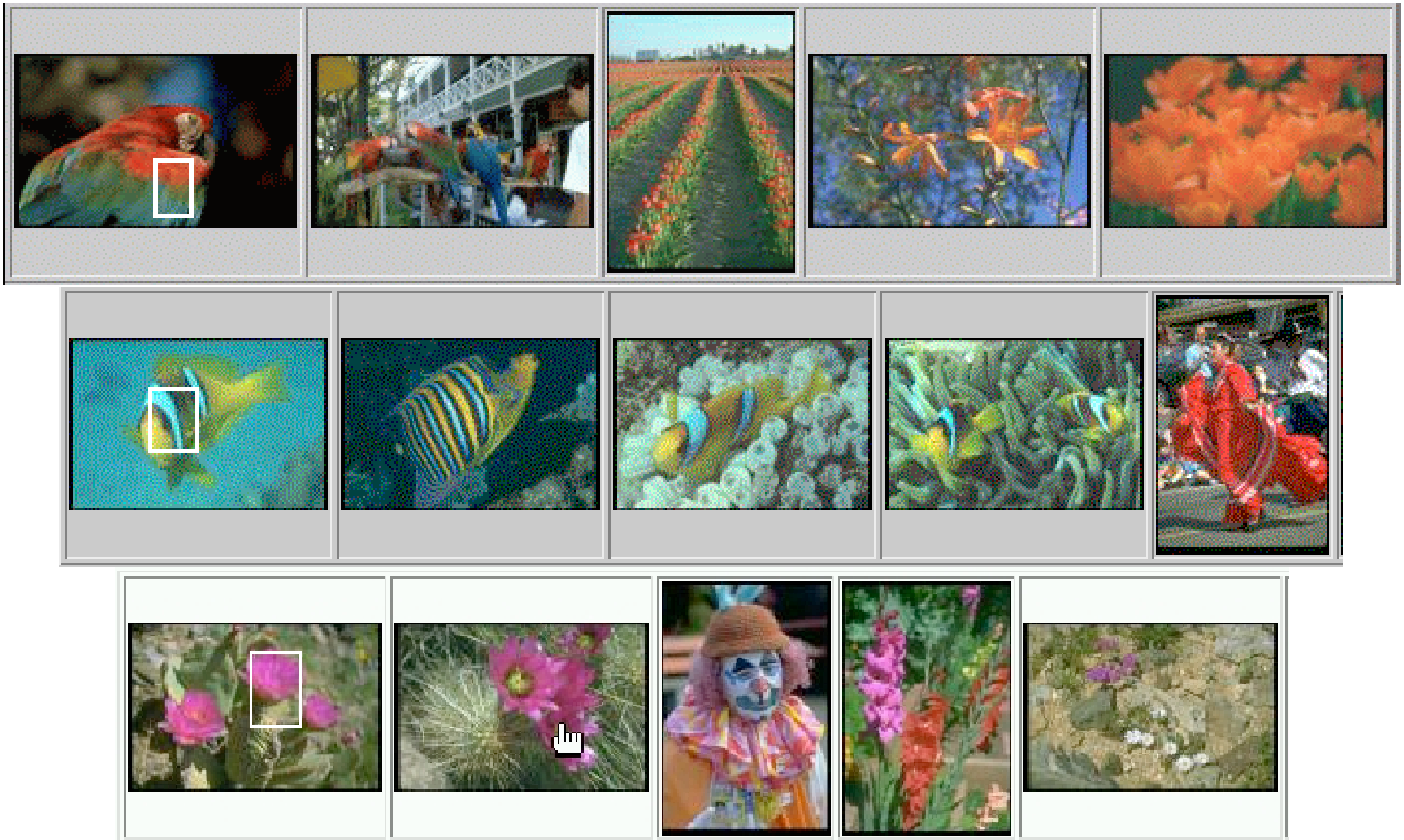


Figure 3.21. First five retrieved images for queries in the natural objects domain, in order of rank with the query marked by a white box.

3.6.3 Effect of cell size and cell boundary location

Since both peak detection and SPG construction are based on sub-division of the image into cells, the size and location of the cells seem to be important parameters embedded in the algorithm which could affect overall performance. In this subsection, we explain the justifications for our choices and examine the effect of these two parameters.

3.6.3.1 Cell size

The cell size selected is appropriate for peak detection when there is at least one cell in the image which covers only the object of interest, and no background. This ensures that an accurate peak description is obtained from that cell and this is added to the peak description of the image. The accuracy of SPG construction increases with smaller cell sizes, since colors within a cell are considered to be adjacent without pixel-level evidence of adjacency. In the bounding case where the whole image is a single cell, the SPG is useless since it will be completely connected, providing no discrimination between different color relationships. So from both these considerations, it appears that the cell size selected should be as small as possible, limited by the number of pixels needed to populate a color histogram and make detection of peaks possible.

However, the peaks and corresponding SPGs generated from small cell sizes suffer from a lack of robustness. These peaks distinguish between very fine differences in shades of a color i.e. a major color present in an object may be represented by many different peak labels, each representing a shade of the color which formed the majority (peak) in a small section of the object. Since these cells may occupy a very small portion of a large object, peaks may be detected which are obscured when the same object is represented at a smaller scale, resulting in a mismatch. In addition to failure of peak matching in many cases, SPG matching is also affected. This is due to the difficulty in obtaining one-to-one correspondences between query peaks and target peaks when the peaks are very close to each other in the color space. For example, according to the query graph a particular shade of green (say, *green1*) must be next to another shade (say, *green2*), while the target may have a third shade *green3* next to *green1* resulting in a mismatch. With a larger cell size, usually there are only one or two shades of a single color, and peaks are spread apart and distinct in the color space.

It is expected that larger cell sizes will cause system performance to degrade since small objects like the "Macintosh" logo will not be accurately described. Even in the absence of interfering background around a very small object such as the logo, the size of the peaks when expressed as a fraction of the total number of pixels in the larger cell would fall below the minimum threshold for peak size. The SPG description would also be more approximate. On the other hand, a very small cell size would also cause a drop in recall because of the factors discussed above. Figure 3.22 shows a comparison of retrieval performance between the system with the default 100x100 cell size, a cell size of 200x200 and a cell size of 50x50. The recall-precision scores are based on a query set of 20 images with three or more colors. The system performance with the larger and the smaller cell size is poorer at all recall levels. The average precision fell from 71% to about 63%. The retrieval results are particularly poor on small objects in the query set when the larger cell size was used, and on objects that have gradual color transitions (many shades of a color) when the smaller cell size was used.

3.6.3.2 Cell location

If the cells are shifted with respect to the image while keeping the cell size fixed, the image area covered by each cell changes. This could lead to a difference in the peaks detected as well as the SPGs obtained.

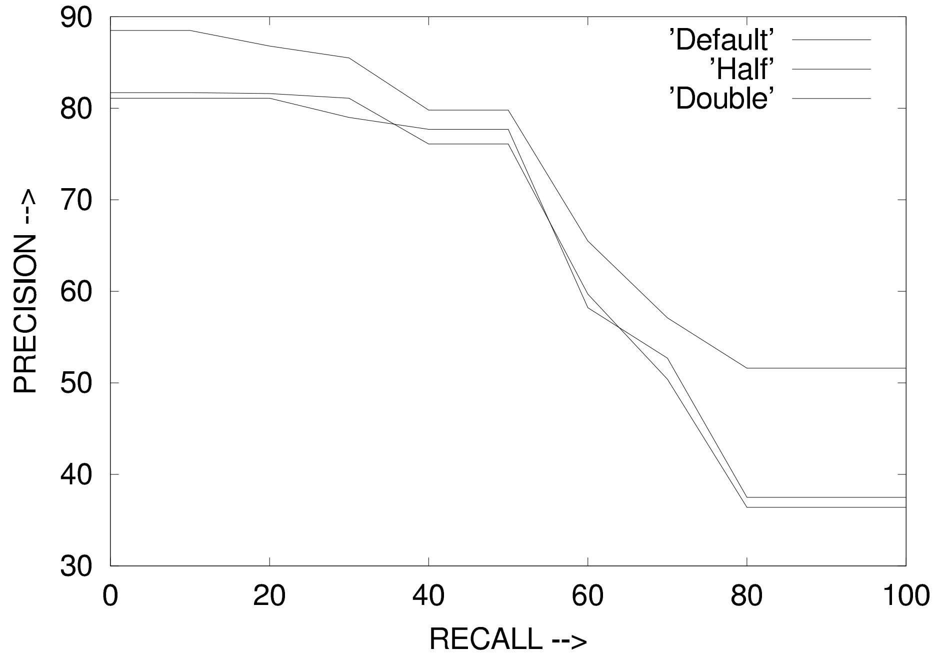


Figure 3.22. Recall-Precision graph after Phase 2 for a set of 20 queries (set III) with cell sizes of 100x100 (default), 200x200 (double) and 50x50 (half)

Figure 3.23 shows the comparison in system performance when the cells are shifted by half default cell width (50 pixels) while keeping the cell size at 100x100. Though there are some differences between the two graphs, there is practically no change in system performance metrics - the average precision with the shift is 70% (compared to 71% with the default set-up).

On examining the images which caused the differences, it was observed that these were images where the object size in the target image was small (comparable to the cell size). In such cases, better retrieval was obtained when the cell boundary did not pass through the object, and the cell location which produced such boundaries did better. There was very little effect on the peaks, SPG and retrieval for larger target objects. Figure 3.24 shows two such images. In the "Macintosh" image, the shift causes the cell boundaries (blue) to split the logo between two cells. This results in missing some of the color peaks from the logo since the number of pixels is too small to cross the peak size threshold once the object is divided. Whereas, the shifted

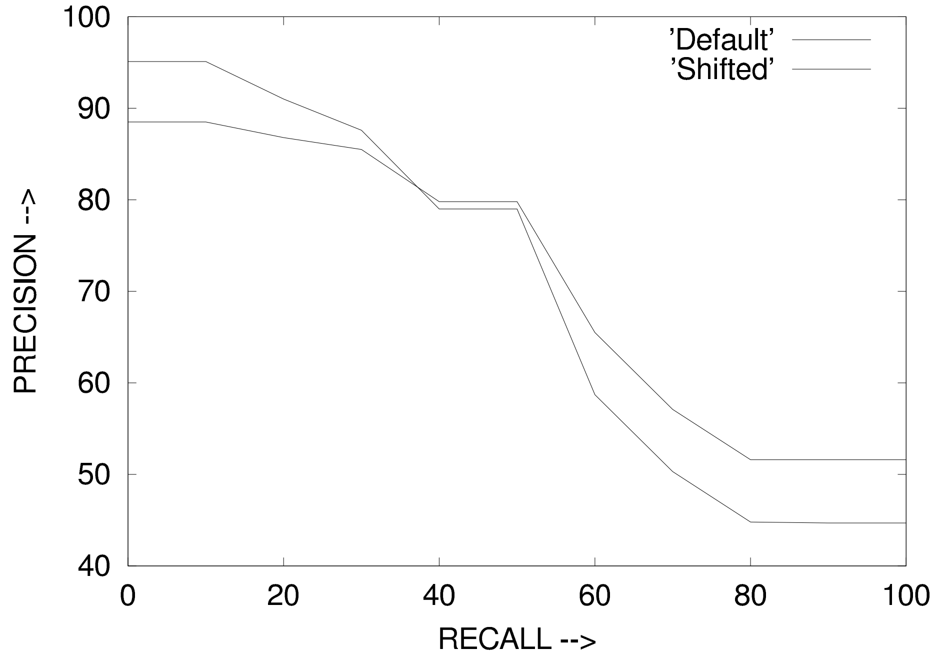


Figure 3.23. Recall-Precision graph after Phase 2 for a set of 20 queries (set III) with default cell locations and cell locations shifted by half cell width (50 pixels)

cell location provides better boundary locations in the "Taco Bell" image. The small yellow portion of the bell was earlier split between two cells and not detected as a peak. In the shifted cells, all the major colors, including the yellow portion of the bell, are correctly detected.

We conclude that the cell location has an effect on retrieval performance for small target objects, but the effect could be positive or negative depending on the location of the object in the image. However, the overall system performance is not affected, the positive and negative effects cancel each other out.

3.7 Conclusion

We have presented a fast, background-independent color image retrieval system which produces good results with multi-colored query objects. The main contributions of this work is to propose two scale- and orientation-invariant features which can be combined to produce good retrieval results even with database images with

(a) (b)

Figure 3.24. Examples of images where the shift in cell location creates major differences in peaks detected (a) Default produces better peak description (b) Shift produces better peak description. The black dashed line shows the default location of the cell boundary and the blue lines show the shifted locations.

significant background clutter where the query object appears at different scales, orientations and location in the candidate images. The speed of the system and the small storage overhead make it suitable for use in large databases with online user interfaces.

Generating histograms in local cells, combined with the use only of peak locations provides a reliable color description of complex images. The spatial proximity graph structure proposed is simple enough to be easily generated for complex images and yet captures color ajacency information that can be used to reduce false positives. We also propose an effective two-phase strategy for matching where information computed after the first phase is exploited during the second phase computations to make the process computationally feasible.

CHAPTER 4

INDEXING A DATABASE OF FLOWER IMAGES

4.1 Introduction

Most existing image retrieval systems cast the retrieval problem as the users' need to find other images *similar* to a given query from an image database, where similarity is computed using a distance metric in a low-level image feature space. However, similarity is a semantic notion, not necessarily captured by low-level image features computed from the global image. This has lead to a great deal of interest in the problem of *meaningful* retrieval from image databases in recent years.

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 in an image, has no semantic interpretation by itself. In images, meaning is found in objects and their relationships. However, segmenting images into such meaningful objects is in general an unsolved problem in computer vision. Fortunately, many low-level image attributes like color, texture, shape and "appearance" may often be directly correlated with the semantics of the problem. For example, in our previous database, product packages (e.g., a box of Tide) have the same color wherever they are found.

These low-level attributes must be used with care if they are to correlate with the semantics of the problem. For example, many image retrieval systems (see [3, 49]), 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. Even in a restricted domain, the background can play an undesirably important role in the retrieval. For example, a picture of a flower against a background of green leaves may not be able to retrieve images of the same flower against a background of soil or in a close-up without any background. This is because the query contains green areas which are given equal importance as the flower regions. The presence of backgrounds is a major problem which needs to be handled intelligently before retrieval can be effective.

While many image retrieval algorithms have focused on retrieving images from general image collections, there is a growing number of large image databases which are dedicated to specific types and subjects of images. When using general-purpose retrieval strategies on these databases, it is easy to lose sight of characteristics of the domain which could be used to substantially improve the retrieval performance. There may also be special querying requirements in applications in the domain covered by the database. Restricting image retrieval to specialized collections of images or to specific tasks is more likely to be successful and useful. The restriction to specific domains does not make the task any less interesting, since the goal now is to provide better retrieval than what is possible using general-purpose algorithms.

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. A person who would like to check whether flowers similar to a new patent application exist in the database can provide an example image obtained from the new application. On the other hand, a person looking for flowers to cultivate may only be able to specify the flower type and a color name.

The research goal in this chapter is to provide a framework for using domain knowledge to isolate the object of interest (flower). The color of the flower can then be accurately computed from the extracted region. Unlike many other color based retrieval systems, 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.

4.2 Our Approach

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. For example, when the user provides the image in Figure 4.1(a) as a query, it is obvious to him that the query consists of a flowering plant with pink flowers. A naive retrieval system, on the other hand, will produce images which are predominantly blue, an effect of the background being blue and occupying a large portion of the query image.

Despite its specialized nature, this database offers some challenging problems. 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 Figure 4.1. There are two main problems to be addressed in this application : the problem

(a) (b) (c)

Figure 4.1. Example of database images showing different types of background distributions

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.

We would like to use the characteristics of the flower patents domain to automate the segmentation and indexing process. Most of the domain knowledge is in the form of natural language statements. For example, for most natural subjects, a lot of information about the object color is common knowledge e.g. flowers are rarely green. 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, translating these into rules which can be used to build automated algorithms is non-trivial.

We have constructed a mapping from the 3D color space (commonly used to represent digital images) to a natural language color name space, so that color name-based rules can be exploited. The color name space also allows us to group specific color names into larger color classes which enables a fine or coarse division of the color space as required.

We have also identified a set of observations about the spatial distribution of background and foreground colors which are true of most of the images in this domain. We have developed an iterative segmentation algorithm which uses the available color and spatial 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.

4.3 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 show that domain knowledge can be used to automatically extract a region from the image which has a high probability of being a flower region.

4.3.1 Domain knowledge for flower database

The types of *a priori* information available for this application can be categorized into *spatial* and *color-based* domain knowledge. Examples of spatial domain knowledge include information about the location of foreground and background elements and the sizes of the objects in an image. Spatial domain knowledge used in our approach is derived from commonly followed photographic principles of focusing on the object of interest and keeping it in the central part of the image, as enumerated below.

1. Background colors are usually visible along the image periphery.

2. There is only one type of flower in the image.

3. Other colored objects present in the image do not dominate the flower regions.

4. Flowers occupy a reasonable part of the image.

5. The flower regions are unlikely to be present *only* near the image periphery.

Color-based domain knowledge takes the form of natural language facts about the color of the object of interest. In this case, we know that flowers are rarely *green*, *black*, *gray* or *brown*. This domain knowledge is true of most insect-pollinated flowers (most of the decorative flowers of value fall under this category). These flowers depend on insects for pollination and therefore, have evolved to be attractive to insects by producing nectar or perfumes. The flowers are also designed to be easily distinguishable from the background so that insects can locate them. Since the background usually contains green (leaves), brown (soil), gray and black (shadows) color regions, these colors are avoided to make the flower stand out from the background.

Since color-based domain knowledge is available in terms of natural language color descriptions, the color space needs to be mapped to commonly used color names. This mapping is useful for our goal of providing color name-based retrieval as well.

4.3.1.1 Mapping from color space to names

We need tables mapping points on a 3-D color space to color names which should agree with the 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 (Table 4.1) and generates 267 color names using hue modifiers (Table 4.2). 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 (Table 4.3) comprising groups of base hues.

The color names in ISCC-NBS system often have simpler commonly used alternatives, for example, 'very pale yellowish white' in the ISCC-NBS system is the color 'ivory' and 'light brownish yellow' is the color 'khaki'. The simpler names, like 'ivory'

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

Table 4.1. 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	

Table 4.2. Hue modifiers in the ISCC-NBS system

and 'khaki', which are often derived from commonly known objects of the same color, are obtained from the definitions in the 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 [25] 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.

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

Table 4.3. Color classes derived by grouping ISCC-NBS hue names and adding three 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

Table 4.4. 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 (Table 4.4). 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.

4.3.2 Iterative segmentation with feedback

Our approach to extracting a region which has a high probability of being a part of a flower is to successively eliminate 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 (shown in Figure 4.2) from the segmentation results obtained to redirect our choice of background colors and try a different hypothesis. The domain knowledge discussed in the earlier sub-section is used to eliminate some colors, generate a list of possible background colors and evaluate the correctness of the remaining segment.

4.3.2.1 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. The color classes black, gray, brown and green defined in Table 4.3 can be termed "non-flower" colors, since these colors are unlikely to be the flower color. We can

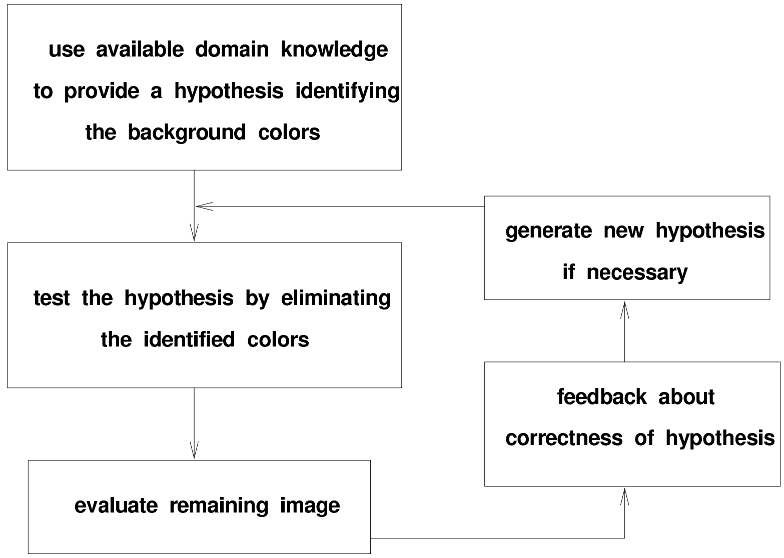


Figure 4.2. Our approach for automatic segmentation of flower regions : domain knowledge is used to generate the background color hypothesis and evaluate the remaining segment

eliminate some of the most frequently occuring background elements in flower images by deleting pixels which belong to non-flower color classes. 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.

Apart from using 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, as shown in Figure 4.3.

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 Figure 4.1. The flower may be totally embedded in the background (type **(a)**), the background and flower regions may interlace along the margins (type **(b)**) or the flower may fill the whole image (type **(c)**). In the first case, the observation is true, in the second case one needs to decide

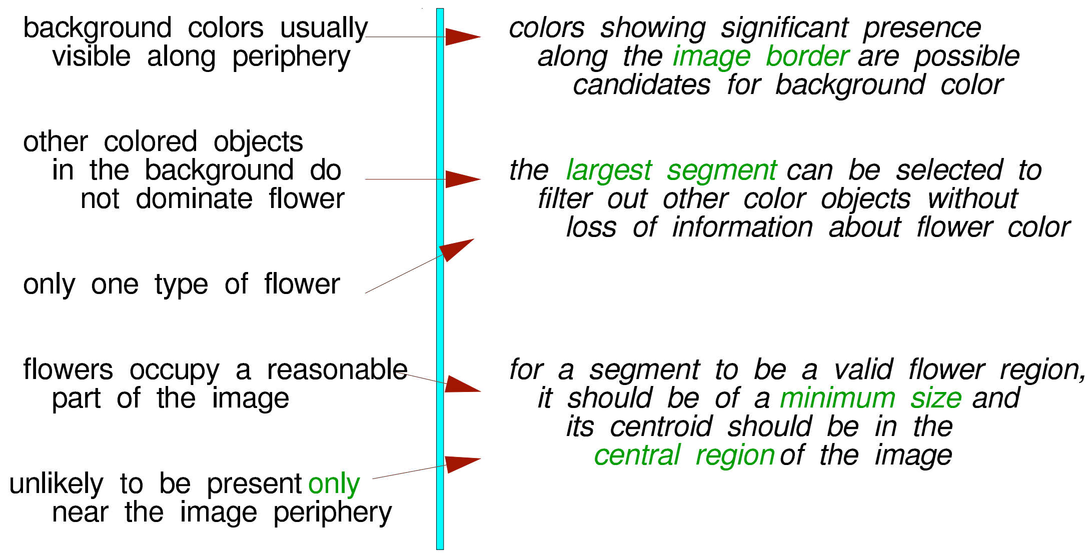


Figure 4.3. Translating domain knowledge into rules : Raw spatial domain knowledge is shown on the left, and the rules derived from them are shown on the right

which of the colors represent the background and in the third case the observation is false; and we need to be able to distinguish between these cases. The correctness of the choice of background color(s) is tested by evaluating the remaining image after the background color has been eliminated.

We can derive some useful guidelines for evaluating whether a segment in an image is a possible flower region 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, placing a constraint on the minimum expected size of the flower regions. 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. Thus, the center of the flower region is unlikely to be in the boundary region. The background may have other colored objects but they will not usually dominate the main subject, which is the flower, which makes the largest segment the most likely to be the flower region.

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.

4.3.2.2 Implementation of domain knowledge-based rules

The translation of the above general rules into algorithmic steps requires the definition of various regions in an image. These are marked in Figure 4.4. The ad hoc choices made in the implementation were selected based on observations on a set of 200 sample images and verified by checking the resultant segmentation output on a large database.

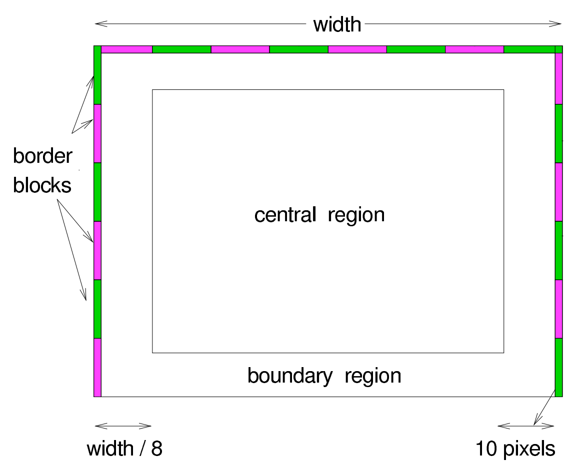
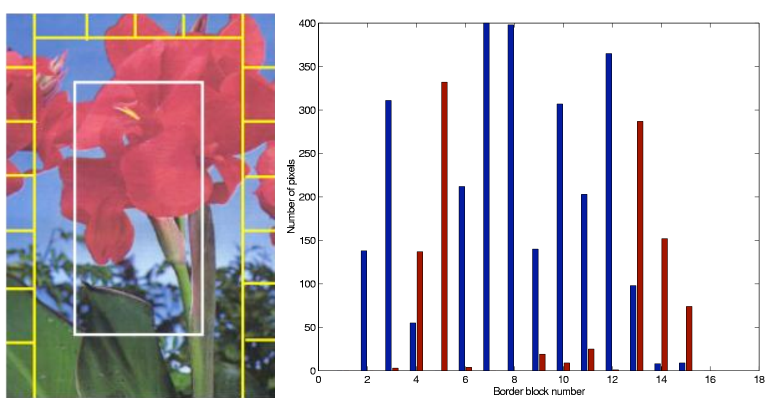


Figure 4.4. Definitions of image regions : Border blocks (shown in alternating color), central region and boundary region

The *central region* consists of the middle three-quarters of the image, and it is expected that most of the object of interest will lie in this region. The rest of the image is termed the *boundary region*. The image periphery or border is defined as a

10 pixel width region along the edge of the image. The image border is divided into 18 equal sized *border blocks* as shown in Figure 4.4.

We use two criteria for evaluating whether a segment produced is valid; its size and the location of its centroid. The minimum size of a valid segment is expressed as a fraction of the largest segment obtained after deleting the non-flower color classes in an image. Since some of the flowers are small, especially in images downloaded from the world wide web, this fraction is set to 0.025 i.e. the size of a valid segment has to be at least 2.5% of the largest segment obtained after deleting the non-flower colors. The centroid of a valid segment should fall within the 'central region' of the image as defined in Figure 4.4. These requirements are based on the domain knowledge discussed in the previous sub-section. The first requirement is based on the fact that in a photograph of the flower, the flower itself should occupy a reasonable part of the image. The second requirement is based on the fact that the object of interest should not be present *only* near the boundaries of the image.



(a) (b)

Figure 4.5. Detecting potential background colors (a) Dividing an image into border/central regions (b) Color distribution in border blocks of image, where the blue bars represent the color blue in the image and the red bars represent the color red

The possible list of background colors is detected by analysing the color composition along the image margins. The margins of the image are divided into border blocks as defined in Figure 4.4(a). The distribution of color classes in these blocks is computed and colors showing substantial presence in more than one-third of the blocks are marked as possible background colors. For example, Figure 4.5 shows the color distributions for the two color classes (red and blue) present in the border of the image. From this distribution, both the color *blue* and *red* are marked as potential background colors, since blue is present in 11 and red in 7 out of 18 border blocks in the image.

4.3.2.3 Segmentation strategy

We use the *connected components* algorithm whenever we need to identify segments in the image, where we consider an 8-neighborhood for computing connectivity and 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.

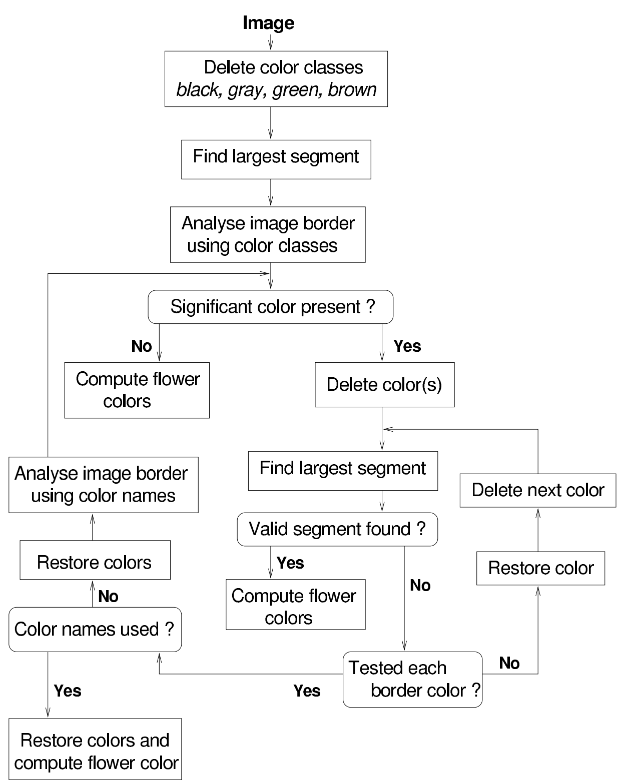


Figure 4.6. System overview of automatic region of interest segmentation in the flower domain

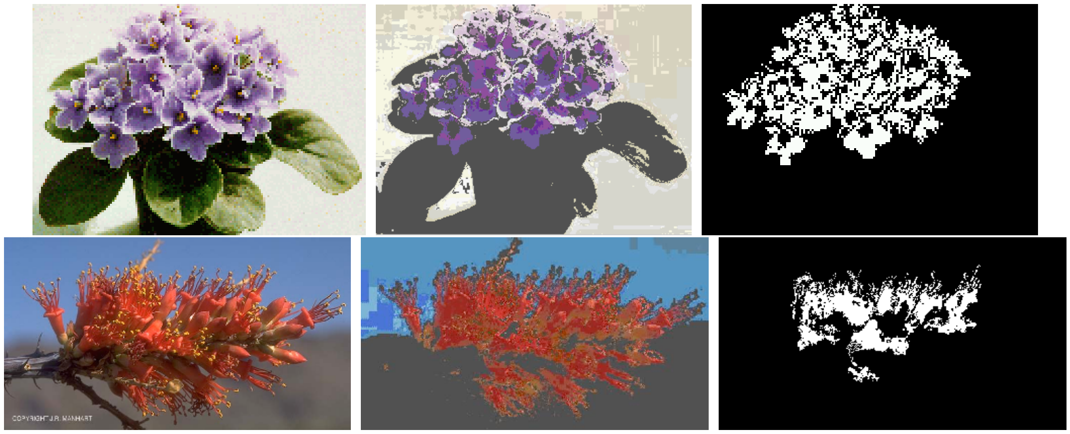
The image pixels (originally in the RGB color space) are labelled by their color classes as well as their nearest X-Window system color name. We use a coarse-to-fine strategy when using the color labels - the color class description is used first, so that the image has a few regions of broadly similar color, and the finer color name distinctions are used subsequently only when necessary.

The outline of the algorithm used to produce a segment from which the flower color is estimated is shown in Figure 4.6. In this section, we will discuss the steps in the algorithm along with illustrative examples.

(a) (b) (c)

Figure 4.7. Detecting a reliable flower region by eliminating non-flower colors : (a) original images (b) images left after deleting non-flower colors (c) largest valid segments

The color composition along the image border is analyzed to test if the pixels belonging to the color classes *black*, *gray*, *brown* and *green* are eliminated since these are non-flower colors. The remaining image contains the flower regions and may



(a) (b) (c)

Figure 4.8. Background elimination : (a) original images (b) image left after deleting non-flower colors (c) largest segments after the hypothesized background color (white for image on top, blue for bottom image) is deleted. The segments are both valid.

also have background colors not falling in any of these four classes. This image is segmented using connected components considering the image to be a binary image where the two classes of pixels are those which have been labeled as background (and thus eliminated) and those that still remain.

In photographs of flowers taken from a distance in natural surroundings, this process is sufficient to produce a good flower segment. Some examples are shown in Figure 4.7 where the final result of segmentation are the regions shown in (c). 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 rocks in Figure 4.7. 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.

The color composition along the image border is analyzed to test if the largest segment contains background colors in addition to the flower regions. If there are no colors present (as in the examples in Figure 4.7) and a valid segment has been obtained, no further processing is required. Otherwise, all pixels belonging to colors

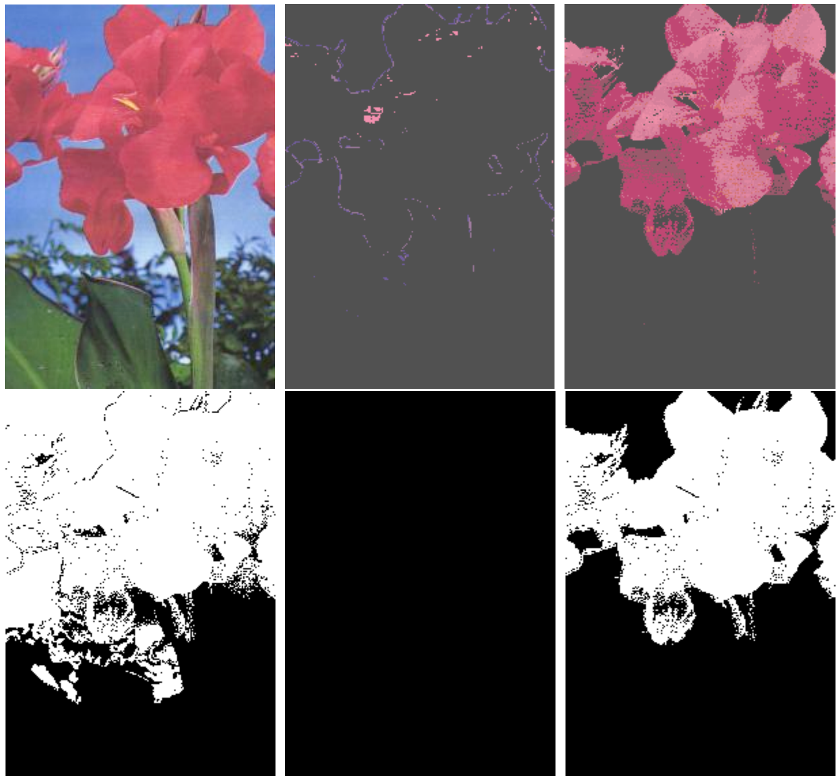


Figure 4.9. Recovery from erroneous deletion of background colors : (First column) Original image and segment found after deleting non-flower colors (Second column) Result of deletion of the color classes *blue* and *red* which were hypothesized to be background colors. No segment passing the minimum size criterion was detected. (Third column) Trying color deletion one at a time starting with the largest border color *blue* and the valid segment obtained as a result

which are hypothesized to be background colors based on the border block analysis, are eliminated. The validity of the largest segment obtained in the remaining image is tested to determine whether the choice of background colors was correct. If a valid segment is obtained, this is output as a flower region. Figure 4.8 shows some examples of the final flower segment obtained when the color classes correctly hypothesized to be the background (*white* and *blue* respectively for the first and second row in Figure 4.8) were deleted.

This method of detecting background colors is not guaranteed to produce correct results. It will fail for images of type **(c)** (shown in Figure 4.1(c)), and may also fail for images of type **(b)** if there is sufficient overlap between the flower and the margin.

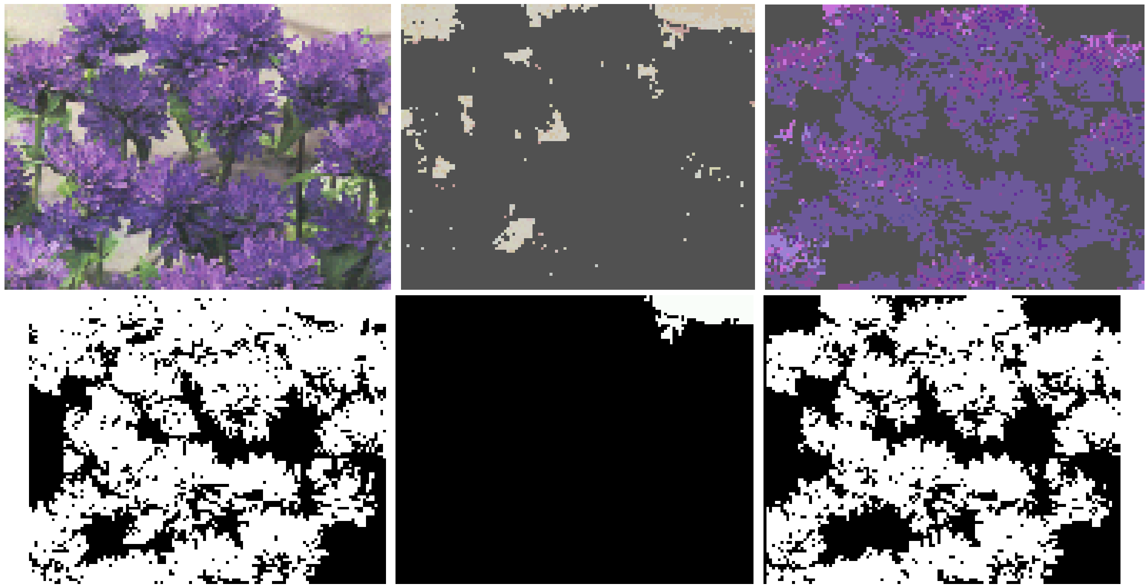


Figure 4.10. Recovery from erroneous background color selection : (First column) Original image and segment found after deleting non-flower colors (Second column) Result of deletion of the color class *purple* which was hypothesized to be a background color and the largest segment obtained (which is not valid since its centroid is in the boundary region) (Third column) Trying the new hypothesis that the color white is the background color and the valid segment obtained

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 could be 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. The background colors are arranged in the order of presence in

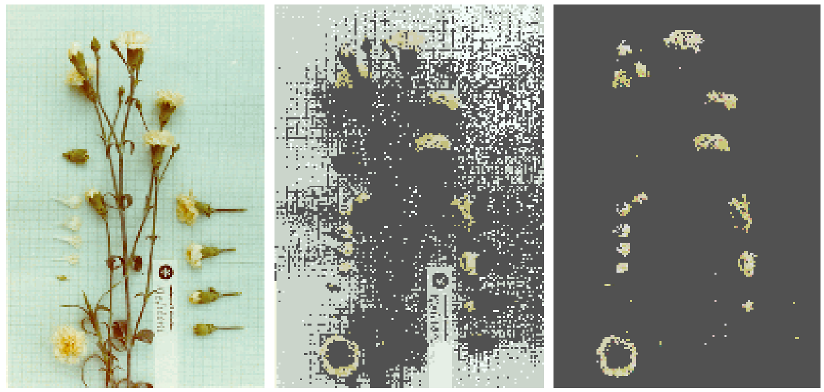


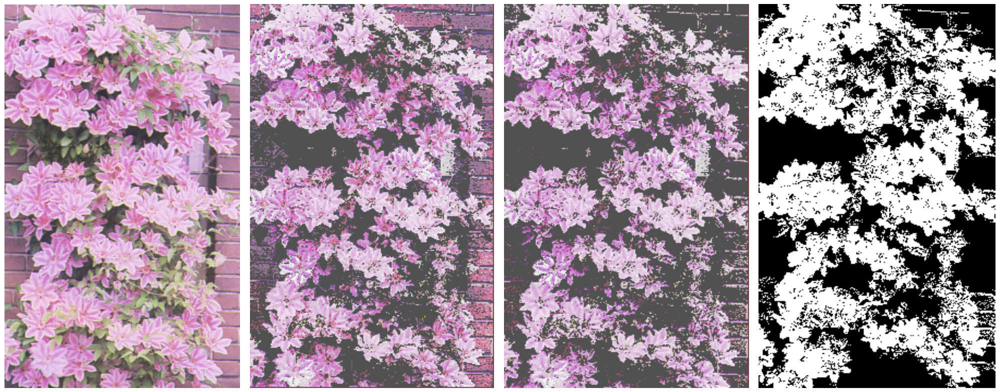
Figure 4.11. Using color names for labeling : (a) Original image (b) image left after deleting non-flower colors (c) result of eliminating background colors based on color *names*

decreasing number of border blocks during testing i.e. colors most dominant along the periphery are deleted first.

An example where each background color needs to be tested separately to get a correct flower region is shown in Figure 4.9. Here, the hypothesized background colors are *blue* and *red*. When both these color classes are deleted, there are no segments left. However, when the image is restored and only *blue* is deleted (blue is selected first as it is more dominant along the periphery than red), we get a valid flower segment.

It is possible that the first selection of background color during the iteration process is erroneous and we need to backtrack to a different hypothesis. Figure 4.10 shows an example of recovery from an incorrect background selection where the image is of type **(b)** but the flower color is more predominant along the periphery than the background. The color class *purple* is eliminated first as a possible background color. 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 color 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 type **(c)** and the

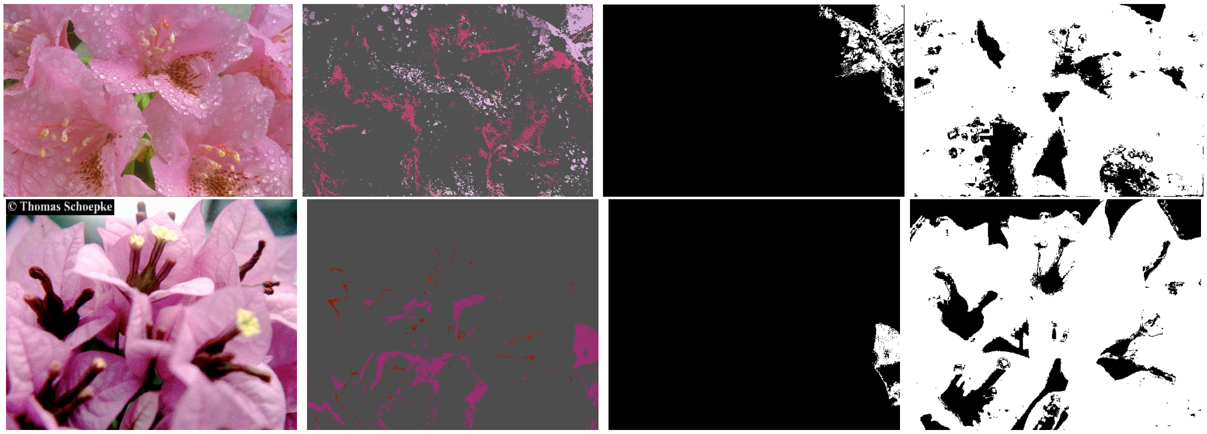


(a) (b) (c) (d)

Figure 4.12. Another example of use of color names for labeling : (a) Original image (b) image left after deleting non-flower colors (c) remaining image after eliminating background colors based on color *names* (d) final flower segment obtained

flowers cover the full image. However, since we are looking at color *classes*, there is an alternative situation (though uncommon) 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 where color name-based labeling is necessary to remove background elements is shown in Figure 4.11. 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 Figure 4.11(c) produces a valid segment which does not include any background. Figure 4.12 shows another example. In this case, the wall in the background and the flower are both



(a) (b) (c) (d)

Figure 4.13. Detecting an absence of background : (a) original images (b) image left after deleting non-flower colors and hypothesized background color (c) largest segments obtained from remaining image. Note that both these segments are invalid since their centroids lie within the image boundary region. (d) segment used for flower color determination.

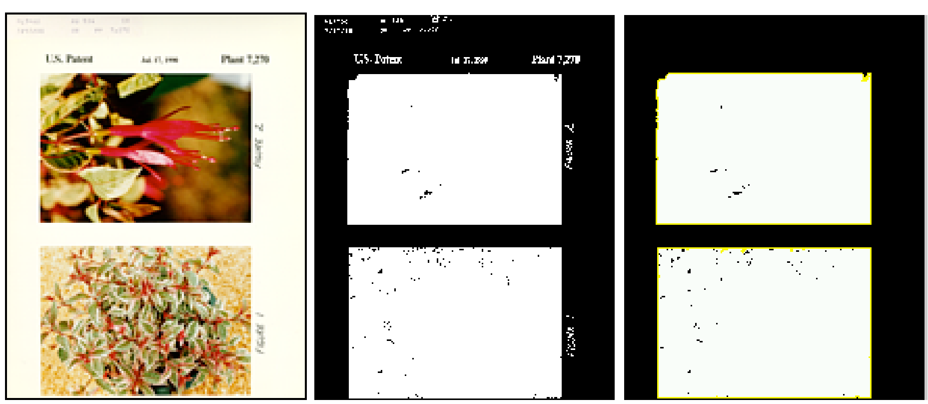
of color class *pink*. A final segment which includes very little background is obtained when background colors are deleted based on color names rather than color class.

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. Figure 4.13 shows two examples of images of type **(c)** where no background removal was possible. Figure 4.13(c) shows the segments obtained (which are not valid) when the hypothesized background colors are deleted. Figure 4.13(d) show the final output segment which consists of the image after the non-flower colors are deleted.

The segmentation strategy is likely to produce 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.

4.4 Test database

A database of flower images was constructed to test the automatic segmentation of flower regions and subsequent indexing and retrieval based on color indexes generated from the segmented regions. The test database consists of about 1300 images. Out of these, about 100 are actual flower patents provided by the U.S. Patent and Trademarks Office. We have added 100 images from CD-ROM collections, and another 100 images were scanned from photographs and pictures of flowers hand-picked by us as being similar in characteristics to the images from flower patents. These 300 images constitute the part of the database which meets all our assumptions about the type of images we expect our algorithm to encounter. The remaining 1000 images were downloaded from the world-wide-web using a web crawler. The wide variety and complex backgrounds in these images are intended to test the failure points of the segmentation algorithm. Many of these images violate one or more of our assumptions - flowers may be small, there may be significant non-uniform background or multiple types of flowers may be present. However, we still expect to produce good flower segments on most of these images or at least eliminate major background components.



(a) (b) (c)

Figure 4.14. 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 Figure 4.14(a), containing both text and images. Images were 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 Figure 4.14(c). These segments are approximated by rectangles and the cropped image corresponding to each segment is added to the database.

4.5 Segmentation results

The results of the proposed automatic segmentation algorithm were evaluated by viewing the output segments produced on images from the test database, and comparing them against the original image which provides the ground truth.

Source of images	Patent/CDROM/scanned	World-wide-web
Correct segmentation	93%	86%
Some background	6%	4%
Wrong segmentation	1%	10%

Table 4.5. Results of automatic segmentation on flower images

Table 4.5 shows the tabulated results for the test database. The results on the database are broken up into two components based on the source of the images. This is done to distinguish between the results of automatic segmentation on ideal and less-than-ideal images. In each of these categories, the table shows the percentage of images which produced correct segmentation (final segment contains flower regions only, no background), partially correct segmentation (final segment contains flower regions and some background) and incorrect segmentation (final segment either contains no flower regions or flower regions occupy a very small fraction of the final segment). Examples of each type of segmentation and the causes of failure are discussed in the rest of this sub-section.

Figure 4.15. Some examples of images where a correct flower segment was obtained by the iterative segmentation algorithm

Figure 4.15 shows examples of images from the world wide web where automatic segmentation produced perfect results. The final segments obtained (shown on the right) were solely from flower regions, and no background was included. The examples illustrate the wide variety of backgrounds and the large variations in the area covered by flower regions that the automatic segmentation algorithm can handle.

Figure 4.16. Some examples of images where the segment obtained does not cover a whole flower, but is sufficient for the purpose of flower color determination

It should be pointed out that a complete flower need not be present in the final segment for the segmentation to be useful. Figure 4.16 show some examples where the final segment covers some petals out of a complete flower. Since all the petals are of the same color, the color description obtained from the partial flower is the same as when the whole flower is considered. Therefore, the segmentation is still considered to be a success.

Partially correct segmentation results constitute those images where the final segment includes the flower and some background. Even though all background could not be eliminated automatically, the color indexes generated from the segmented images are still more representative of the main subject than if no background elimination was performed. Some examples of such images are shown in Figure 4.17.

There were just two images which produced erroneous segmentation in the flower patents/CDROM/photos database. These images are shown in Figure 4.18. In the first image, which is a part of a patent application, the white pot forms the largest segment, rather than the flower regions. This situation is rare, and this was the only instance of the background element being more dominant than the flower in this domain. The second image was scanned from a photograph. In this case, most of the flower pixels were classified as *brown* and therefore, were eliminated as non-flower pixels. Mis-classification of flowers into non-flower classes is also encountered in some images from the web, and is probably caused by color shifts in the scanning/acquisition process.

The rate of wrong segmentation, where the flower itself is missing from the final segment, is rather high in the images downloaded from the web, and this needs further investigation. Table 4.6 shows the break-up of causes of failure of the automatic segmentation algorithm on these images. It can be seen that most of the flowers were missed because they were too small. Figure 4.19 shows some examples of this case. Since these images barely qualify as images of flowers, these failures are not siginificant. A few other flower images show a color cast which shifts the color of the flower into non-flower colors in color space. Some examples are shown in Figure 4.20.

Flower too small	60%
Flower color labeled as non-flower color	20%
Background segment found	20%

Table 4.6. Break up of images which generated incorrect segmentation based on the cause of failure

The third class of erroneous segmentation was caused by the presence of background and represent true failures of the automatic segmentation algorithm. These

Figure 4.17. Some examples of partially correct segmentation where the final segment (shown on the right) contains some background in addition to the flower regions. However, the background included does not dominate the flower region in the final segment, and a reasonable flower color description can be obtained

Figure 4.18. Images on which the segmentation algorithm produces errors: the image on the left is from a flower patent, the image on the right is scanned from a photograph taken by the author

Figure 4.19. Some examples of images (from the world wide web) where the flower was missed because the flower regions were too small compared to the image size

Figure 4.20. Some examples of images (from the world wide web) where the flower was missed because the flower color was classified as shades of green (top), brown (middle) or gray (bottom) and was therefore omitted from the segmented image. The images on the right show the remaining image after the non-flower colors were deleted

constitute 2% of the images in the world wide web image database. Some examples of this situation is shown in Figure 4.21.

Overall, the segmentation helps produce a more accurate description of the area of interest (flower regions) in the images in more than 90% of the cases and therefore, is a significant improvement over using the whole image for color-based indexing.

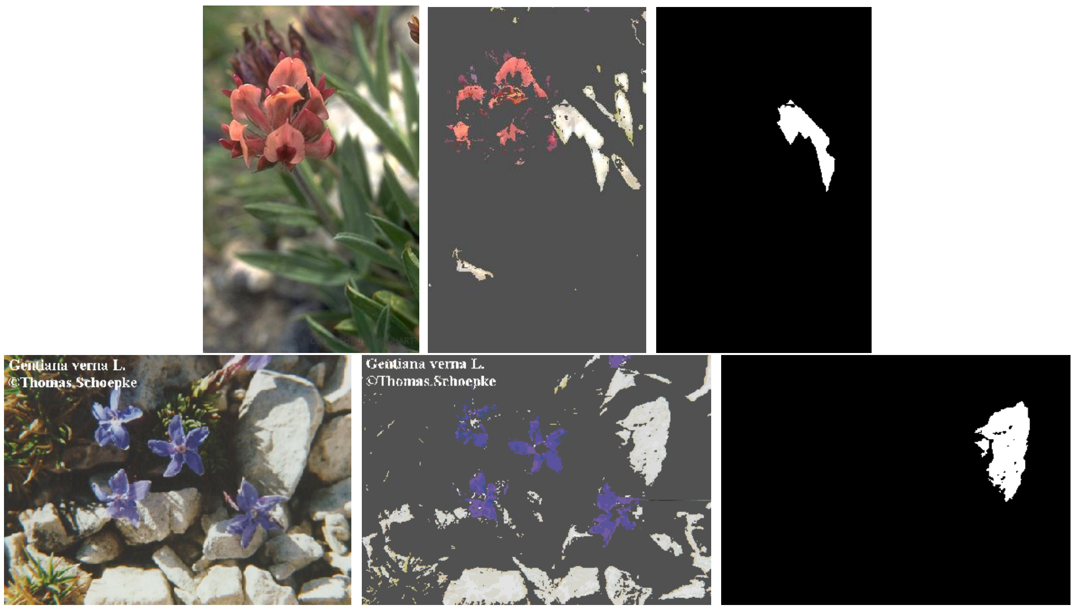


Figure 4.21. Some examples of images where the flower was missed because the flower region was smaller than a background segment which could not be removed. (left) original image (middle) image after deletion of non-flower colors and any detected background colors (right) final segment

4.6 Indexing and Retrieval

Color information is extracted from the segment identified as a flower region in the earlier section, to be used as features during retrieval from the flower image database. The flower database indexing is based on the types of queries we would like to support. This includes queries using color names, color classes and example 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 of the flower. So, in addition to the presence of particular color names, the relative proportions of colors in the flower region is also an important factor to be considered.

4.6.1 Query by name

The color names defined in X are used as keys for color-name-based indexing. In addition, an index table is also generated to access the images by the color classes present in the images. When a color name is provided as query, the X name index is searched for the query color name and its variants. The variants are included since the X naming system 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 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.

4.6.2 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.

Therefore, 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 $(\frac{p_1h_1+p_2h_2}{p_1+p_2}, \frac{p_1s_1+p_2s_2}{p_1+p_2}, \frac{p_1v_1+p_2v_2}{p_1+p_2})$. 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.

4.7 Retrieval experiments

We tested the retrieval results obtained using 50 queries of different types. On 25 queries using color names, we 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 [78] obtained is shown in Figure 4.22. The average precision obtained was 88% and the precision at 100% recall was 66%.

Figure 4.23 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. The retrieved images are

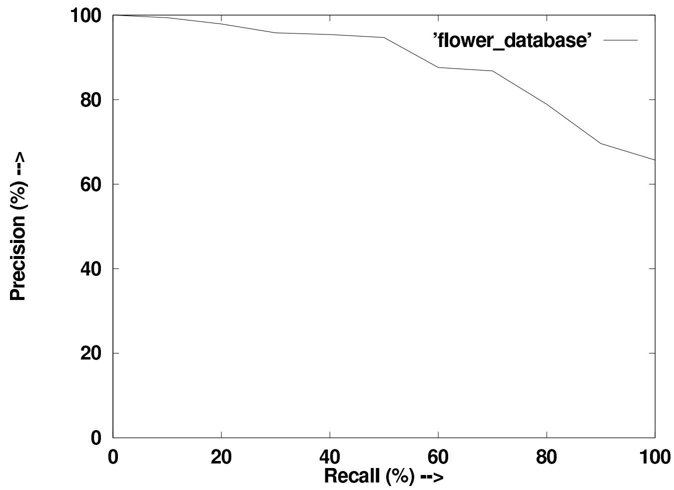


Figure 4.22. Recall-Precision graph for 25 queries by example on the flower patent database

displayed at the bottom of the interface. Figure 4.24 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. The example image selected is displayed in the right frame and the retrieved images are displayed at the bottom.

Figure 4.25 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.

4.8 Conclusion

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 im-

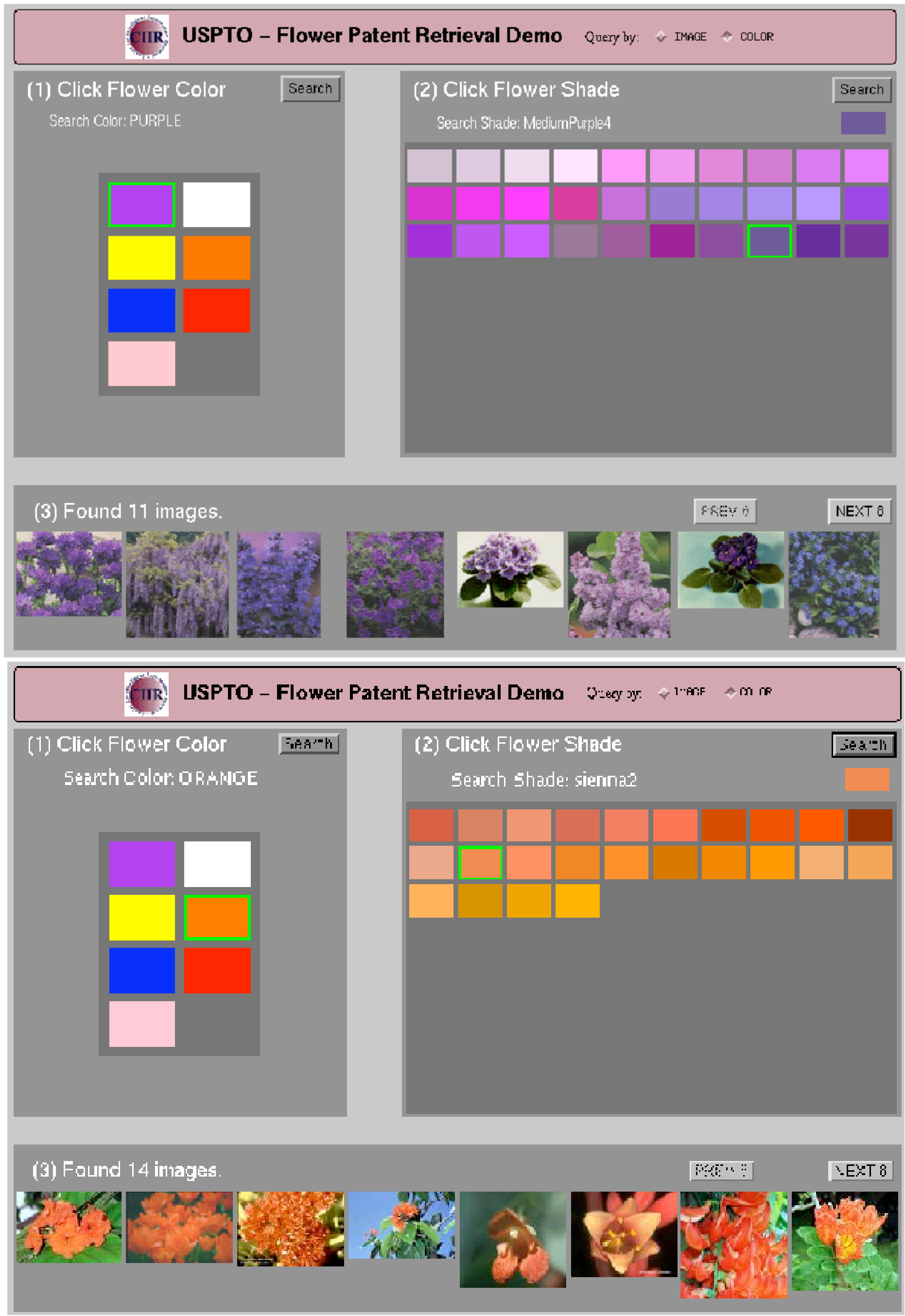


Figure 4.23. Retrieval by color name : The color shade selected here are ‘medium purple’ (top) and ‘sienna2’ (bottom)

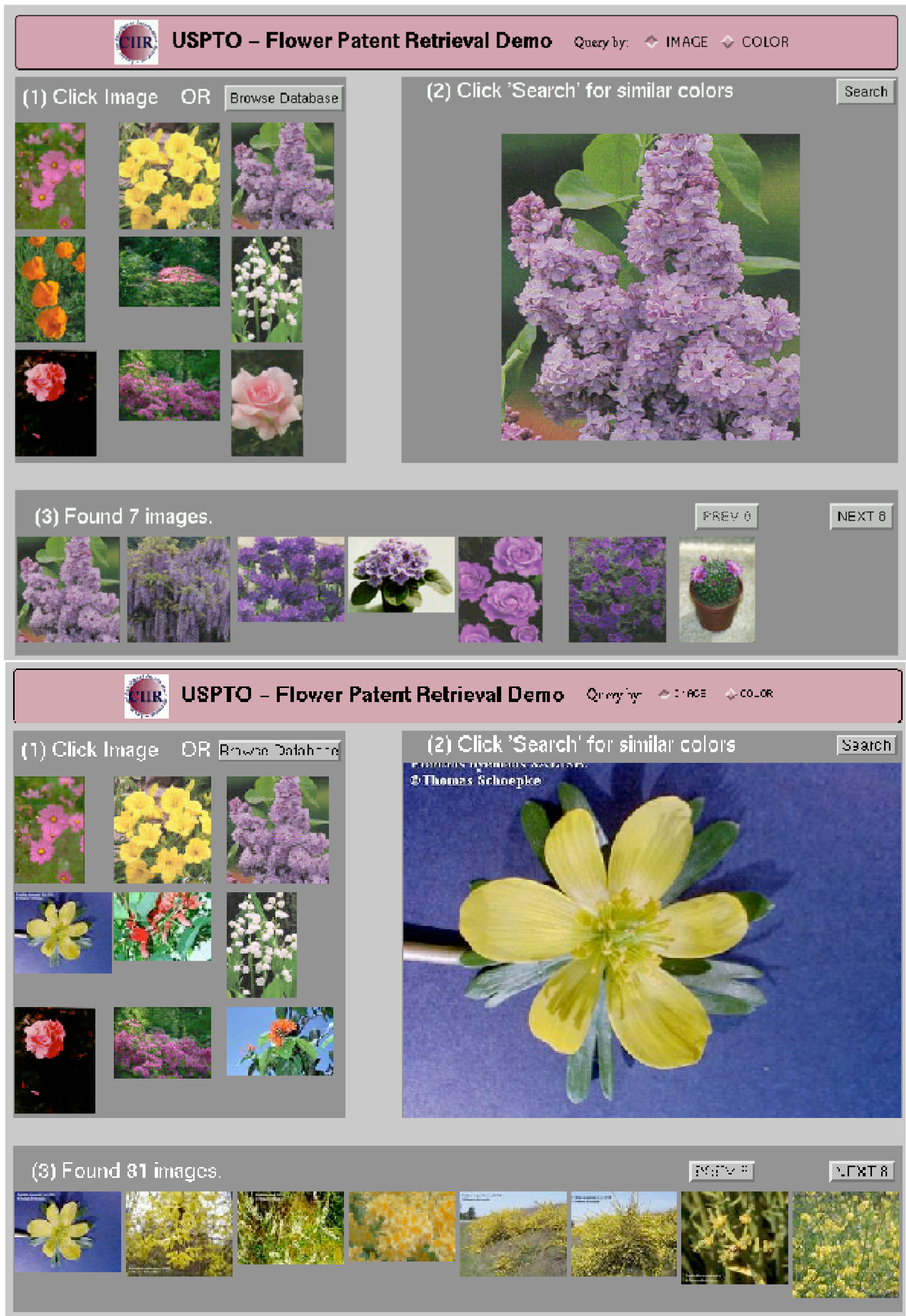


Figure 4.24. Retrieval by example : The query selected is shown on the right.

Figure 4.25. 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'

age 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.

The core contribution in this chapter is the automatic flower segmentation algorithm. The flower region is isolated from the background by progressively eliminating background elements. The domain knowledge provides the necessary feedback to complete the loop. The color of the flower is defined by the color names present in the flower region and their relative proportions. The test flower 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.

CHAPTER 5

INDEXING A DATABASE OF BIRD IMAGES

5.1 Introduction

In the previous chapter, we have addressed the problem of extracting the object of interest (flower) automatically and indexing the database based on features gathered from the object only, thus eliminating the effect of the background and providing more meaningful retrieval results. The proposed solution used domain knowledge specific to the subject (flowers) and the application (flower patents). However, there is a large number of image databases dedicated to specific subjects, where there may not be any easily identifiable subject-specific domain knowledge available which can be used for segmenting the subject from the background. These databases are usually characterized by images which portray a single object which can be clearly identified by a human user, the challenge is to extract the object of interest automatically. This work is motivated by the need for such an object-of-interest finder as a preprocessor to any indexing and retrieval system which will be working on a database of images with clearly defined subjects.

Segmentation of an image into its constituent objects is a very difficult and ill-defined problem. In addition, the solution to the given problem also requires the discrimination of foreground object(s) from background elements. One approach used for face detection is to train a classifier using a large number of examples of images of the object [61, 64]. This strategy works well because human faces are structurally similar. In other databases, images of birds for example, the large differences in appearance between different subjects within the domain and the variations in appearance due to change in 3D viewpoint imply that this approach would be very difficult to implement.

Though this work is related to the problem of image segmentation, there are notable differences because of the difference in the final goals. In our case, the primary goal is to identify a region in the image which will produce features derived from the subject only, enabling image indexing and retrieval based on the subject of the image. This does not require perfect segmentation of the subject, as long as the region considered is predominantly covered by the subject. The final segment may have small parts of the bird missing or include small areas from the background without much impact on retrieval performance. Also, we are not interested in segmenting the background correctly e.g. sky and foliage in the background can all be treated as one "background" mass.

We show that we can use general characteristics of photographs of single objects to propose an approach to automatic segmentation for finding the figure or subject of interest. For example, we observe that for aesthetic reasons, photographers try to ensure that the subject of interest is "prominent" and that the background is less prominent. This is usually done by placing the subject closer to the center of the image, by making the subject of interest larger than other objects in the image and by having the subject in sharper focus than the background. The databases we are interested in retrieving from , for example, pictures of birds, flowers, or other animals often have these characteristics.

Our approach involves eliminating the background, leaving the part of the image most likely to be the figure or object of interest, as in the previous chapter. The primary differences between these two pieces of work stem from the amount of available domain knowledge and the generality of the assumptions made. In the previous chapter, we provided a solution to the problem of object-of-interest identification on a database of flower images. In that case, we depended heavily on domain knowledge available about the color of flowers (e.g. flowers are rarely gray, brown, black or green) and the type of images (submitted in applications for flower patents). A database of birds, on the other hand, has no particular domain specific knowledge that can be exploited. The problem is made more difficult by the fact that most birds have evolved to merge into their natural backgrounds to avoid detection by predators, unlike flowers which are designed to stand out against their background to attract pollinators. In this case, we do not use any characteristics specific to birds. Indeed, we show some examples where the object of interest is correctly detected in domains other than bird images.

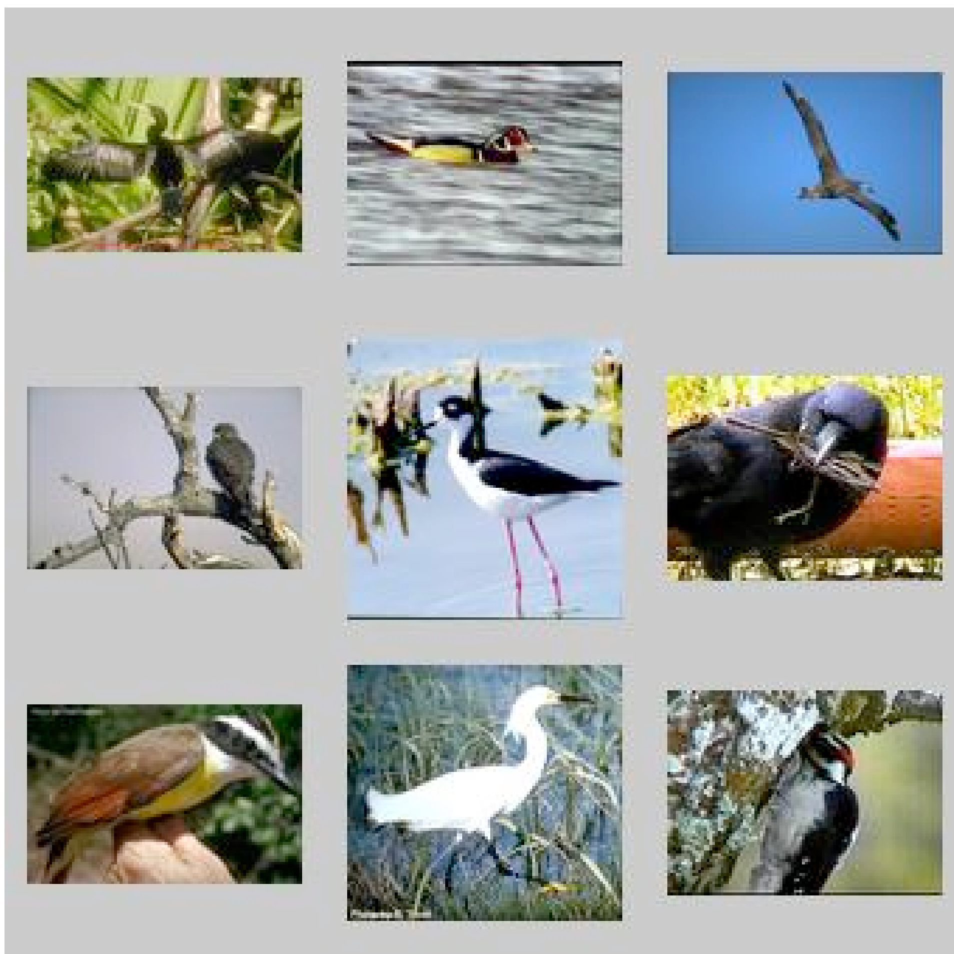


Figure 5.1. Some images in the bird database

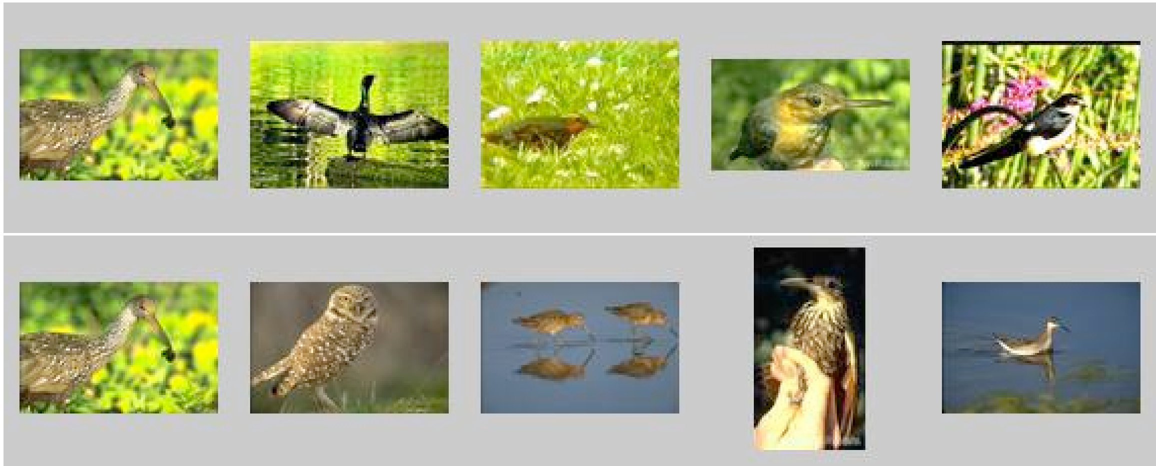


Figure 5.2. Qualitative improvement in retrieval obtained when only the bird region is used for indexing. The query is the leftmost image. (top) whole image color-based retrieval (bottom) retrieval after indexing only the colors from the object of interest found by the method described in this chapter

The result of our work is illustrated by examples from a database of images of birds. These images were downloaded from the world wide web and show wide variations in the type of background (water, sky, ground, man-made surroundings) as well as the size of the object of interest as shown in Figure 5.1. Figure 5.2 shows an example of the top five retrieved images (with the query image ranked first) when the color signature from the whole image is used for retrieval and when only the colors present in the region of interest are used for indexing. The query image shows a brown bird with white spots. When the whole image is used as a query, it is clear that the green and yellow background plays a prominent role in the retrieved images, since the second and fifth ranked images show a black-and-white bird, and the third and fourth images show birds which are brown but without any white coloration. Clearly, these birds do not represent the color composition of the query bird. When only the region of interest computed is used for indexing, birds with colors similar to the query are retrieved in a variety of backgrounds. All the retrieved images show birds which are brown and white, and thus relevant to the colors of the queried bird. If there was another image of the same bird in the database (which there was not), it would have a high probability of matching the query even if the background was different. It is to be noted that using color alone cannot ensure that the same species of birds are retrieved, and these results represent the best that can be achieved with color alone.

This chapter is organized as follows : section 5.2 discusses the detection and elimination of background colors based on a combination of color analysis and edge information. Section 5.3 discusses experimental results on segmentation and subsequent indexing and retrieval, with section 5.4 containing concluding remarks.

5.2 Detection and elimination of background

The strategy for background elimination is based on color and edge information combined with rules derived from general observations about photographs of single subjects. The observations are also used for evaluating the likelihood that a segment could be the object of interest.

5.2.1 Observations about photographs

The specific observations we exploit are derived from general rules-of-thumb followed when photographing a subject. Since no domain-specific assumptions are made, these observations are true of most images with clearly defined subjects. The subject is usually centered in the middle three-quarters of the image (defined as the "central region" in Figure 4.4) and occupies a reasonable portion of the image. When photographing a specific subject, there is usually an attempt to keep other competing foci-of-interest out of the picture. For example, the subject is often in sharper focus than the background. Further, a picture of a parrot and a sparrow has two subjects, unless one is clearly larger and more in focus than the other. In such cases, we assume that the larger region is more significant and ignore smaller regions.

Based on these observations, we know *a priori* that we are looking for a segment in the image which is large enough, is centered somewhere in the central region of the image and has prominent edges, since it is in focus. Conversely, the background regions surround the main subject and thus, are more likely to be visible along the periphery of the image. If the background is out-of-focus, there may not be significant edge information detected in that region. However, none of these observations are true in *all* cases. In such cases, it may not be possible to discriminate between the foreground and background of the image in the absence of additional constraints. The design of our algorithm takes this possibility into account, and produces no segmentation where good subject extraction is not possible based on the color and edge information gathered from the image. In the context of image retrieval, this would mean that the whole image is used for indexing, which is the starting point we are trying to improve on.

5.2.2 Segmentation strategy

Our approach to elimination of background based on color entails the generation of a hypothesis identifying the background color(s), elimination of those colors and checking the remaining image for the presence of a valid segment. The check provides a feedback mechanism for background elimination which indicates whether the hypothesis was correct or a new one needs to be formulated. Figure 4.2 from the previous chapter illustrates our approach. Since no color-based domain knowledge is available and birds tend to blend with their background colors, additional information is necessary to produce sufficiently accurate segmentation. Thus, the remaining image after elimination of detected background colors is combined with information from an edge description of the image which captures the major structures present in the parts of the image that are in focus. The final result is a segment containing the object (figure) region. The outline of the algorithm used to produce a segment from which the color of the bird can be estimated is shown in Figure 5.3. The elimination of background color is described in this sub-section and the incorporation of edge information is discussed in the next sub-section.

The first step in producing a list of possible background colors is to select a suitable color space to label the image pixels. The RGB space in which the original image is described, has too many colors to be useful. As before (sub-section 4.3.1.1), we use the colors defined by the X Window system which has only 359 colors and is also perceptually grouped into visually distinct colors. Since the mapping from the RGB space to X Color names is sparse, for points with no exact map the nearest color name (by city block distance) is used to map the point to a color defined in X. This mapping both reduces the number of colors and also ensures that small variations in the color of an object are classified as the same perceptual color. The multi-tiered ISCC-NBS naming system used for the flower database is not necessary in this work, since the grouping of finer color descriptions into color classes leads to colors that are too general to be useful in the bird domain. Since birds tend to be camouflaged against their backgrounds, the broad color class describing the bird and the background are often the same.

The presence of background colors is detected by analyzing the color composition of the image margins. The margins of the image are divided into border blocks which are narrow rectangles as shown in Figure 4.4. In this case, we use the complete image periphery (all four sides) and divide the periphery into 24 equal border blocks, whereas the bottom edge of the periphery was ignored in the case of flower images. The distribution of X colors in these blocks is computed and colors present in more than one border block are marked as possible background colors.

After eliminating all the pixels of the hypothesized background color(s), the largest segment in the remaining image is computed. We use the *connected components* algorithm for identifying segments in the image, where each segment is a connected component. The connected components algorithm is run after *binarizing* the image,

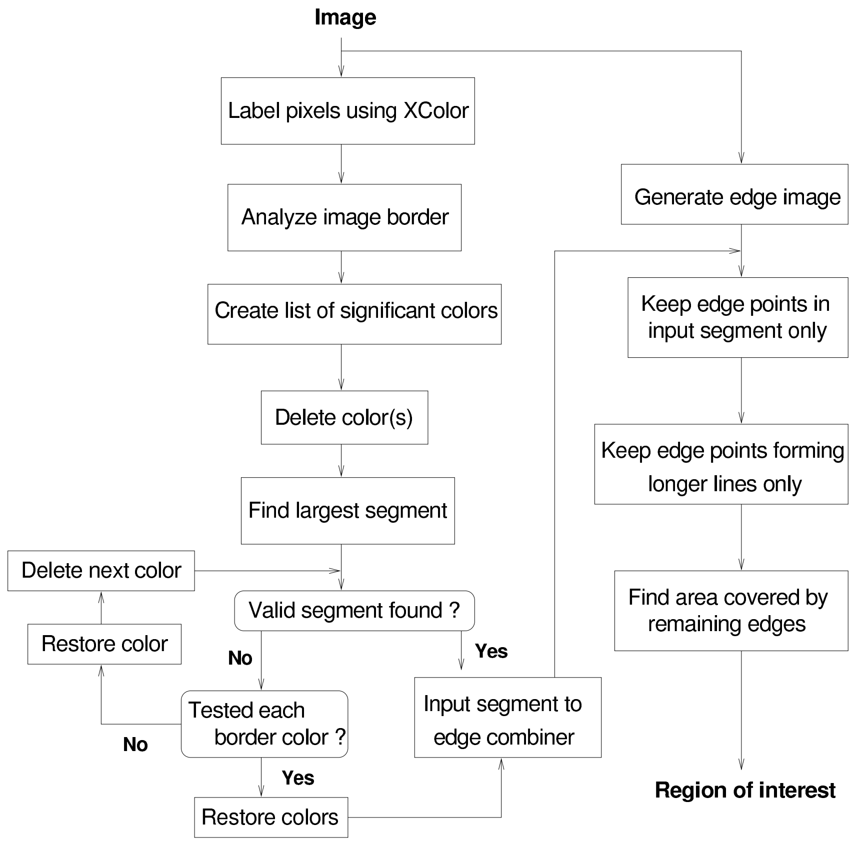


Figure 5.3. Overview of segmentation strategy

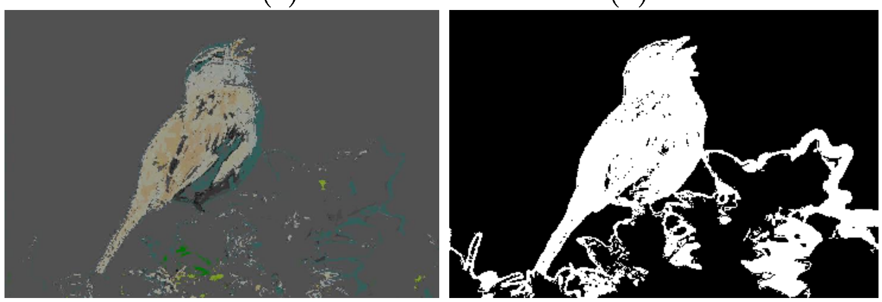
where the only two classes are pixels which have been eliminated and those that remain. Figure 5.4 shows an example of the largest segment obtained when the colors detected along the periphery are deleted after being identified as background colors (this segmentation is further improved by inclusion of edge information). Some examples where the largest segment obtained closely matches the bird region of the image are shown in Figure 5.5.

We use two criteria for evaluating whether the segment produced is valid; its size and the location of its centroid. As discussed in the previous sub-section, the segment cannot be a possible candidate for the subject of the image if it is too small or if its centroid falls in the boundary region of the image (as defined in Figure 4.4). Examples of segments that are correctly flagged as invalid are shown in Figure 5.6. A lack of valid segments after elimination of the hypothesized background colors, is

(a)

color	no. of blocks
SteelBlue3	**14**
olive green	**10**
cornflower blue	**8**
SteelBlue2	**4**
DarkSeaGreen4	**4**

(b)



(c) (d)

Figure 5.4. Background elimination : (a) original image (b) significant colors detected along image periphery (c) image left after deleting colors in *(b)* found along the image periphery (d) largest segment obtained from *(c)*

an indicator that the background color selection was wrong. A detailed description of the rules derived about the location of background colors and the characteristics expected in the final segment can be obtained from section 4.3.2.2 of chapter 4.

When there is feedback that the background color chosen was incorrect, the color(s) is restored and each color present in the image periphery is tested separately as a potential background color. If no valid segments are found when any of the colors present in the border are eliminated, we can conclude that the bird and the background cannot be differentiated based on color, and the whole image is output as the segment of interest. This happens when the background color and the color of the bird match. Figure 5.13 show two examples of this case, which is not uncommon in this database because many birds depend on camouflage to remain undetected.

In some images, background color deletion is sufficient to produce a good segmentation of the bird from the background as shown in Figure 5.5. In most images,

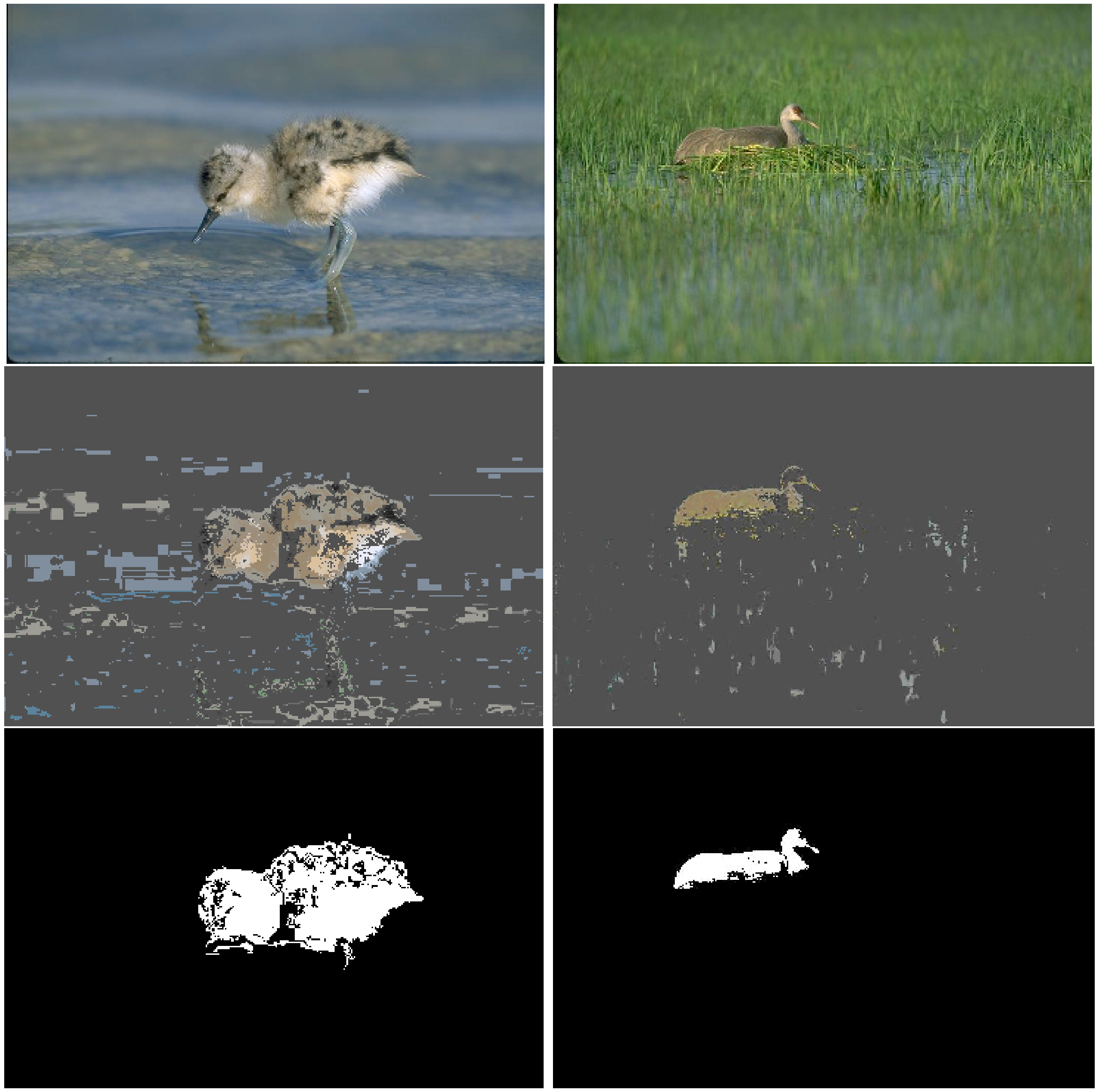


Figure 5.5. Examples showing extraction of bird segment obtained where the background color elimination step is very effective : (row 1) original images (row 2) image after deleting background colors (row 3) largest segment produced

however, the output can be further improved by additional processing as described in the next section.

5.2.3 Using edge information

It is not always possible to extract a segment containing only the bird on the basis of differentiation of background and bird colors. In addition to images where the color

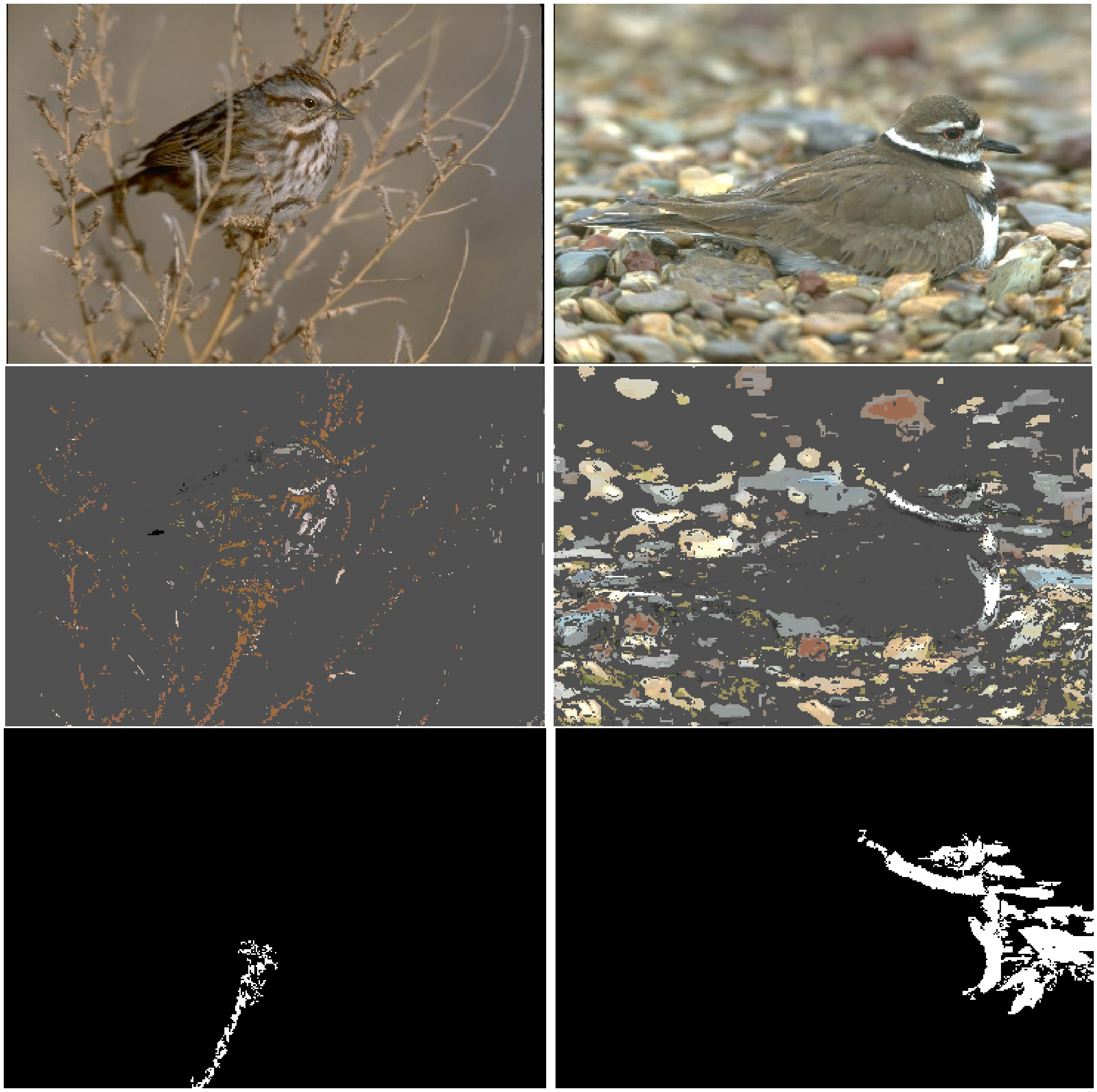


Figure 5.6. Examples showing detection of invalid segments : (top) original images (mid) after deletion of hypothesized background colors (bottom) largest segments produced (invalid since too small (left) or centroid is in the image boundary region (right))

of the bird closely matches the background colors, there are images where background colors remain because they were not present along the image periphery, and therefore, were not detected by the background elimination process. Edge information can be used in many cases to refine the segmentation.

There are differences between edges associated with the outline of the bird and edges contributed by other parts of the image. The edges associated with the background are usually present only at smaller scales. This is due to several reasons:

- The background often consists of uniform regions such as sky in which edges, if any, appear only at the smallest (finest) scales.

- The background may often be blurred (for example, the top left images in Figure 5.7 and Figure 5.8) because of the limited depth of focus of cameras, an effect that is often accentuated by the photographer. In this case too, there are no strong edges at larger scales.

- Many backgrounds associated with bird images consist of textured surfaces such as grass, mud, water or trees. The scale of such textures is usually much smaller than that of the bird.

In contrast, the bird is usually large and distinctive in the image. Thus, the edges associated with it are usually present at a wide range of scales. It is to be noted that the edge structure of the internal feathers of the bird is often present only at small scales. However, this does not matter for our purposes since we are only interested in the external contour of the bird.

These effects can be taken advantage of in eliminating background regions by using a relatively larger scale for detecting edges. Thus, only edges present in the bird's contour would be detected. The main steps in computing an edge image are listed below :

- The image is convolved with the two first derivatives of a Gaussian [26] to include both vertical and horizontal edge directions. The derivatives of Gaussians are energy normalized (by dividing by the scale).

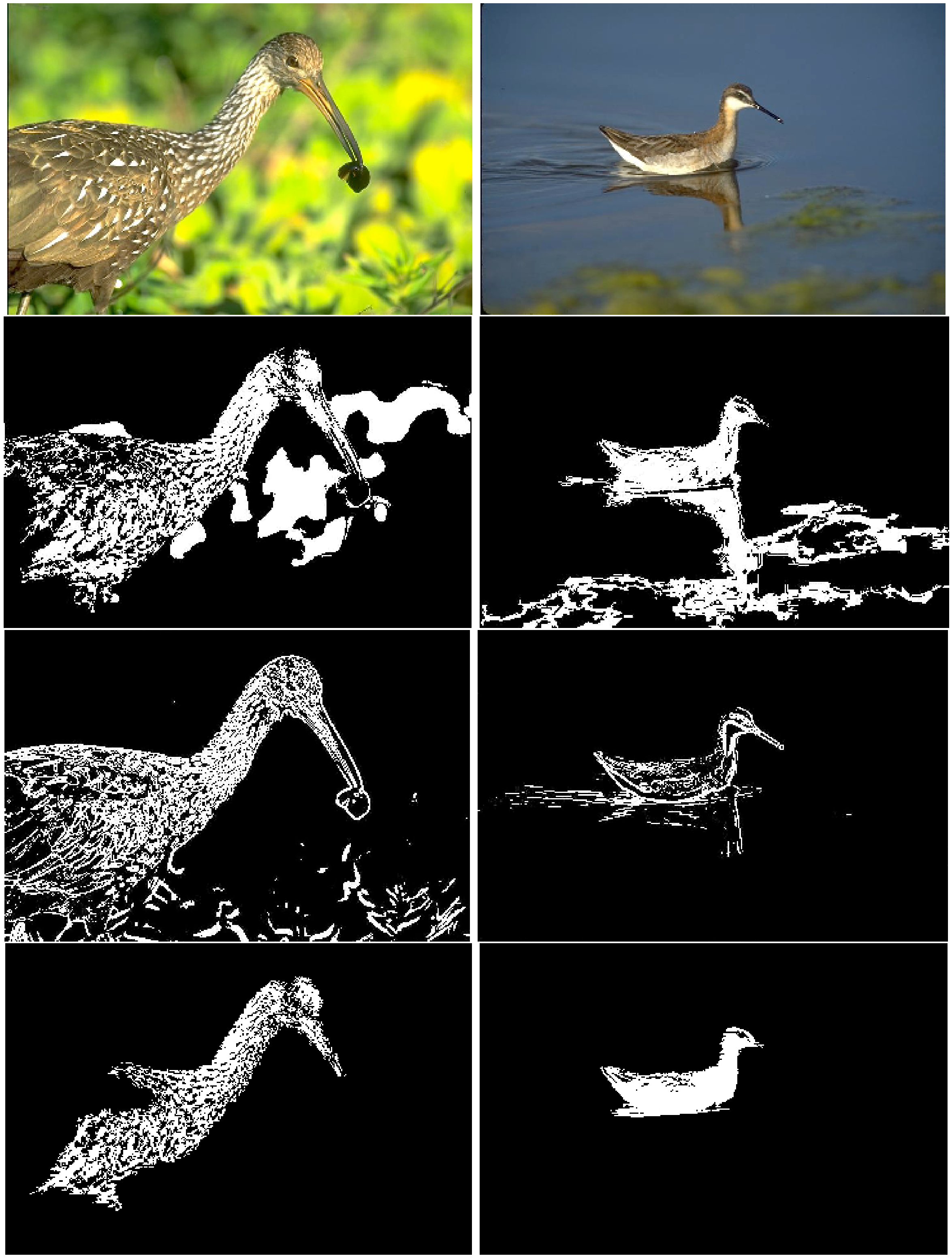


Figure 5.7. Examples showing improvements in the bird segment extracted when edge information is incorporated : (row 1) original images (row 2) largest segment after background color deletion (row 3) edge image (row 4) final output

Figure 5.8. Examples of region of interest segmentation : (row 1) original images (row 2) remaining image after background colors are eliminated (row 3) edge image (row 4) final segment obtained

- The derivative outputs are combined to produce the gradient magnitudes. The energy normalization ensures that the range of the gradient magnitude images is roughly the same at all scales.

- The output of the image is then thresholded to find edges. We have found that a scale of $\sigma = 2$ and a threshold of 15 works for all our images.

The third row in Figure 5.7 shows the output of the edge detector on the bird images in the first row. Note that large portions of the background do not have any edges present while the edges on the bird are still present. It is clear from the image on the right side that the edge image alone is insufficient for eliminating the entire background and the combination of edge and color information provides improved background elimination.

5.2.4 Generation of final region of interest

The inputs to this system are the edge image and the segment of interest output by the color-based background elimination process. The segment of interest based on color and the edge information present in the edge image need to be combined to arrive at a final region of interest. The combination process places higher confidence on the color-based background removal, since the edge-based background elimination is effective only when the conditions described in the previous sub-section are satisfied. Often, edges from large structures in the background or smaller structures close to the bird (and therefore, in sharp focus) are present in the edge image generated. The main steps involved in this process are listed below and illustrated using the second bird image in Figure 5.8.

- The edge pixels that are not included in the color-based segment of interest are eliminated. For example, when the edge image (Figure 5.9(b)) is filtered using the color-based segment of interest (Figure 5.9(a)), edge pixels shown in Figure 5.9(c) remain. This should eliminate most of the edges from the background.

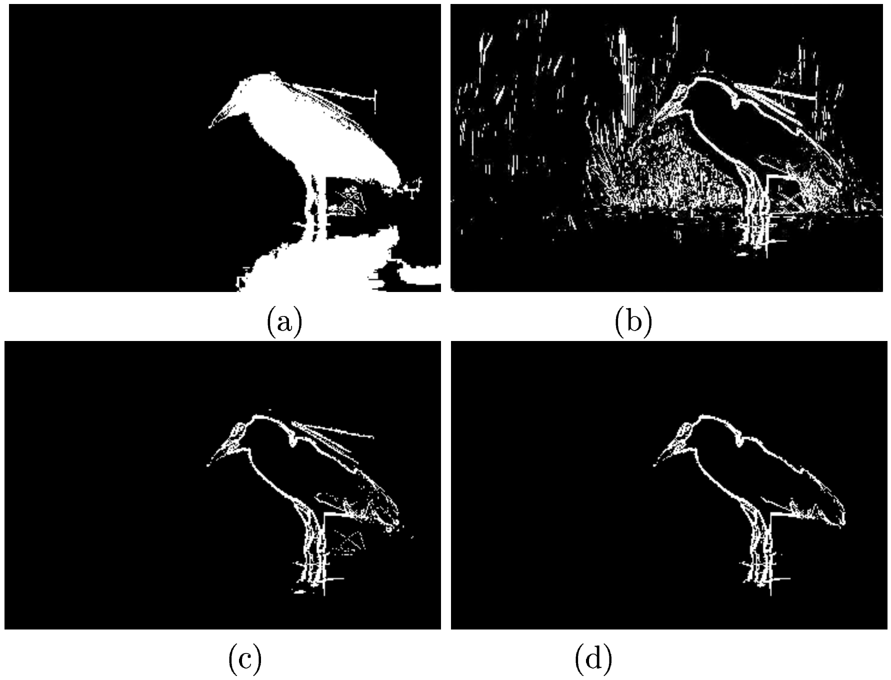


Figure 5.9. Combination of color-based segment of interest with edge information : (a) region of interest output from color-based segmentation (b) edge image (c) edge image left after deleting pixels which do not overlap with *(a)* (d) remaining edge image after small edge segments have been removed

- The next step finds connected components linking edge pixels into edge segments in the remaining edge image. Small and isolated edge segments are eliminated (edge segments containing less than 20% of the total number of edge pixels remaining are considered to be too small). This process leaves the longer edge segments only as shown in Figure 5.9 (d).

To estimate the area covered by these remaining edge lines, a closed contour is assumed and a commonly used technique from computer graphics is used to determine the inside/outside relationship [77]. The image is processed one scanline at a time and the region between the odd and even edge crossings on each scanline is included in the final output segment which represents the object of interest (bird) in the image. Figure 5.10 shows that the segment between the odd and even crossings represent the part of the scanline inside the object. The scanlines containing only one edge crossing

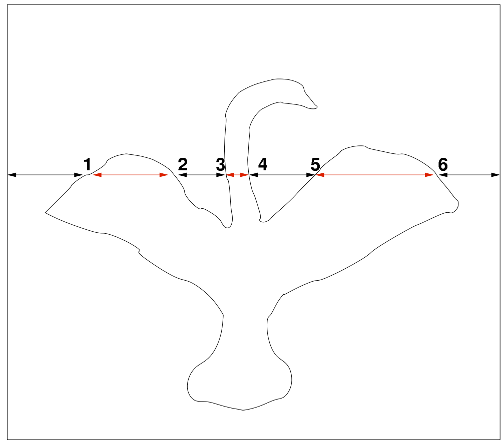


Figure 5.10. Example showing the edge crossings (numbered 1 to 6) on a scan line. Note that the parts of the line between an odd and even crossing are within the object, and the segments between an even and odd crossing are outside the object

are ignored, these occur when there are pieces of the background remaining or when the bird contour is incomplete. A reasonable bird region will be obtained even when some scanlines are missed if the contour of the bird is mostly detected correctly. Some examples where the edge information is able to improve the segmentation produced by color-based foreground-background discrimination are shown in Figure 5.7 and Figure 5.8.

5.3 Experimental results

The test database used for this work consists of 1200 images of birds downloaded from the world wide web. These images vary widely in quality – the resolution of the images varies from barely acceptable to very high and the photographs themselves range from professionaly taken to clearly flawed. The image sizes range from 12Kb to 40Kb. There is a wide variation in the type of background (water, sky, ground, man-made surroundings) as well as the size of the object of interest. Some examples of the database images can be seen in Figure 5.1.

5.3.1 Results of automatic segmentation of region of interest

The automatic segmentation results were manually verified [1] and divided into five classes as follows.

1. *No background remaining* : This class consists of images where the background is totally eliminated and the region of interest includes the bird region only. Some examples of this case is shown in Figure 5.8. In most of these images, the color of the background is different from the bird or the background is sufficiently blurred.

2. *Insignificant background remaining* : In these images, the greater part of the background is eliminated and the remaining background is not large enough to alter the color distribution of the final segment significantly. Some examples of this case are shown in Figure 5.11. In most cases, the background included consists of the object (branch, rock) on which the bird is resting, since this object is in as sharp a focus as the bird, and is often of the same color (to provide the bird with camouflage).

3. *Significant background remaining* : In this class of images, a significant amount of background remains in the final segment so that the color distribution computed for the bird is not accurate. Examples of such images is shown in Figure 5.12. The region of interest produced includes some large and prominent object(s) from the background in addition to the bird region.

4. *Image unchanged* : In images where the bird is well camouflaged, it is not possible to extract the foreground on the basis of color or edge information. In such cases, the whole image is used for indexing. Figure 5.13 shows some

[1] The manual verification was done on half (600) of the total images because of the labor intensive nature of the checking process.

examples of this case. Note that the background contains the same colors as the bird and is very cluttered (so the edge image provides no useful discrimination).

5. *Incorrect segmentation* : Figure 5.14 shows two cases where the segmentation algorithm failed; the bird was eliminated altogether and the output consists of parts of the background. This happens when the main background color matched that of the bird, but there were other background colors in the central region of the image occupying a significant area.

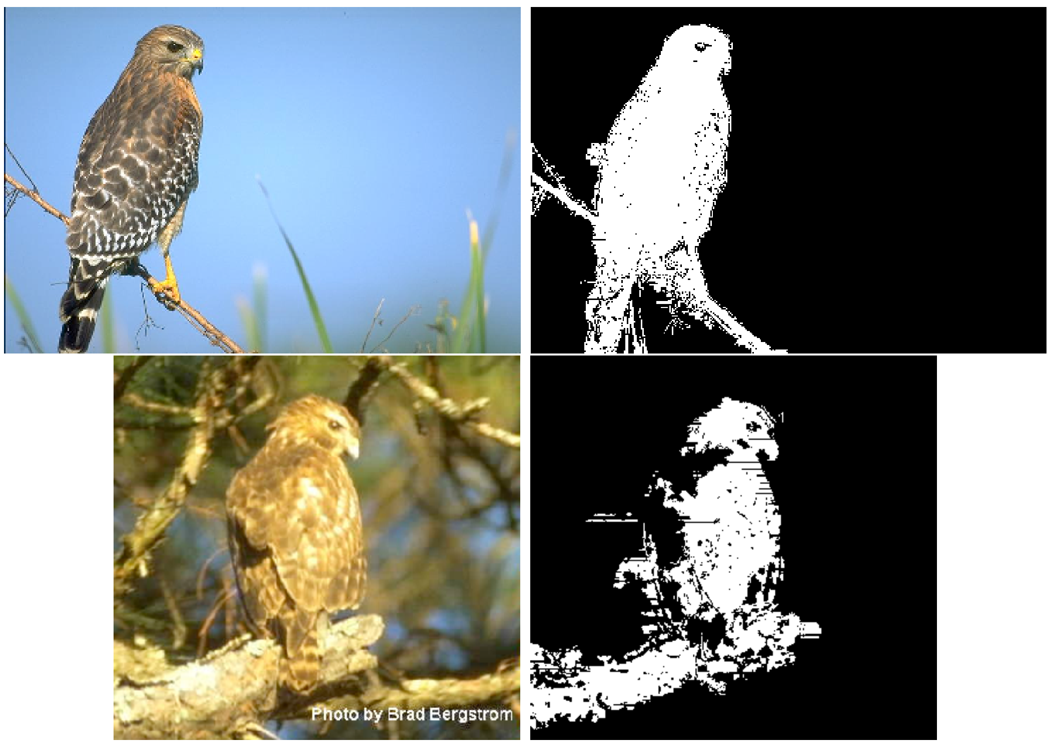


Figure 5.11. Examples showing partial elimination of background where the included background does not affect the color distribution of the final segment significantly.

The percentages of images falling under each of the above classes is listed in Table 5.1. Since segmentation acts as a pre-processing step to indexing and retrieval, the success of the segmentation can be judged by the impact it has on the image retrieval problem. The following discussion lists the implication of the segmentation obtained for image retrieval.

Figure 5.12. Examples showing partial elimination of background where the included background does affect the color distribution of the final segment.

Figure 5.13. Examples showing cases where a valid bird segment could not be extracted based on color

1. The first class of images (no background remaining) is the ideal case, where only the colors of the bird is used for indexing. In this case, the retrieval results are based on the color of the bird alone, and should closely match the user's expectations.

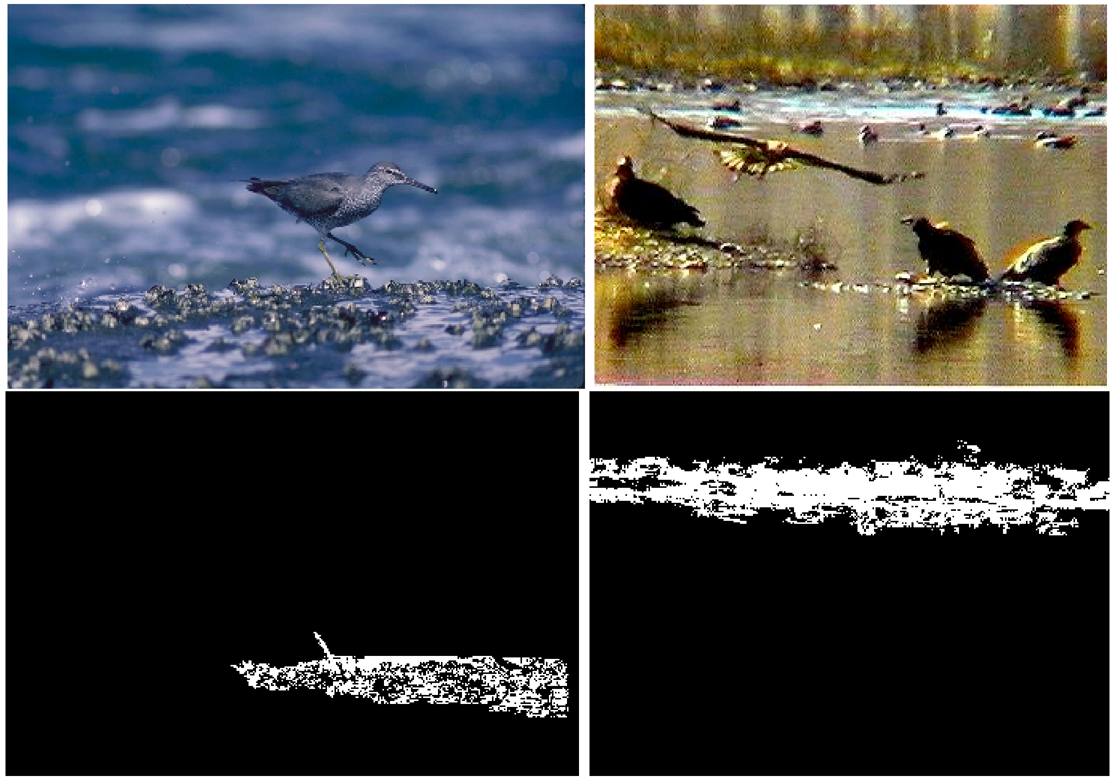


Figure 5.14. Examples showing failure cases where the bird segment was deleted : (top) original images (bottom) final segment obtained

2. The second class of images (insignificant background remaining) is indistinguishable from the first class in terms of image retrieval, since the colors indexed are predominantly from the bird regions. The small amount of pixels contributed by the background does not play a significant part in the retrieval.

3. When significant background remains (as in class 3), the images are indexed by colors from some background elements in addition to the bird regions, and this results in degraded retrieval performance (though still an improvement on

No background remaining	57%
Insignificant background remaining	13%
Image unchanged	17%
Significant background remaining	11%
Incorrect segmentation	2%

Table 5.1. Results of automatic segmentation on bird images

using the whole image. For example, in the top image of Figure 5.11, with the elimination of the blue sky which dominates the image, the emphasis placed on the colors from the bird is increased (since they now occupy a larger portion of the image).

4. Though it seems obvious that the cases where no segmentation could be achieved should result in poor performance (at par with using whole image indexing), just the opposite is true. The retrieval performance in this case is very close to the ideal case where no background is present. This is due to the fact that in these images (Figure 5.13) the background colors match the bird very closely, and including these colors does not make a significant difference to the indexing process, though they may change the proportion of colors to some degree,

5. The last class (incorrect segmentation) represents the true failures of our approach. The user would have better results using whole image indexing in these cases. In fact, since the bird region is excluded from the final segment, it is ensured that the user will not find any birds with matching colors in the retrieval results. and the output consists of parts of the background. However, this problem was encountered in a very small proportion of the images. In most cases where the bird was indistinguishable from the background, the segmentation algorithm was able to detect this situation, and output the image without segmenting it.

The overall results suggests that indexing based on the color of the bird is achieved in 87% of the images. In 11% of the images, the indexing includes some background colors, and in 2% of the images the colors of the bird were absent in the index.

Since the proposed foreground segment detection method does not use information specific to birds, it can be used without alteration on other images with single subjects with good results. Figure 5.15 show an example of other subjects (snake and butterfly)

Figure 5.15. Examples showing correct detection of subject in other domains (top) original images (bottom) final segment obtained

extracted correctly (in the case of the snake the segment extracted is sufficient to determine its color).

5.3.2 Results of indexing and retrieval

The retrieval performance of this system is compared with color-based whole image indexing which is very popular and forms the baseline we are proposing to improve upon. The database of bird images is indexed using color histograms [73] generated from the region of interest determined by the color and edge-based background elimination process described here, and using the whole image. Some examples of retrieval after using our region-of-interest pre-processing are shown in Figure 5.16. The retrieval results show that birds with colors similar to the query are retrieved in a variety of backgrounds. In some cases, the top-ranked images contain the same species of bird as the query, for example, in the first row of retrieved images in Figure

5.16, the top four images contain the same bird (cormorant). In the second example, all the retrieved birds are predominantly brown, with black and white specks. Except the second bird, all the other birds retrieved in this case are birds of prey, which are relevant to the query image of a kite. The third example has an owl as a query, but none of the retrieved images feature an owl, even though the brown and white coloration of the retrieved birds match the colors of the queried bird. As noted in the introduction to this chapter, color alone is insufficient to guarantee that the same species of birds are retrieved. Species with unusual and distinctive colorations are more likely to produce retrieval results where the species of the bird matches that of the query. In other cases where the bird has no distinctive colors, a color-based system can be expected to find other birds of similar color, at best. However, even in this case, it is likely that other birds of the same species would be ranked high in the retrieved list because of the similarity in the colors present and their relative proportions.

For comparison, Figure 5.17 shows the retrieval obtained using the same queries but using the whole image for color-based indexing. The examples clearly demonstrate that the background elements dominate the retrieval in this case. The first query produces other images with water as the background where none of the retrieved birds match the query bird's colors. The query features a black cormorant, while the retrieved images show brown ducks. The second query produces other birds against a blue sky where the third and fifth images show black and white birds, unlike the query which is brown. The third query generates birds against green backgrounds, with the color of the bird playing a secondary role in the retrieval. The same queries when posed on the database after background elimination retrieve images of other birds of similar color which are relevant to the query, without being affected by the type of background they are viewed against.

Figure 5.16. Region-of-interest-based retrieval : retrieved images in response to the query (the first retrieved image) when the database images were indexed by color from the automatically detected segment of interest only

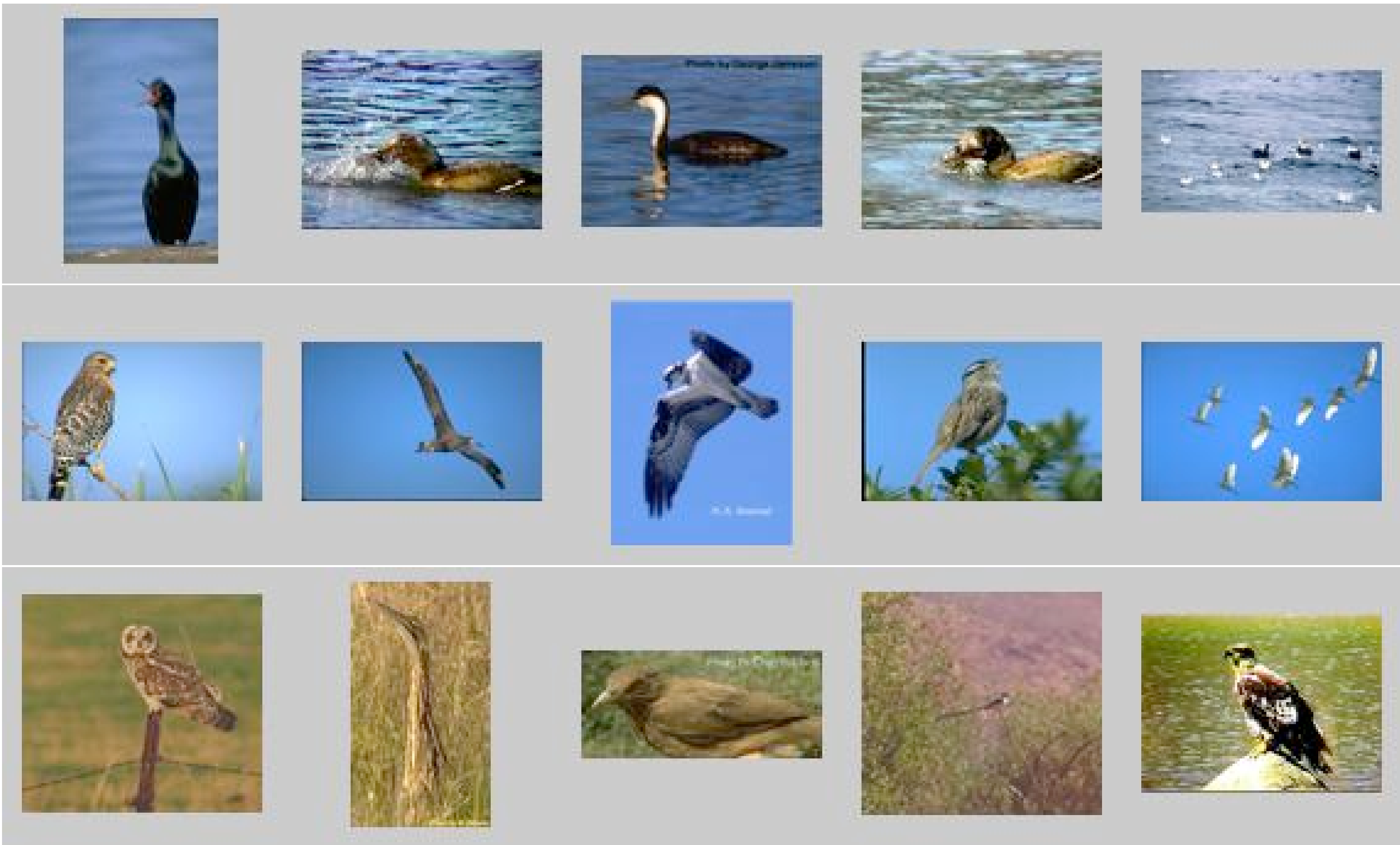


Figure 5.17. Whole image-based retrieval : images retrieved in response to the same queries as the previous figure, but when the database was indexed by color from whole images, with no segmentation

Unlike some other databases (advertisement images, for example), it is very difficult to judge the retrieval results in the bird database without having the knowledge-base of a birdwatcher. The difficulty lies in determining which birds in the database can be considered "similar" to the query. In the flower image database, the color of the flower was distinctive enough to make this determination. In the case of birds, there are hundreds of images in the database in which the bird can be categorized by a layperson to be "brown" or "black" (the most common colors encountered). Therefore, recall-precision scores could not be computed without a user study among a group of birdwatchers.

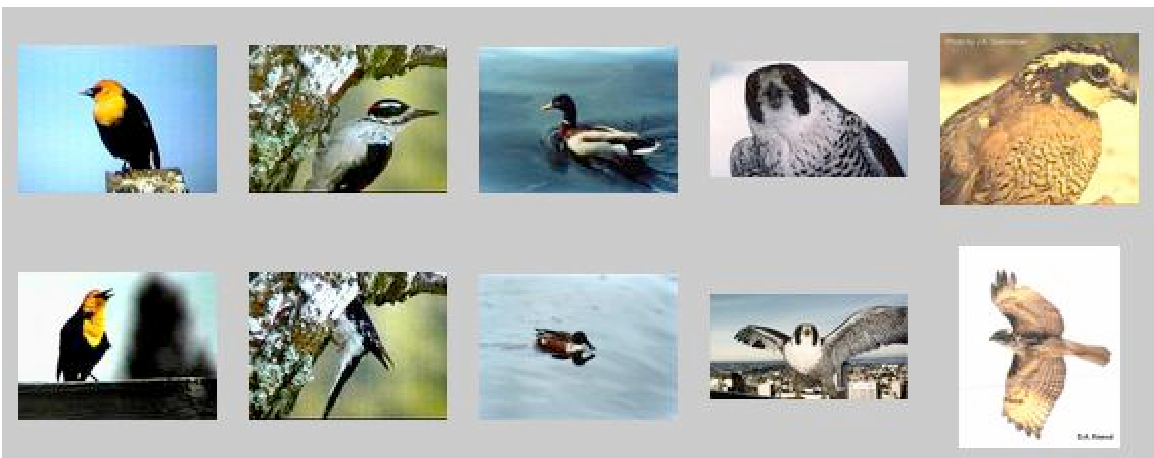


Figure 5.18. Examples of image pairs used to test retrieval results showing wide variations in size, pose and background

Retrieval method	Average rank of pair
Whole image indexing	6 (for a sub-set of 18) (≥40 for remaining 12)
Indexing using region of interest	3 (on the set of 30)

Table 5.2. Comparative retrieval results using whole image indexing and indexing in a region of interest

Instead, we have adopted an objective measurement criteria which effectively compares our system with whole image retrieval, without the need for judging each image in the database. A set of 30 pairs of birds were selected where each pair is known to be pictures of the same bird (either by obvious similarity or using cues from the image name given by the original photographer e.g. cormorant1 and cormorant2 obviously refer to two images of the same bird species). Some examples of image pairs used are shown in Figure 5.18. The pairs show wide variations in the appearance of the birds due to their non-rigid structure, in addition to differences in backgrounds. Using one image of the pair as a query, the rank of the other image of the pair was noted. It is expected that the corresponding image, being the same bird as the query, should appear near the top of the retrieved images. Table 5.2 summarizes the observed results from this test. It is to be noted that the rank of the corresponding pair in 12 out of the 30 image pairs was beyond 40 (effectively, the pair was not retrieved) using whole image indexing. The average rank of the rest of the pairs (18) was also far worse than that obtained by indexing after our region-of-interest segmentation. So we can conclude that using the computed region-of-interest for indexing significantly improves the effectiveness of retrieval.

5.4 Conclusion

We have proposed a solution to the problem of region of interest extraction while making very general assumptions about the images in the database, which are true of a broad class of images. Our approach to foreground segment detection is based on the elimination of background. This is accomplished by combining a color-based background detection step with refinement of the segmentation using edge information.

Color histograms from the automatically detected foreground segment are used to index a database of bird images. The retrieval results on this database show that the color of the bird is used for retrieval, without being affected by the colors present in the background. This is a very important improvement in a database of images with single subjects where the query is usually on the subject, and the background is incidental.

A possible extension of this work involves the incorporation of a region of interest selector in the user interface (as described in the chapter on retrieval from an advertisement image database). This would ensure that the color of the bird in the query image would be correctly assessed. On its own, this would be of limited use, since the database images would still have to be indexed based on the whole image. However, when combined with the region-of-interest pre-processor described here, it could be ensured that a failure to segment the query image correctly (which would lead to very poor results) is avoided. There is some robustness in the database images to erroneous segmentation, since at worst, it would result in the non-retrieval or false retrieval of a few images. In the case of the query, which is a single image, incorrect segmentation would guarantee poor results.

In a domain such as birds, the success of retrieval based on color alone is limited, since color cannot be used to distinguish between birds of different species. Other information such as shape, texture and rules formulated by expert birdwatchers need to be incorporated to ensure better discrimination between different types of birds. However, our method still provides the starting point for computing additional information by segmenting the region of interest from which such information should be gathered.

CHAPTER 6

SUMMARY AND FUTURE WORK

The overall goal of the research described in this dissertation was to develop content-based retrieval strategies for specialized image domains where the performance of general-purpose image retrieval techniques were poor and could be improved by taking the special characteristics of the domain into account. Three test domains which are representatives of a broader class of domains, were selected for this work. Effective color-based retrieval strategies were proposed and tested in each of the three domains.

Where the database images depict objects, these objects are the primary content of the images. For successful content-based retrieval, the object of interest needs to be described accurately by the features used during indexing, and irrelevant background needs to be ignored. The underlying aim of all the retrieval strategies developed in this research is to isolate the object of interest from the background; using explicit pre-segmentation where possible, and using features which are robust in the presence of background where the object of interest cannot be pre-segmented.

6.1 Contributions

The first test domain (advertisement images) contained the query object embedded in a lot of background and at a wide variety of sizes. Both the presence of background and scale variation pose major problems for existing retrieval systems. We propose a new two-phase, color-based image retrieval system [13] which is capable of identifying multi-colored query objects under such adverse conditions. The retrieval system is based on two new, scale-invariant color features which can be computed reasonably accurately even when there is interfering background present. An efficient matching phase is designed so that the proposed system is also very fast, enabling its use with online user interfaces. This retrieval engine is appropriate for any other multi-colored object database where wide variations in object size and background is expected.

When the domain characteristics are such that there is a prominent object of interest in the presence of simple backgrounds, we propose methods for the automatic extraction of the object of interest. Features computed from the object of interest only are then used to index the database images, making the retrieval independent of the image background. The two domains with these characteristics that we examine in this thesis differ in the amount of usable domain knowledge available which can be used to automate the segmentation process. In the domain of flower images, there is specific color-based domain knowledge which can be used directly to eliminate a lot of the naturally occuring backgrounds like leaves, soil, shadows etc. making the segmentation task simpler. As part of the solution for image retrieval in this domain, we develop an iterative algorithm for object of interest segmentation using domain knowledge to provide feedback about the correctness of the extracted region [10, 11]. This framework can be used in other domains where usable domain knowledge is available about the subject or the background in the image.

The framework developed for the flower domain is extended to the domain of bird images where domain knowledge is limited to general observations about photographs. In this case, we combine spatial and edge-based information with color to provide reasonable segmentation of the object of interest [12]. Since no information specific to the particular subject (birds) is used, the proposed segmentation methodology can be used in any domain where the object of interest is prominent in the image and is the focus of the image.

6.2 Future Work

The performance of the retrieval strategies described in this work are conditional on whether the target database meets the characteristics of the domain for which the strategy was designed. Even database characteristics which are more restrictive than the assumptions made about the domain during the design of the retrieval strategy, can make the strategy inappropriate. For example, if the FOCUS system developed for domains with large scale and background variations is used on a database where objects are always prominent and there is no significant background, the performance of the system would compare unfavorably with general purpose retrieval systems. This is because FOCUS does not use some object characteristics like the areas occupied by each color. This is necessary for scale and background invariant retrieval since the area occupied by the colors is highly variable when the size of the object varies or when there are interfering colors from the background. However, once these constraints are absent, it can be an important color feature which would provide more discrimination between different objects. So the description of database characteristics plays an important part in the selection of an appropriate retrieval strategy. In this thesis, this characterization is done manually; which is appropriate when a retrieval engine is being selected for a specific application. An open area of research is the automatic determination of object categories in a database, so that appropriate features and retrieval strategies could be selected. There has been some preliminary work recently in this area [80], where objects are represented as a probabilistic group of features, and common groups are selected by maximization of expectation.

Some specific future work directly relevant to the algorithms developed in this thesis include

- The second phase of matching in FOCUS which uses sub-graph isomorphism is currently a binary decision i.e. the graphs either match or they do not. However, there are cases where there are errors in the graph due to errors in peak detection (missed peaks, in particular) since peaks form the nodes in the graph. Recent work in error tolerant sub-graph isomorphism [44] could produce improved recall in the system. Also, a recently proposed method for detecting subgraph isomorphism using a decision tree, could produce faster results [45].

- Bayesian approaches [64] could be relevant to the flower and bird databases where there is a clear object of interest whose characteristics could be learned. The main drawback is the need for a large number of labeled images for training such systems.

- Experiments on other databases similar to the domains used in this thesis are necessary for testing the versatility of the proposed algorithms.

BIBLIOGRAPHY

[1] Ashley, J, Barber, R, Flickner, M, Hafner, J, Lee, D, Niblack, W, and Petkovic, D. Automatic and semi-automatic methods for image annotation and retrieval in QBIC. *SPIE Conference on Storage and Retrieval for Image and Video Databases* (1995), 24–35.

[2] Babu, G P, Mehtre, B M, and Kankanhalli, M S. Color indexing for efficient image retrieval. *Multimedia Tools and Applications 1*, 4 (Nov 1995), 327–348.

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

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

[5] Beveridge, J R, Griffith, J, Kohler, R R, Hanson, A R, and Riseman, E M. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision 2* (1989), 311–347.

[6] Buluswar, S.D., and Draper, B.A. Color recognition in outdoor images. In *ICCV98* (1998), pp. 171–177.

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

[8] Chen, Q, Wu, H, and Yachida, M. Face detection by fuzzy pattern matching. *Fifth International Conference on Computer Vision* (Jun 1995), 591–596.

[9] Cox, I.J., Miller, M.L., Minka, T.P., and Papathomas, T.V. The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing 9*, 3 (March 2000), 524.

[10] Das, M, , Manmatha, R, and Riseman, E M. Indexing flowers by color names using domain knowledge-driven segmentation. *IEEE Workshop on Applications of Computer Vision* (Oct 1998), 94–99.

[11] Das, M, , Manmatha, R, and Riseman, E M. Indexing flower patent images using domain knowledge. *IEEE Intelligent Systems 14*, 5 (Oct 1999), 24–33.

[12] Das, M, and Manmatha, R. Automatic segmentation and indexing in a database of bird images. *Ninth International Conference on Computer Vision* (Jul 2001), 351–358.

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

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

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

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

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

[18] Fuh, C.S., Cho, S.W., and Essig, K. Hierarchical color image region segmentation for content-based image retrieval system. *IEEE Transactions on Image Processing 9*, 1 (January 2000), 156.

[19] Funt, B.V., and Finlayson, G.D. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 5 (May 1995), 522–529.

[20] Gevers, T., and Smeulders, A.W.M. Content-based image retrieval by viewpoint-invariant color indexing. *Image and Vision Computing 17*, 7 (May 1999), 475–488.

[21] Gevers, T., and Smeulders, A.W.M. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing 9*, 1 (January 2000), 102.

[22] Gong, Y, Chuan, C H, and Xiaoyi, G. Image indexing and retrieval based on color histograms. *Multimedia Tools and Applications 2*, 2 (Mar 1996), 133–156.

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

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

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

[26] Horn, B K P. *Robot Vision.* The MIT Press, 1993.

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

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

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

[30] Kankanhalli, M S, Mehtre, B M, and Wu, J K. Cluster-based color matching for image retrieval. *Pattern Recognition 29*, 4 (Apr 1996), 701–708.

[31] Kankanhalli, M.S., Mehtre, B.M., and Huang, H.Y. Color and spatial feature for content-based image retrieval. *Pattern Recognition Letters 20*, 1 (January 1999), 109–118.

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

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

[34] Korth, H F, and Silberschatz, A. *Database System Concepts*. McGraw-Hill, 1991.

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

[36] Liu, S, and Yang, R. Multiresolution color image segmentation. *IEEE transactions on Pattern Analysis and Machine Intelligence 16*, 7 (Jul 1994), 689–700.

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

[38] Mahamud, S, Thornber, K K, and Williams, L R. Segmentation of salient closed contours from real images. *Seventh International Conference on Computer Vision* (Jun 1999).

[39] Malik, J, Belongie, S, Shi, J, and Leung, T. Textons, contours and regions : Cue integration in image segmentation. *Seventh International Conference on Computer Vision* (Sept 1999).

[40] Manduchi, R. Bayesian fusion of color and texture segmentations. *Seventh International Conference on Computer Vision* (Sept 1999).

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

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

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

[44] Messmer, B.T., and Bunke, H. A new algorithm for error tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 5 (May 1998), 493–504.

[45] Messmer, B.T., and Bunke, H. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition 32*, 12 (December 1999), 1979–1998.

[46] Milanese, R., and Cherbuliez, M. A rotation, translation, and scale-invariant approach to content-based image retrieval. *Journal of Vision Computing and Image Representation 10*, 2 (June 1999), 186–196.

[47] Nagin, P A, Hanson, A R, and Riseman, E M. Studies in global and local histogram-guided relaxation algorithms. *IEEE transactions on Pattern Analysis and Machine Intelligence 4*, 3 (May 1982), 515–522.

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

[49] Niblack, W, Barber, R, Equitz, W, Flickner, M, Glasman, E, Petkovic, D, Yanker, P, Faloutsos, C, and Taubin, G. The QBIC project: Querying images by content using color, texture and shape. *SPIE Proc. on Storage and Retrieval for Image and Video Databases 1908* (Feb 1993), 173–187.

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

[51] Ohlander, R B, Price, K E, and Reddy, D R. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing 3*, 8 (May 1979), 515–522.

[52] Pal, N.R., and Pal, S.K. A review on image segmentation techniques. *PR 26*, 9 (September 1993), 1277–1294.

[53] Park, I.K., Yun, I.D., and Lee, S.U. Color image retrieval using hybrid graph representation. *Image and Vision Computing 17*, 7 (May 1999), 465–474.

[54] Pauwels, E.J., and Frederix, G. Nonparametric clustering for image segmentation and grouping. *Image Understanding 75*, 1 (2000), 73–85.

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

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

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

[58] Ravela, S, and Manmatha, R. Image retrieval by appearance. *SIGIR* (Jul 1997), 278–285.

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

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

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

[62] Rui, Y., Huang, T.S., and Chang, S.F. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation 10*, 1 (March 1999), 39–62.

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

[64] Schneiderman, H., and Kanade, T. A statistical method for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition* (2000), pp. I:746–751.

[65] Sclaroff, S, Taycher, L, and La Cascia, M. 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).

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

[67] Shi, J, and Malik, J. Self inducing relational distance and its application to image segmentation. *Fifth European Conference on Computer Vision* (Jun 1998).

[68] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE transactions on Pattern Analysis and Machine Intelligence 22*, 8 (August 2000), 888–905.

[69] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., and Jain, R.C. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 12 (December 2000), 1349–1380.

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

[71] Stricker, M A. Bounds for the discrimination power of color indexing techniques. *SPIE Conference on Storage and Retrieval for Image and Video Databases 2185*, II (Feb 1994), 15–24.

[72] Swain, M J, and Ballard, D H. Indexing via color histograms. *Third International Conference on Computer Vision* (1990), 390–393.

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

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

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

[76] Tremeau, A., and Colantoni, P. Regions adjacency graph applied to color image segmentation. *IP 9*, 4 (April 2000), 735.

[77] Ullman, S. *High level vision : Object recognition and visual cognition*. MIT Press, 1996.

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

[79] Wang, J. *Integrated Region-based Image Retrieval*. Kluwer, 2001.

[80] Weber, M., Welling, M., and Perona, P. Towards automatic discovery of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition* (2000), pp. 101–108.

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

[82] Williams, L R, and Thornber, K K. A comparison of measures for detecting natural shapes in cluttered backgrounds. *International Journal of Computer Vision*, 34 (1999), 81–96.

[83] Woelker, W. Image segmentation based on an adaptive 3d-analysis of the ciel*a*b* color space. *Proceedings of SPIE 2727*, 3 (1996), 1197–1203.

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

[85] Yee, C Y, Tan, K L, Chua, T S, and Ooi, B C. An empirical study of color-spatial retrieval techniques for large image databases. *IEEE Conference on Multimedia Computing and Systems* (June 1998), 218–221.

[86] Zhang, Y.J. A survey on evaluation methods for image segmentation. *Pattern Recognition 29*, 8 (August 1996), 1335–1346.

[87] Zhang, Y.J. Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters 18*, 10 (October 1997), 963–974.