ABSTRACT

Title of thesis: Presenting Visual Information to the User : Combining Computer Vision and Interface Design

Gaurav Agarwal, Master of Science, 2005

Thesis directed by: Professor David Jacobs

In this work, we suggest better ways to present visual information (image databases) for browsing and retrieval. Thumbnails obtained from an image set give a good overview of its contents. Instead of simply downsampling images to obtain thumbnails, we first find salient regions (saliency map) using local statistical features of the image. We crop and downsample the images based on these saliency maps, and obtain better thumbnails. The suggested methods of finding salient regions are faster than existing methods while giving comparable results.

Secondly, we have developed a Content Based Image Retrieval (CBIR) system to provide empirical evidence (by user study) that similarity based grouped and hierarchical placement of images is better than random placement. Using an effective shape based similarity measure we conclude that visual search is very useful in image retrieval systems. We conducted a field test to check the robustness of the system in varying photography conditions.

Presenting Visual Information to the User : Combining Computer Vision and Interface Design

by

Gaurav Agarwal

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2005

Advisory Committee:

Professor David Jacobs, Advisor
Professor Rama Chellappa, Chair
Professor Min Wu

ACKNOWLEDGMENTS

I owe my gratitude to all the people who have made this thesis possible.

First and foremost, I'd like to thank my advisor Prof. David Jacobs for making himself available whenever I asked for any help and advice. Regular discussions with him have molded my approach to problem solving. He showed me the importance of intuitive, clear understanding and that the quality of work is what matters in the long run. His enthusiasm and dedication for any work has always been a source of inspiration for me. It has been a pleasure to work with and learn from such an extraordinary individual.

I am grateful to the Chair of the committee and my academic advisor, Prof. Rama Chellappa. He has been very supportive from the time I started my graduate studies.

I would like to thank my colleagues, Haibin Ling, Aravind Sundaresan and Sameer Shirdhonkar for the fruitful discussions we had and for their help.

Prof. Ben Bederson and his students Bongwon Suh and Hilary Hutchinson have provided help in the field of Human Computer Interaction. I am grateful to Prof. Bederson for permitting us to use their technology and for other suggestions. Bongwon provided support for the software and Hilary guided me in the design and analysis of the user study.

I would like to acknowledge Dr. John Kress and Rusty Russell of Smithsonian

Institute, who provided us with the data, in the way we wanted. They have also been instrumental in organizing the user study.

I would like to take this opportunity to thank all the participants of the user study who provided invaluable feedback.

I would also like to acknowledge financial support from the National Science Foundation (NSF Grant Number: ITR-03258670325867), for the project discussed herein.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Content-based image retrieval (CBIR), also known as Query by Image Content (QBIC) and Content-based Visual Information Retrieval (CBVIR), has gained importance in recent years [2, 3]. The basic purpose of these systems is to provide a user, visibility to the contents of large image databases. In general, Computer Vision based techniques are used to extract features of database images that are matched with user input in the form of an image, sketch or low-level features like color and texture. Browsing and navigation has never been considered an important part of these systems and only recently have these techniques gained importance [4, 5, 6].

We have a leaf database from Smithsonian Institute which has more than 1500 images from 130 different species. Proper browsing, navigation and search tools are needed to quickly find a particular leaf. We have developed a CBIR system that combines computer vision based shape matching algorithms with interface design techniques. This system uses clustered and hierarchical organization of images, combines browsing and search and uses an animated user interface to give a better browsing experience.

Usually the image databases are very large in size (more than 10000 images). It seems implausible to present the entire data set to the user. Thus, to build a

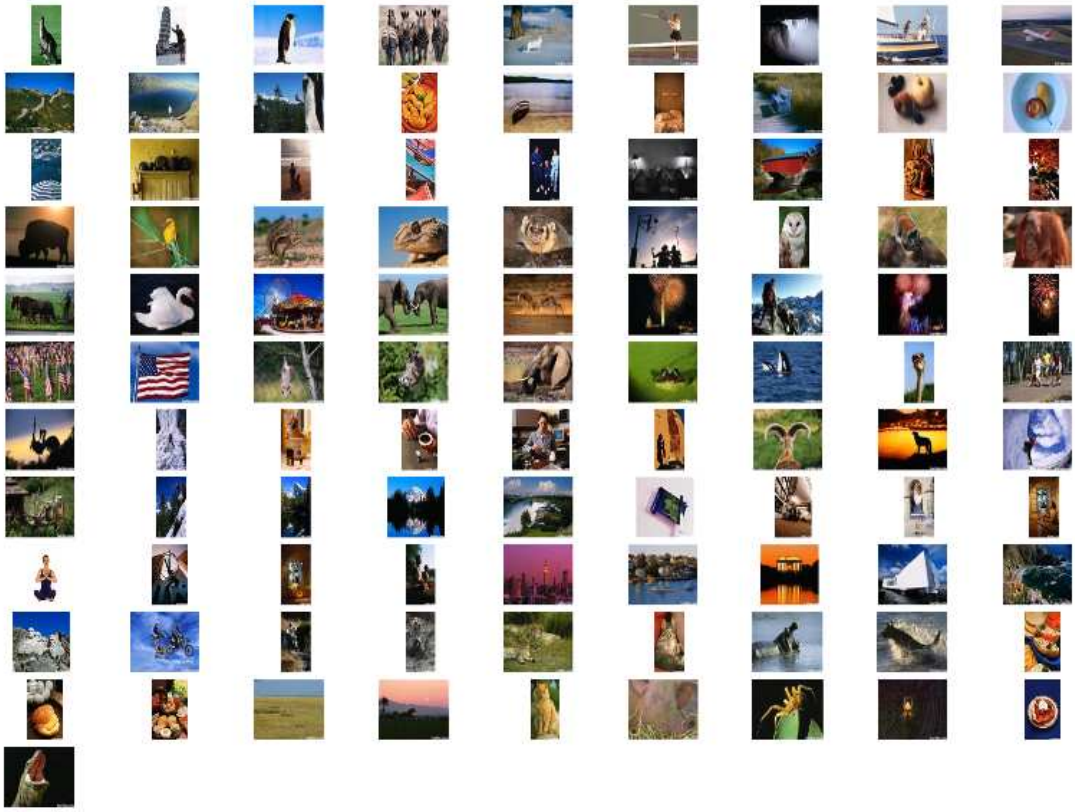


Figure 1.1: 100 images are arranged in a rectangular grid in a random order. User can take a glance and get the idea of the contents of this 100 images database

system having effective image browsing, the issues are to find ways to present the visual information to the user and to provide effective mechanisms to be able to navigate through the large databases.

Lets start by discussing the methods to present the visual information to the user. Given N (> 100) images (size $> 200 \times 200$), how can information about the images be made available to users in a meaningful way? One way is to arrange images in a 2 dimensional or 3 dimensional space. Combs et al [7] have shown that a 2D arrangement of images is better than 3D. Figure 1.1 shows 100 images arranged in a 2D rectangular grid in a random fashion. The user can take a glance

and get the idea of the contents of this 100 images database. Most recent image retrieval systems show thumbnails (downsampled versions of the images) of images arranged in a 2D array fashion on the screen. Thumbnails are used as they give a good idea of the image content and because of their small size, a large number of these can be displayed on the screen. The thumbnails are generated in general by pure downsampling of the image. In images, usually there is a foreground and a background. Most of the semantic information is contained in the foreground. If we crop the background and downsample only the foreground, this would give us much better thumbnails. In this work, we have proposed computationally efficient methods to automatically indicate regions of importance (foreground) in an image.

Only a limited number of thumbnails can be shown on the screen. The rest of the data has to be made available by browsing and navigation. To deal with the problem of organizing a large amount of visual information for effective navigation, we use clustering and hierarchical placement of data. These schemes have been suggested earlier and we provide empirical evidence to prove the usefulness of this approach. We conducted a user study to show that the clustered and hierarchical placement of data is better than random.

The CBIR system that we have developed is meant to be used in the field with varying photography conditions. So we also conducted a field test with the system to check its robustness.

1.2 Previous Work

The input given to the CBIR system by the user has been classified as query by example (QbE), query by painting (QbP), query by color (QbC) and query by text (QbT) [4]. The first generation of CBIR systems focused on using these query types to find matches with minimal feedback from the user [8, 9]. Due to the complexity of the images, this approach has been found insufficient to express the visual information the user is looking for (this problem has also been called the semantic gap [10]). This leads to unsatisfactory results. This semantic gap increases when the database is of similar objects like a collection of faces and leaves. The second generation of CBIR systems tried to bridge this semantic gap by having successive feedback from the user, thus refining the query input [10]. Based on this feedback, the system updates the present understanding of the query and modifies the successive query results. In this approach, with multiple feedback, the system might add undesired features thus degenerating the results.

In the above mentioned "system suggested" searches, users follow paths given by the system. Every time the system responds to a query, the user gets some local view of a part of the database. This restricts the amount of information a user can have. Rubner et al [11] added a new paradigm to image retrieval by suggesting a global placement (using Multi Dimensional Scaling [12]) of images on a screen based on image similarity. Chen et al [5] and Pecenovic et al [4] extended this idea by integrating browsing and searching, highlighting the importance of browsing. They suggested that the user should have access to the global view of the database and

the system should "guide" the user in the combined browsing and search through the database.

To provide the global view, it is essential to show a large number of images on the screen. As discussed earlier, thumbnails are a good way to display images on the screen. Usually thumbnails are downsampled version of the images. Suh et al [13] improved the way these thumbnails are generated and showed that presenting cropped and down-sampled thumbnails (which capture the saliency of an image) is better than displaying normal down-sampled thumbnails. This cropping essentially means that more thumbnails can be incorporated in the same screen space thus making more of the database visible on the screen. They used the computationally expensive Itti's algorithm to find the regions of importance. Using this algorithm for applications on battery powered handheld devices like cell phones and PDAs which have limited processing power is not feasible.

To deal with the problem of organizing a large amount of visual information for effective navigation, Laaksonen et al [14] and Chen et al [5] suggested self-organizing maps and pyramids respectively, to provide a hierarchal structure to large image databases. The top most level was shown initially to the user (by means of thumbnails), to help choose an image. The chosen image represents a group having similar images based on some similarity criteria. These group are not displayed initially on the screen. By selecting that image, the user is going to navigate through that group. This can go on depending on the number of levels. Using this pyramid structure, a user can recursively navigate through the whole database. Content-based Image Retrieval and Consultation User System (CIRCUS)

of Pecenovic et al [4] followed similar lines and displayed images on a non-uniform 2-D grid. They used progressive multi-resolution coding of images to display the required level of images. The user can zoom in to see more images of that cluster and can initiate search at any level.

These approaches highlighted the importance of browsing and navigation, though no controlled user studies were conducted that confirmed the improvement (over earlier approaches) empirically. Rodden et al [15] conducted a user study that concluded that labelled captions on groups assist browsing as compared to groups without captions or randomly displayed images. However, they also found that a random placement of images is better than grouping by image similarity because in a random placement images "pop-up" while in a similarity based placement they dissolve into their surroundings. This is, in general, counter intuitive and these results might be because of improper grouping of images, which would hurt more. On the other hand, Liu et al [6] found empirical evidence that visual similarity based grouping assists browsing. They compared clustered sets of images with random placement and found browsing a clustered set is faster and more accurate. Their study was targeted more towards arranging images (from a web based image search).

Improving the ways the visual information is presented to the user has been gaining interest in the research community. Recently Su [16] used texture based features to make the background less distracting, thus increasing the overall perceptual quality of the image. Rother et al [17] used saliency based technique to merge various images into a single image. This gives an effective visual summary of a collection of images.

1.3 Organization

The thesis is organized as follows: Section 2 presents the new methods and previous work to find the saliency map. The CBIR system (Electronic Field Guide) that we have developed is discussed in Section 3. Section 4 and Section 5 talks about layout schemes. Section 6 presents the results of the user study and Section 7 describes the field test done for the system. Section 8 concludes the thesis.

Chapter 2

Saliency

A thumbnail of a digital image can be defined as a down-sampled version of the original image. The basic advantage of using a thumbnail is less use of resources (storage media, screen space, decoder bandwidth) for processing, while they give a good overview of the full resolution image. Because of this, they are used extensively in the digital world. Most image processing software use thumbnails in some way or other. Many webpages show thumbnail versions of the image and the user can click on the image to see more details. Even the Windows operating system folders has an option to show thumbnails. As mentioned earlier, recent CBIR systems that emphasize combined browsing and search also use thumbnails on the initial display screen.

A thumbnail should provide a preview of all the semantic information that the full resolution image has. The level of details might be reduced. To save more resources, more downsampling is preferred, to a level such that any significant detail present in the original image should not be lost or become unclear. There is no good answer to the amount an image should be downsampled.

A large variety of images can be categorized as "object-oriented" or having a foreground (object) and a background. Foreground is usually an object of interest (like humans, animals, man made object etc) while background is the environment

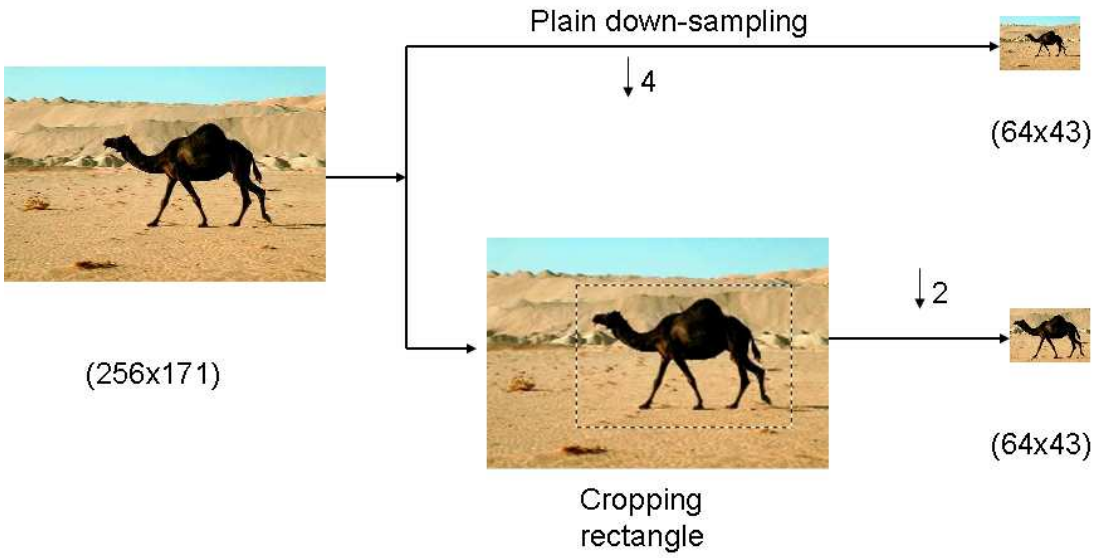


Figure 2.1: Cropping and downsampling is better than pure downsampling to generate thumbnails

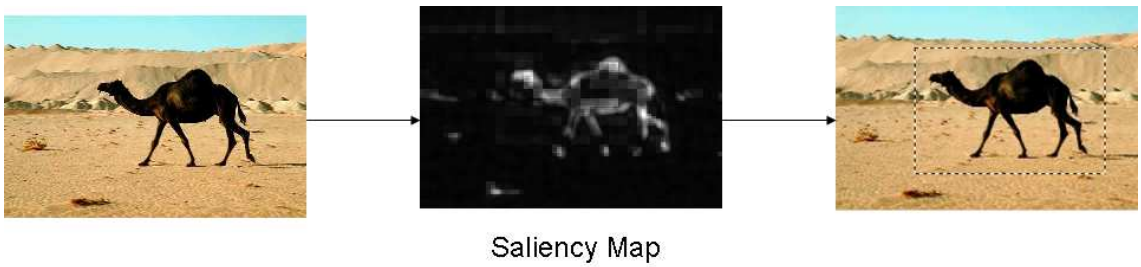


Figure 2.2: Saliency Map and the selection of the rectangle with optimum saliency keeping the size of the rectangle minimum

containing that object. An example of an exception to this is an image of natural scenery. If the image has a distinct foreground and background, usually foreground is of interest and most of the meaningful information that the image conveys is contained in the foreground. If this foreground is cropped and downsampled, this would be a better thumbnail as compared to thumbnail generated by same amount of pure downsampling of the image. Figure 2.1 shows the comparison of the thumbnail generated by pure downsampling and by cropping and downsampling. Clearly, the thumbnail generated by cropping and downsampling is better. Suh et al [13] used Itti's [1] algorithm to automatically identify the foreground or the salient regions in an image. They showed that careful cropping and downsampling is a better technique to generate thumbnails. They conducted user studies involving recognition and searching tasks to conclude this. Chen et al [18] also used a similar approach.

The saliency map generated using Itti's algorithm is used by Suh et al to decide the area to crop. The saliency map indicates the importance of a pixel as compared to other pixels in the image. They find the smallest rectangle that includes the optimum saliency for cropping. Figure 2.2 shows an example of the saliency map and the chosen rectangle.

Itti's algorithm is based on a computational model for visual attention [19] which is described in the next section. Itti's method is effective but computationally expensive. In this section, we suggest faster methods to find the salient regions or the saliency map for an image.

2.1 Motivation

The primate's visual system efficiently processes the enormous amount of information it receives. In recent years computational models based on the structure and experiments on the visual system have been suggested [20] which attempt to explain the effective real time processing of information by the visual system. While looking at an image, initially (early vision) the attention of the visual system moves across different locations trying to find the regions of interest. The maximal rate of the movement of fovea (called "saccades") is around 5 locations per sec (the location of attention and the location of fovea has mostly been found the same). The location where the attention will move next depends on two general classes of selection mechanism. First, bottom-up [20] selection that involves fast and stimulus driven mechanisms. The stimulus, which guides the selection mechanism has been found to be based on the properties of some parts of the visual input. On the other hand, top-down selection, is a slower goal-directed mechanism where the observer's intentions and expectations direct the path of the attention.

After each move, the attention stays at the new point for some time. This is called fixation. Then the attention moves to a new location guided by the combination of bottom-up selection and top-down selection mechanisms. The bottom-up selection mechanism has been found to be controlled by the statistical properties of the visual input [21]. The low-level features of image like color, contrast and orientation in a center-surround fashion has been found effective in modelling the bottom-up selection mechanism [22].

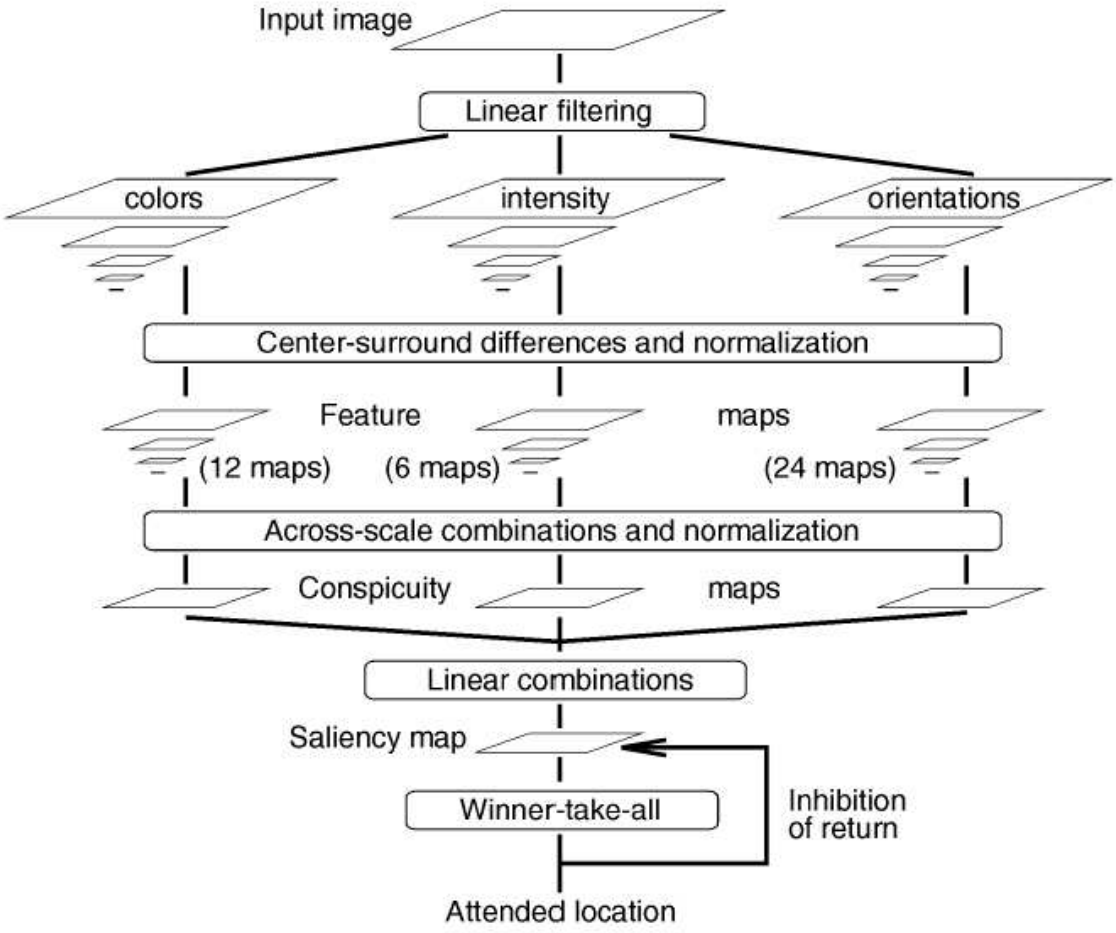


Figure 2.3: Itti's algorithm to generate various feature maps. Taken from [1]

Itti et al [1] presented a computational model (using the low-level features of the image) for the bottom-up selection mechanism. Their results showed that their model is able to match the initial locations selected by the visual system. The main points of the algorithm are as follows (Figure 2.3):

- Color, intensity and orientation are used as input features.
- Center surround (Section 2.2.3) is implemented using a multiscale pyramid.
- The output maps are normalized and combined in a Winner Takes All (surround inhibit (Section 2.2.4)) strategy.

Recently Parkhurst et al [23] also validated the correlation between locations of eye fixation (under natural viewing conditions) and saliency (calculated using Itti's algorithm [1]) at that points. Parkhurst et al [24] confirmed that the location of fixations have higher order statistical properties (they used local contrast, local spatial correlation and spatial frequency content). Similar claims were made earlier by Reinagel et al [25], Krieger et al [26] and Mannan et al [27, 28].

2.2 Is this what we want?

All the methods discussed above intend to find a correlation between the statistical properties of an image and bottom-up selection mechanisms of active vision. There are some issues that need attention.

Our intention is to find objects or regions of importance (ROIs), that should give all the semantic information in the image. Suh et al [13] and Chen et al [18]

used Itti's model assuming that the bottom-up selection mechanism of the active mechanism selects the regions that are the foreground or the ROIs.

2.2.1 Do we need an algorithm as complex as Itti's ?

The image set used by Parkhurst et al [23] (which uses Itti's algorithm [1] to find the saliency map) were: Fractals, Natural Landscapes, Building and City Scenes and Home Interiors. These are general images and it's difficult to manually define what are the ROIs. On the other hand, Reinagel et al [25] and Mannan et al [27, 28] used images with humans, animals and distinct man-made objects apart from the general images. Reinagel et al and Mannan et al found correlation between local statistical properties (they used local contrast, edge density, local spatial correlation and spatial frequency content) and bottom-up selection mechanisms of the active vision. These local statistical properties of images have also been found effective for the database used by Parkhurst et al. Thus, its not clear if using the complex Itti's algorithm is best for object oriented images, or whether simpler image statistics are sufficient to find saliency.

2.2.2 Is specific orientation a must?

Neither of Reinagel et al and Mannan et al used orientation as a separate feature which was used by Itti et al. This might be because edge density and contrast inherently capture all the orientations. One of the possible reasons Parkhurst et al found orientation significant is that the 2 sets of images they used ("Building and

City Scenes" and "Home Interiors") have a high density of edge orientation features.

2.2.3 What is center surround?

Itti et al implement the center surround mechanism using multiscale techniques. The center corresponds to the value of the pixel at level n of the image pyramid and the surround to the corresponding pixel at level $n+\delta$ where $\delta \in \{3, 4\}$, level 0 being the finest resolution. The level $n + \delta$ is the low pass filtered version of n. Thus essentially what we are capturing is the "change" between these two scales which corresponds to high and medium frequency contents of the image, which are also captured by local contrast and edge density.

2.2.4 Do we need surround inhibit?

Itti's algorithm aims at finding a few salient peaks in the image. To do this, they use computationally expensive surround inhibit mechanism which enhances the regions which are significantly more important than their surroundings. We don't want peaks but approximate regions to find the ROIs. So we need not use surround inhibit.

2.3 Faster methods

From the above discussion it seems plausible to use local statistical features of the image (local contrast, edge density and high and medium frequency energy) to isolate the salient regions or the ROIs. Also, surround inhibit is not needed as

we are not interested in peaks but regions. We present some new methods that are simpler than Itti's approach.

2.3.1 Variance or Local Contrast Map

A good correlation has been found between the location chosen by the bottom up selection mechanism and the variance at that point [24, 27, 26]. In this section, we will describe a method to find the saliency map using variance as a feature.

Algorithm

The input image I is divided into 8×8 (N=64 pixels) non-overlapping blocks (Figure 2.7). For an image size of $M \times K$, suppose there are $m \times k$ ($m = M/8, k = K/8$) blocks. In this method, we calculate the saliency of the block rather than that of the pixel. For each block, the variance is calculated as follows:

Let $I_{i,j}(x, y)$ represent the $(x, y)^{th}, (1 \leq x \leq 8, 1 \leq y \leq 8)$ pixel in $(i, j)^{th}$ block. The variance of a bock is defined by:

$$V(i, j) = \frac{1}{N - 1} \left(\sum_{x=1}^{8} \sum_{y=1}^{8} \left(I_{i,j}(x, y) - \mu_{ij} \right)^2 \right) \tag{2.1}$$

where μ_{ij} is the mean for the i^{th} and j^{th} block defined by:

$$\mu_{ij} = \left(\frac{1}{N} \sum_{x=1}^{8} \sum_{y=1}^{8} I_{i,j}(x, y) \right) \tag{2.2}$$

Figure 2.5 (d) and 2.6 (d), shows the result. The results looks noisy though the region of interest is highlighted.

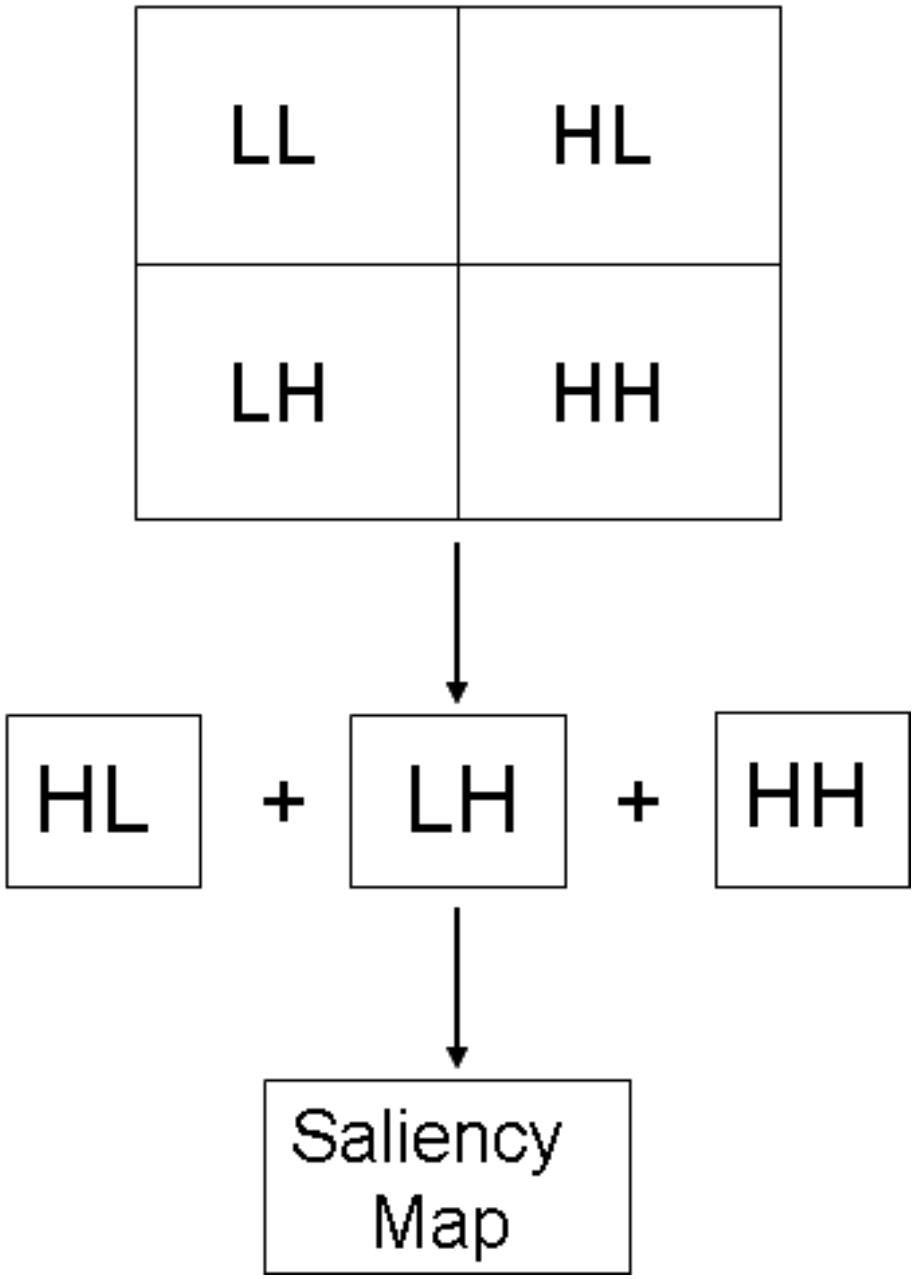


Figure 2.4: Generation of saliency map using wavelet. LH, HL and HH bands are combined to find the final map

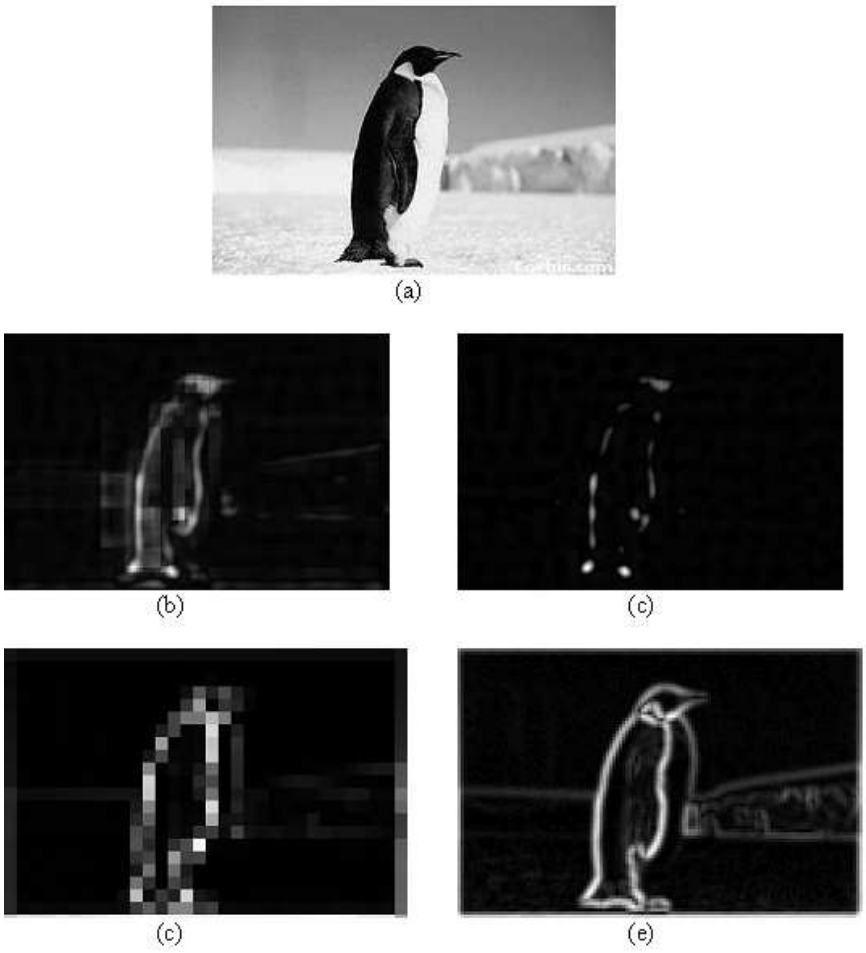


Figure 2.5: Saliency Maps using various methods (a) Original Image (256×170), (b) Using Itti's algorithm without surround Inhibit , (c) Using Itti's algorithm with surround inhibit, (d) Using Variance, (e) Using Wavelet. Color channel has not been used. More salient regions have higher gray scale value

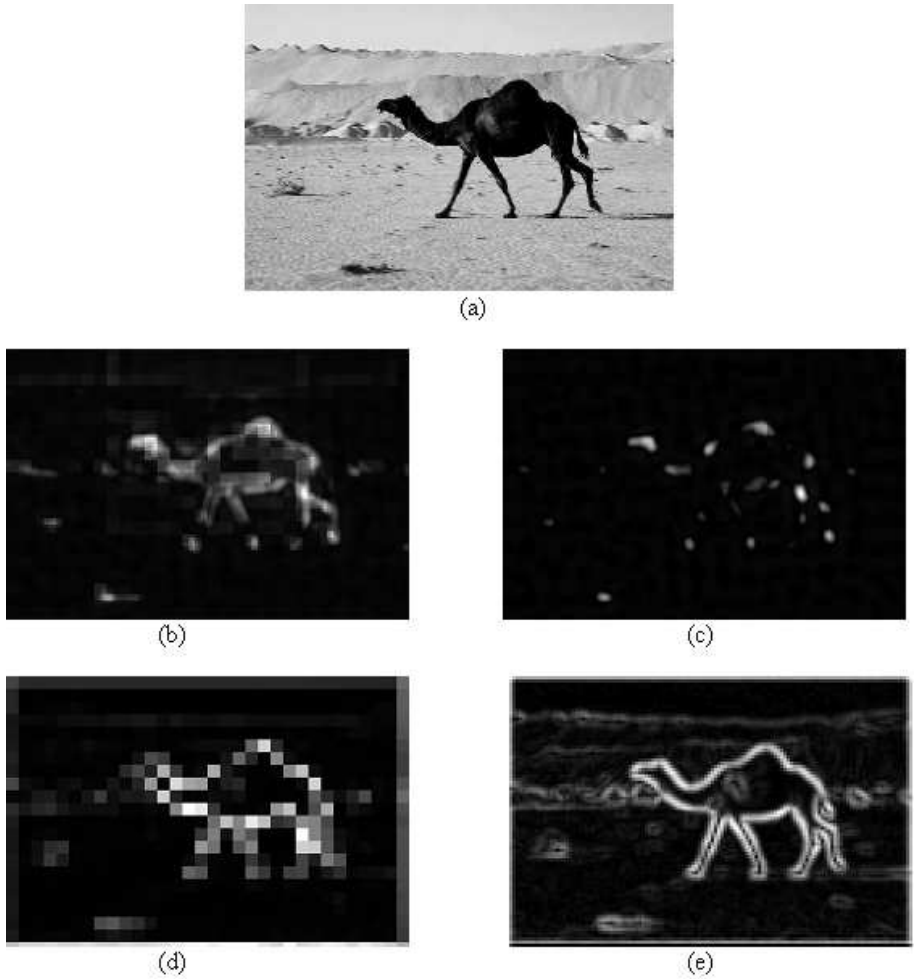


Figure 2.6: Saliency Maps using various methods (a) Original Image (640×427), (b) Using Itti's algorithm without surround Inhibit , (c) Using Itti's algorithm with surround inhibit, (d) Using Variance, (e) Using Wavelet. Color channel has not been used. More salient regions have higher gray scale value

2.3.2 Variance using the DCT coefficients

Agarwal et al [29] used spatial frequency content to generate saliency maps in the compressed domain. But they have not provided any empirical evidence to verify their claims. Mannan et al [27] and Kreiger et al [26] found no direct correlation between the spatial frequency content of the image and the locations chosen by a bottom-up selection mechanism. Parkhurst et al [24] found significant correlation only in "Fractals" and "Building and City Scenes" databases, which questions the consistency of spatial frequency content as a general feature to model the bottom up saliency mechanism.

In this subsection, we adapted the variance criteria explained in the last section, for the DCT domain. As mentioned in the Section 2.3.1, variance has been found a good measure for saliency [24, 27, 26]. We here show that the variance calculated in the pixel domain can fully be calculated using the DCT coefficients. The DCT domain is important as most compressed video (MPEG-x, H26x) and a popular compressed image (JPEG) domain uses DCT to achieve spatial compression. Using the method explained below, the salient regions could be obtained directly in the compressed domain, which has many practical applications, e.g., ROI based coding, video and image transcoding [30].

Algorithm

Like in Section 2.3.1, the image is divided into 8×8 (N=64 pixels) non-overlapping blocks (Figure 2.7). For an image size of $M \times K$, there are $m \times k$

$(m = M/8, k = K/8)$ blocks. For each block a 2 dimensional 8 point DCT is taken. Let's say $I^d = dct2(I)$. Let $I_{i,j}(x, y)$ represent the $(x, y)^{th}, (1 \leq x \leq 8, 1 \leq y \leq 8)$ pixel in $(i, j)^{th}$ block. block. On similar lines, let $I^d_{i,j}(x, y)$ represent the $(x, y)^{th}, (1 \leq x \leq 8, 1 \leq y \leq 8)$ dct coefficient in $(i, j)^{th}$ block. The dc coefficient of the block, $I^d_{i,j}(1, 1)$ is the scaled mean of the pixel domain block.

$$\mu_{ij} = \frac{1}{\sqrt{N}} I^d_{i,j}(1, 1) \tag{2.3}$$

Using Parseval's rule:

$$\sum_{x=1}^{8}\sum_{y=1}^{8}(I_{i,j}(x, y))^2 = \sum_{x=1}^{8}\sum_{y=1}^{8}(I^d_{i,j}(x, y))^2 \tag{2.4}$$

The variance is:

$$\begin{aligned}
V(i, j) &= \frac{1}{N-1}\left(\sum_{x=1}^{8}\sum_{y=1}^{8}\left(I_{i,j}(x, y) - \mu_{ij}\right)^2\right) \\
&= \frac{1}{N-1}\left(\sum_{x=1}^{8}\sum_{y=1}^{8} I^2_{i,j}(x, y) - 2\mu_{ij}\sum_{x=1}^{8}\sum_{y=1}^{8} I_{i,j}(x, y) + N\mu^2_{ij}\right) \\
&= \frac{1}{N-1}\left(\sum_{x=1}^{8}\sum_{y=1}^{8} I^2_{i,j}(x, y) - N\mu^2_{ij}\right) \\
&= \frac{1}{N-1}\left(\sum_{x=1}^{8}\sum_{y=1}^{8}(I^d_{i,j}(x, y))^2 - N\mu^2_{ij}\right)
\end{aligned}$$

where the third step is using Eq. 2.2 and fourth step is using Eq. 2.4. So the variance (or local contrast) can be computed in the DCT domain.

Figure 2.5 (d) and 2.6 (d), shows the result which are same as for Section 2.3.1.

2.3.3 Wavelet Map

Intuition

A wavelet transform is roughly a gradient operation at various scales. It highlights "change" at various scales and this "change" is what the Itti's algorithm models. Thus the output of wavelet transform can be used as a saliency map. LH, HL and HH bands emphasize vertical, horizontal and oriented (45 and 135) edges and thus can be used.

Method

The Daubechies Wavelet Transform has been used to get a 1 level decomposition (LL, LH, HL, HH) . The saliency map is calculated by combining LH, HL and HH bands as shown in Figure 2.4.

2.4 Results and Discussion

Figures 2.5 and 2.6 compare the results. The results have been computed using gray level images. Surround Inhibit seems to bring out the most important areas which are supposed to catch the attention of the viewer. The results using variance and wavelet look comparable to Itti's algorithm if surround inhibit is not used. Complexity-wise, the proposed methods are much faster than Itti's algorithm. Following is the time comparison for various algorithms for a 256×256 image on MATLAB (v7.1, R14):

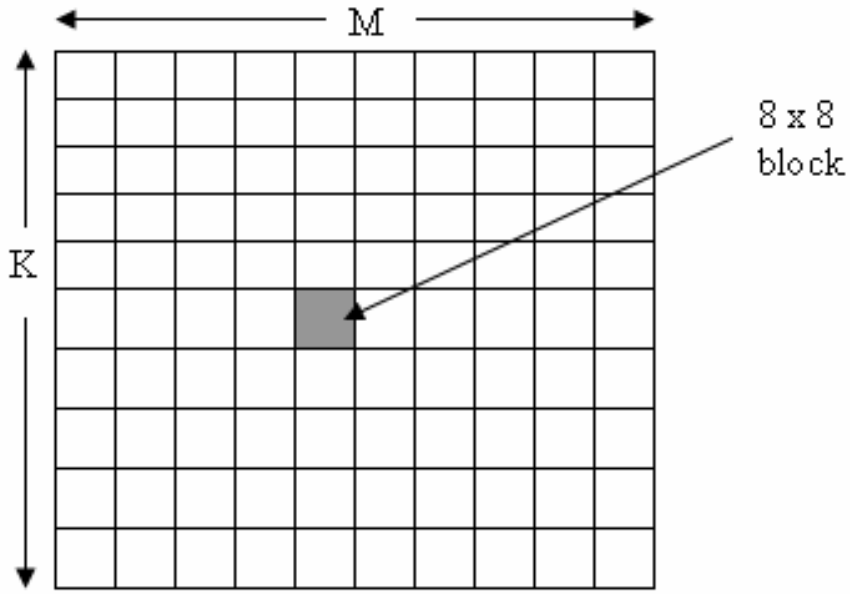


Figure 2.7: The image is divided into non overlapping blocks

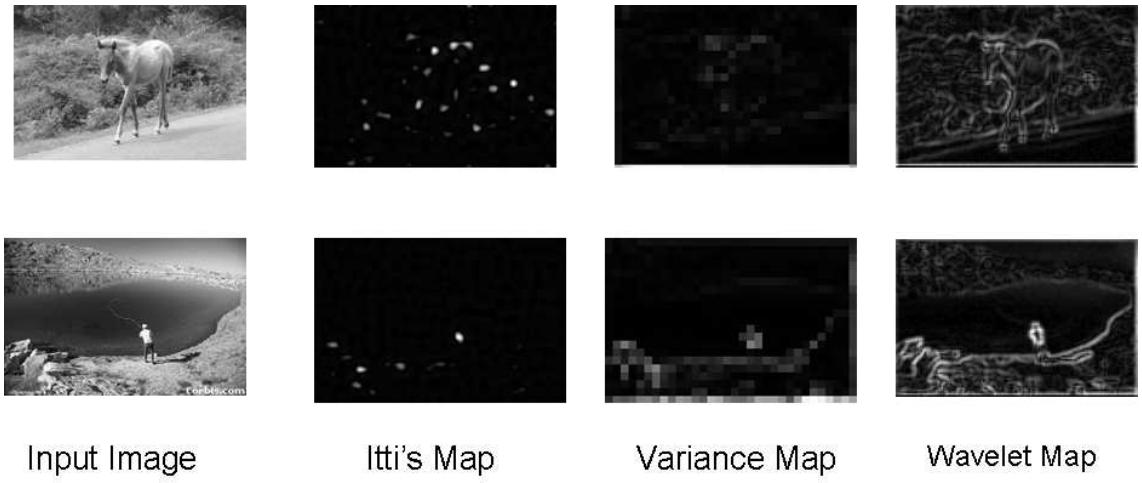


Figure 2.8: Comparison of results for the proposed method with Itti's algorithm. (a) Input Image, (b) Itti's Map, (c) Variance Map and (d) Wavelet Map

- Ittis algorithm : 6.8880 sec
- Variance : 0.3280 sec
- Wavelet : 0.2810 sec

The approach used by the proposed methods might not work if the background is highly textured or complex. Example are shown in Figure 2.8. The saliency suggested by all the maps are noisy, though the variance map seems to work better in the first image.

Itti's method gives equal emphasis to different low level features. It might be possible that for a particular image, some features are more important than the other. Parkhurst et al [24] found that the effect of local contrast is dominant when there are many high-contrast regions in the image. Otherwise other factors tend to dominate. They report similar results for two-point correlation. They found that in the images that have higher degrees of correlation, two-point correlation has a significant effect as a bottom-up stimulus. Thus, it might be worthwhile to be able to suggest the feature which will give a better saliency map for a particular kind of image.

2.5 Conclusion

In this chapter, we have discussed computationally efficient methods to generate saliency map. These saliency maps could be used to generate good thumbnails. Though a user study is needed to verify the claims, the results looks promising.

The saliency map is usually noisy and highlight some non-important regions. New approaches that give better saliency map might be helpful. It might be worthwhile to explore some other techniques (e.g. cross-correlation) apart from variance and wavelets to find the saliency map. Once we have a good saliency map, it can be used to get more semantic information about the foreground.

Chapter 3

Electronic Field Guide (EFG): The Prototype System

3.1 Introduction

The aim of the Electronic Field Guide (EFG) is to assist users in identification of the species of a leaf. This system is targeted for botanists or nature lovers who want to find more information about any new leaf they have found. The idea is that users can take the system (on a handheld computer) in the field and either use recognition or browsing features to help locate the closest match for a query image (photograph of an unknown leaf in the field) to the leaf images in the database.

EFG is a prototype system to demonstrate that computer vision based techniques are indeed helpful. We conducted a user study to test the usefulness of various features of the system and the results are discussed in Section 6. Assuming that the prototype version of EFG is scalable, the features of the EFG might make substantial improvements in search performance for a complete system with large image database. In the next sections various parts of EFG are discussed in detail.

Figure 3.1: The Electronic Field Guide (EFG) base version with random placement

3.2 Features

3.2.1 User Interface

The EFG user interface is based on PhotoMesa [31] which is a zoomable user interface (ZUI). It shows the thumbnails of images arranged in a 2D array. The user can Zoom-in and Zoom-out using the mouse (left/right) clicks. If the user wants to see more images of a particular species, he can use the left double click. Thus he can control the information he wants to see at any instant. These mouse based controls makes browsing and navigation easier. The base version displays the images in a

random order (Figure 3.1).

3.2.2 Similarity based clustering and display of data

The images are displayed so that similar images are closer together. This should help the user to better isolate the group to which the query image belongs. Once the correct group is isolated, he can zoom into that group to get more information about it. Thus the user need not browse the whole database. The results of user study shows that the clustering based placement makes it easier for the user to find the best match (Section 6). k-means is used for clustering which is discussed in detail in Section 4.

3.2.3 Visual Search

In the field, if the user wants to identify an unknown leaf, he can first use the browsing and navigation features which have been simplified using zoomable interface and similarity based clustering. Even then if he is not able to find the correct match, he can input the image of the leaf to the system. The system will match the input query image to the images of leaf in the database and will return best 20 matches based on outer shape (of leaves).

The matching is done using the method suggested by Ling and Jacobs [32], which gives more than 80% correct matches when the query image is from the database. If the top 20 matches are returned for a query image from the database, it guarantees the right result, provided the query image is photographed under

controlled conditions. The algorithm is discussed in detail in (Section 3.5).

We have found in a user study that this feature improves the performance of the system (for details see Section 6).

3.2.4 Text based Search

A user can see the images for a particular genus or species by inputting its name. Partial names are accepted. For example if the user wants to look at the images of species "saccharinum" (genus: "acer") but he is not sure of how to spell it, he can just type "s" and the system will show all the names which starts from "s". The user then can easily choose "saccharinum" and all the images of this species will be shown.

3.3 Database

The database consists of Type Specimen (Figure 3.2) and isolated leaf images (Figure 3.3). Type Specimen images usually have stems, flower, and multiple leaves. On the other hand, isolated images (as the name suggests) are single isolated leaves. The isolated images are for the plant species which exist on Plummers Island, which is a small island on the outskirts of Washington DC.

At present the database has over 1500 isolated leaves from 130 species. There are around 400 Type Specimen images for 70 species. All the images are taken in a controlled environment. Only isolated leaf images are used for matching.

Figure 3.2: Dried Type Specimen

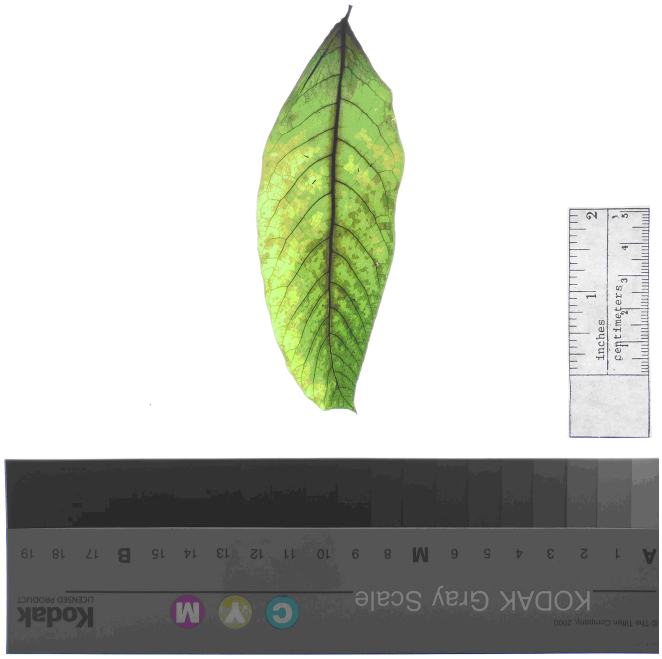


Figure 3.3: Isolated Leaf

3.4 Preprocessing

The shape matching algorithm requires contours of leaves as input. The contour is obtained by applying k-means clustering, with k equal to two, to the input image using the central and the border portion of the image as initial estimates of the cluster centers. Due to noise, there might be more than one foreground. In that case, the largest contour is the contour of interest. This takes care of small noise patches. The input and output of this step is shown in Figure 3.4.

3.5 Visual Search

A new algorithm for matching shapes, Inner Distance Shape Context (IDSC), developed by Ling and Jacobs [32] is used for Visual Search. The key idea here is to

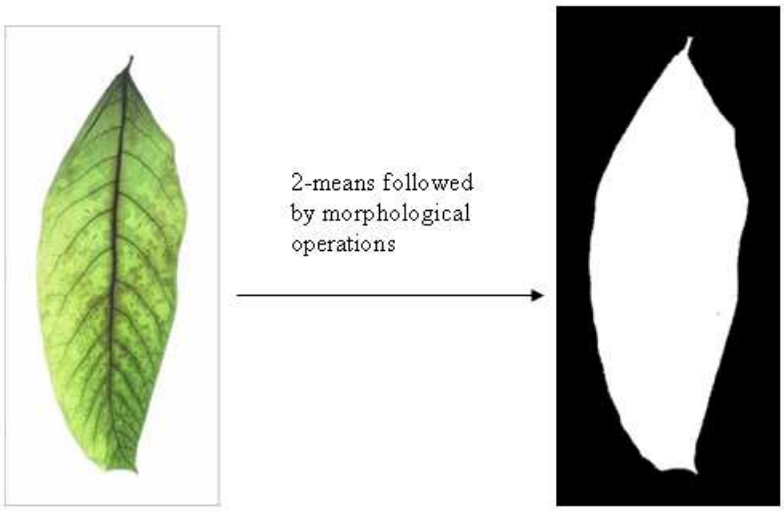


Figure 3.4: Isolated leaf before and after Preprocessing

use the inner-distance instead of the more standard Euclidean distance to describe leaf shapes.

Input to the algorithm is the image of the leaf. A leaf is represented by a sequence of points along its boundary. The boundary is obtained by the preprocessing step discussed in Section 3.4. After that, a distance measure named the inner-distance is computed for every pair of points. With these inner-distances, a histogram based descriptor, IDSC, is built that captures the shape of the leaf. This descriptor is then used to compare two leaves.

The key idea is that the inner-distance captures shapes better than the frequently used Euclidean distance. For example, in Figure 3.5, leaves (b) and (c) looks similar at first glance. After a detailed check, it is clear that (a) and (b) are more likely to come from the same species because they are composed of similar parts with different rotations. As shown below, the inner-distance can be effective

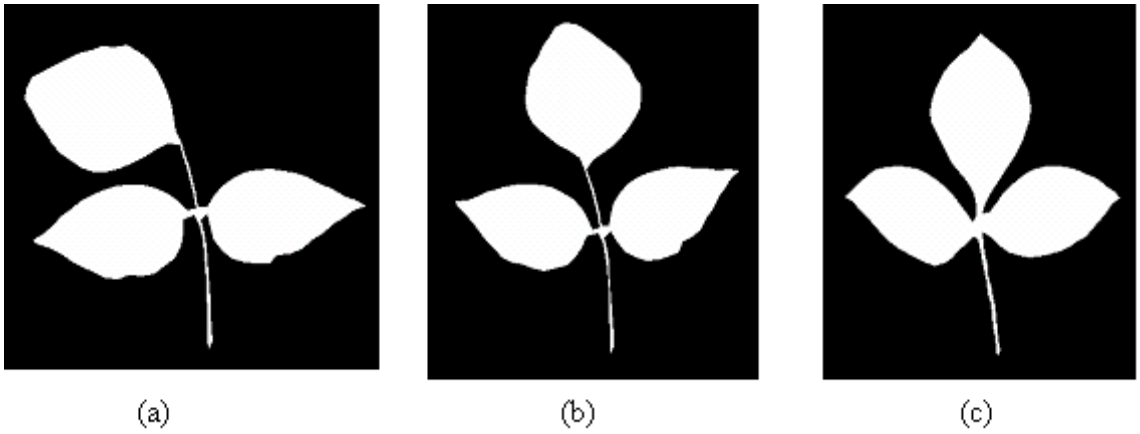


Figure 3.5: Inter-species similarity in leaves. (b) and (c) looks similar but only (a) and (b) are from the same species

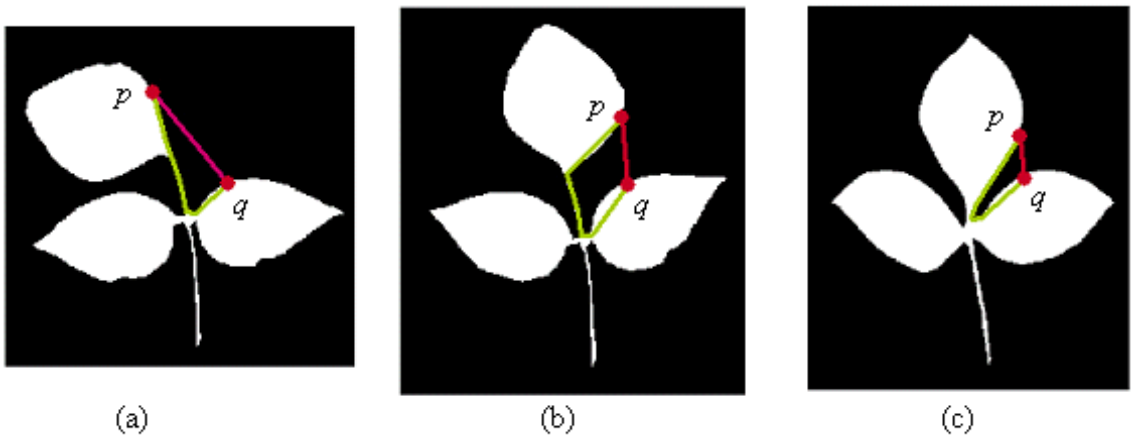


Figure 3.6: Inner Distance can be effective in distinguishing between similar species. (c) can be differentiated from (a) and (b)

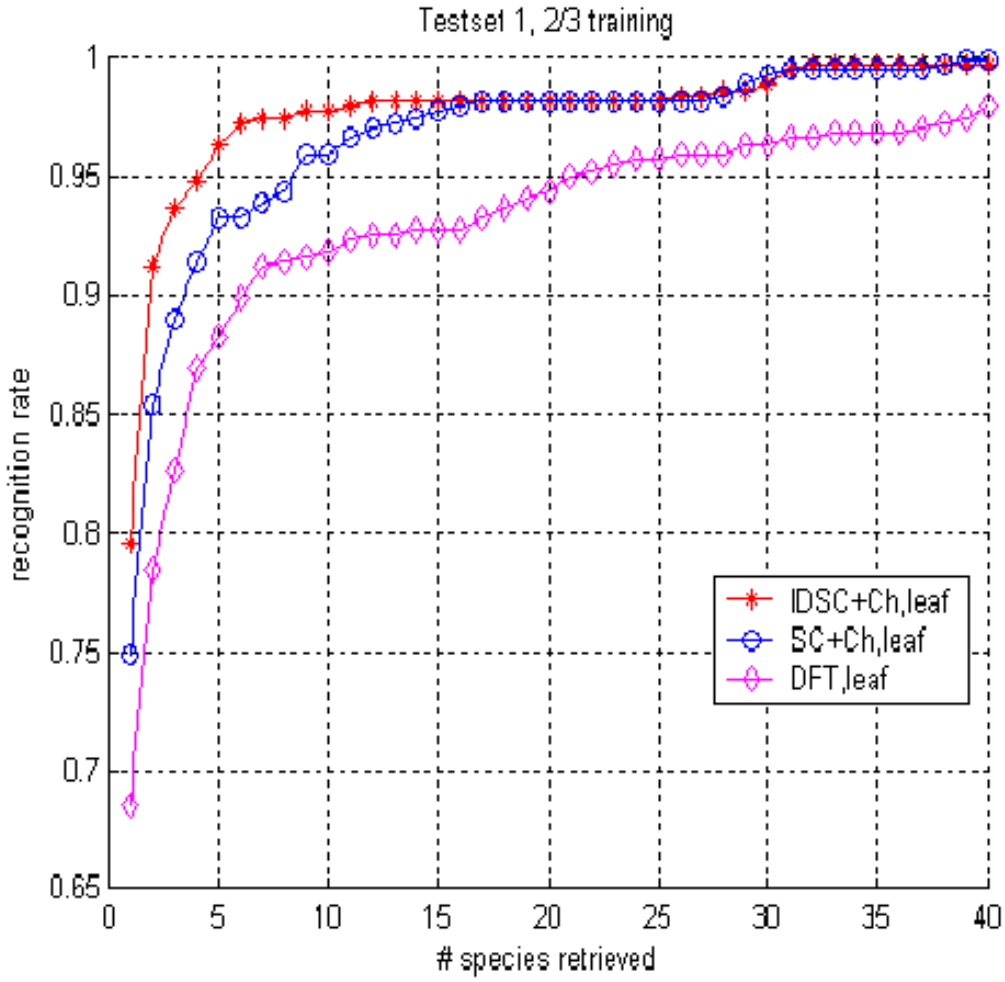


Figure 3.7: ROC Curve. This curve shows the effectiveness of the matching algorithm

for this case, while the Euclidean distance fails.

Traditional methods use the Euclidean distance to measure the distance between points on the shape. The Euclidean distance is defined as the length of the straight line between points, regardless of whether the line is within the shape or not. The inner-distance, defined as the length of the shortest path between points within the shape, can be more effective. For example, for the two given points p and q on Figure 3.6, the lengths of the red lines denotes the Euclidean distances, and green ones corresponds to the inner-distance (the green lines are actually the shortest paths between p and q). It can be seen that for the Euclidean distances, for (b) and (c) are more similar than (a) and (c). While for the inner-distances, (a) and (b) are more similar. Using these inner distance measures, the Inner Distance Shape Context (IDSC) is prepared, which is used to compare the shapes. The result is shown in ROC curves Figure 3.7, which is explained in the next paragraph.

The Receiver Operating Characteristics (ROC) curve is often used to measure the effectiveness of recognition/retrieval algorithms. In this case, the ROC curves are plotted as the recognition rate versus the number of species retrieved. For example, in Figure 3.7, the ROC curve for three different approaches is shown. For example a point on the IDSC curve, with its"x" coordinate being 5 and "y" coordinate being 0.96, the algorithm would return the right match among the top 5 matches 96% of the time. Figure 3.7 shows that this algorithm is more efficient than traditional ones.

The above mentioned shape matching technique is used in EFG for finding the best matches to the input query image. This method is more robust than any known

shape matching techniques and is helpful in improving the search performance of the system.

3.6 Conclusion

We have developed a prototype system, Electronic Field Guide (EFG), which is meant to help botanists identify leaves in the field. This system combines techniques from interface design to present data and computer vision to retrieve best matches. Zoomable interface, clustered data, visual and text based search are some of the features of this system. Clustering based placement schemes are discussed in more details in Section 4 and Section 5.

We have conducted a user study to evaluate the benefits of this system. The results are presented in Section 6. We also conducted a field test to check the performance of the EFG in field conditions. For results and discussion, refer Section 7.

Chapter 4

Clustering

4.1 Introduction

A random placement of images as shown in Figure 3.1 can create a lot of problems if one has to search for an image. Intuitively, placement by similarity might be helpful. Liu et al [6] showed that indeed this is true. They found that placement based on global similarity and grouping is more helpful as compared to a random placement.

Rubner et al [11] suggested using Multi-Dimensional Scaling (MDS) [12] to define the placement scheme. MDS is used to reduce the dimensionality of a vector space. Suppose there are n objects with distance δ_{ij} between them as a p-dimensional vector. We want to find a vector $\hat{\delta}_{ij}$ of dimension d such that $d < p$ (in our case d=2). Kruskal suggested doing this by minimizing STRESS defined as:

$$\text{STRESS} = \left[\frac{\sum_{i,j} (\hat{\delta}_{ij} - \delta_{ij})^2}{\sum_{i,j} \delta_{ij}^2} \right]^{1/2} \tag{4.1}$$

Figure 4.1 shows the results of MDS on our dataset (Section 3.3). The output of MDS can have overlap as the size of image is not taken into account while finding the placement. Figure 4.2 shows the results of modified MDS when the resulting coordinates of MDS are approximated with rectangular grid coordinates. For this, exhaustive search is used to find the nearest available rectangular grid coordinate

Figure 4.1: Placement using MDS on Smithsonian database

Figure 4.2: Placement where MDS output have been corrected to remove the overlap

for a MDS generated coordinate.

4.2 k-means

MDS based placements does not give any specific boundaries to distinct groups. A potentially better approach is to cluster them so that similar shape images are displayed closely. This idea is exploited to give a clustered display. The algorithm used is explained below.

4.2.1 Algorithm

- Calculate the distance matrix using the IDSC [32].
- Use k-means to cluster the database in k groups (k=10 is chosen heuristically).
- Use the Quantum Tree Map [31] algorithm to decide the layout (as in PhotoMesa).

The following describes the **k-means** algorithm:
Suppose there are n images in a group and we want to make k clusters. Let $C_i, i \in 1....k$ denote the k centers. Let G_i be the i^{th} group. Let r be the index for the images $r \in 1...n$. Let M be the number of iterations. Let $d(r, C_i)$ be the distance between the r^{th} image and C_i center based on the similarity measure (IDSC [32]). The steps of the algorithm are:

1. Randomly select k centers C_i from the feature vectors of n images.

Figure 4.3: Placement when the images are grouped in 10 clusters based on similarity of shapes

2. Form groups G_i:

   for i = 1 to k

   $G_i = \{r, d(r, C_i) < d(r, C_l), l = 1...k, l \neq i\}$

3. For M iterations {

   for i = 1 to k

   $G_i = \{r, d(r, C_i) < d(r, C_l), l = 1...k, l \neq i\}$

   for i = 1 to k

   $C_i = center(G_i)$

}

Figure 4.3 shows a cluster representation for k=10. Note that the clusters are good at representing a particular type of shape.

We have used the above groups (also known as method B) in the user study to find the usefulness of this arrangement as compared to the random placement (also known as method A in the user study). The results are encouraging and are discussed in Chapter 6.

Chapter 5

Large Image Databases

5.1 Introduction

For large databases, browsing and navigation seems difficult. If we place thumbnails of 1000 images on a 800×600 resolution screen, nothing will be clear. Even if the saliency based thumbnail cropping algorithm is good and we have the best cropped thumbnails, there is a maximum limit of thumbnails that can be shown on a fixed size screen. Figure 5.1 shows an example with around 1200 images in 5 groups. In this example, it's difficult to make out anything from the initial placement. This makes choosing the right group unfeasible at this scale (level) of thumbnails. Users will have to zoom into some area to get any meaningful information, making the thumbnails useless at this scale (level).

To overcome this problem, Chen et al introduced [5] a hierarchical browsing approach by using a pyramid structure. The full database is represented by the base of the pyramid and as we go up towards the top, each image represents a collection of images. Thus towards the top, each image in a layer is like an "icon" (representative image) which represents more images of a similar type. They suggested that instead of displaying all the images, use representative images from the groups. These would be less in number and would be clearer on the screen. Once the user is able to identify the group, he can double-click on that group to see more images from

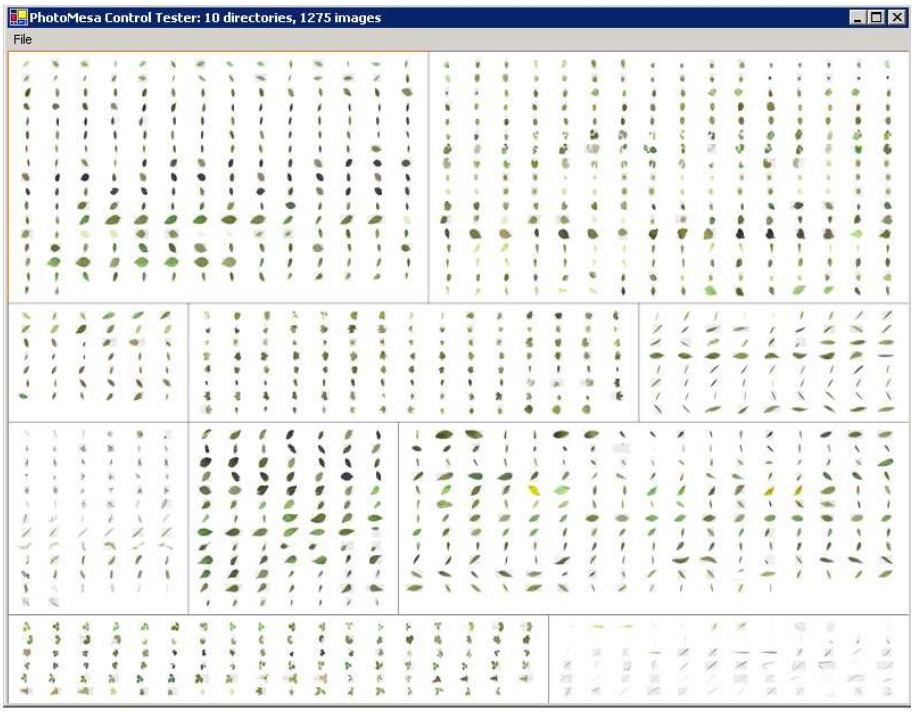


Figure 5.1: Around 1200 images are arranged in 800 × 600 screen size in 5 groups

that group.

They have suggested both top-down and bottom-up clustering approaches. The hierarchical placement of data is intuitive though they have not provided any empirical evidence if this placement schemes improves the browsing experience. Using the EFG (Section 3), we have conducted a user study to provide empirical evidence for this scheme.

5.2 Details for EFG

For the image set of 1500 images and 130 species, we have used a three layered hierarchical placement of data as shown in Figure 5.2. In the figure n_k represent the k^{th} group in the n^{th} layer. The algorithm used to find the first two layers is:

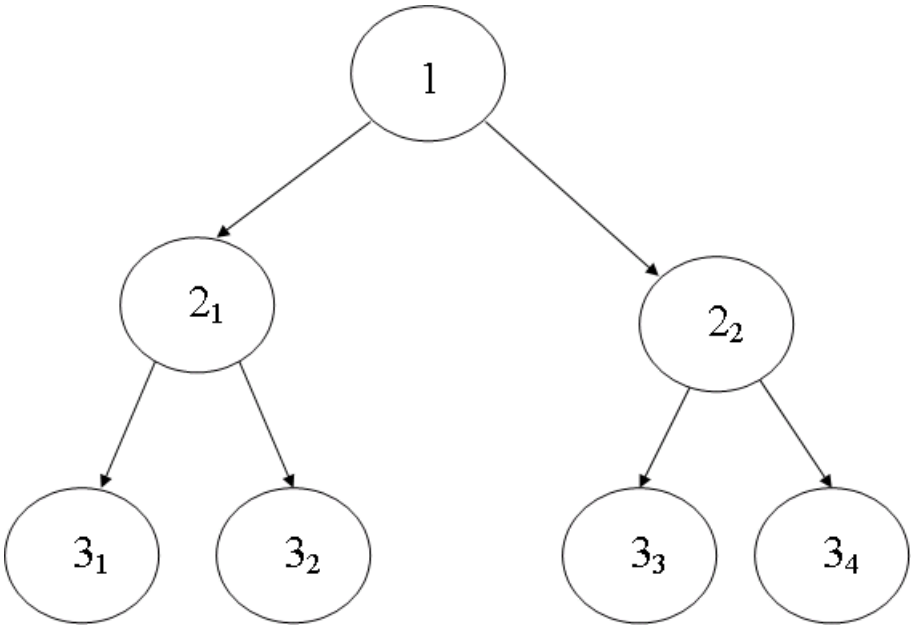


Figure 5.2: Hierarchical placement of data showing 3 layers. $1_1 = 2_1 \cup 2_2, 2_1 = 3_1 \cup 3_2, 2_2 = 3_3 \cup 3_4$

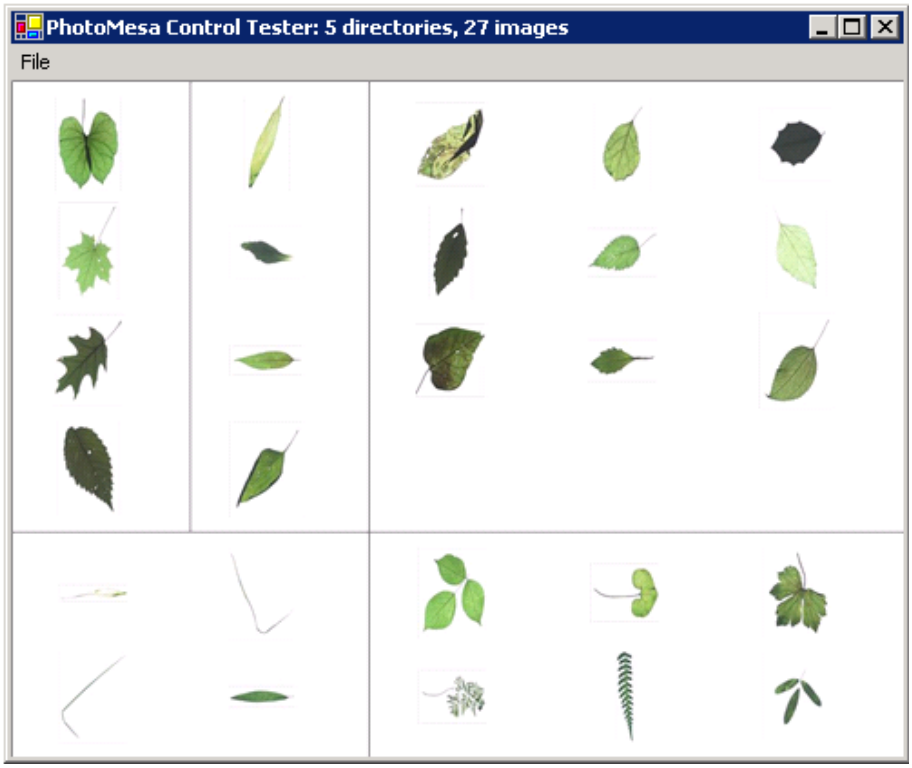


Figure 5.3: Initial placement (level one) with 27 images and 5 groups

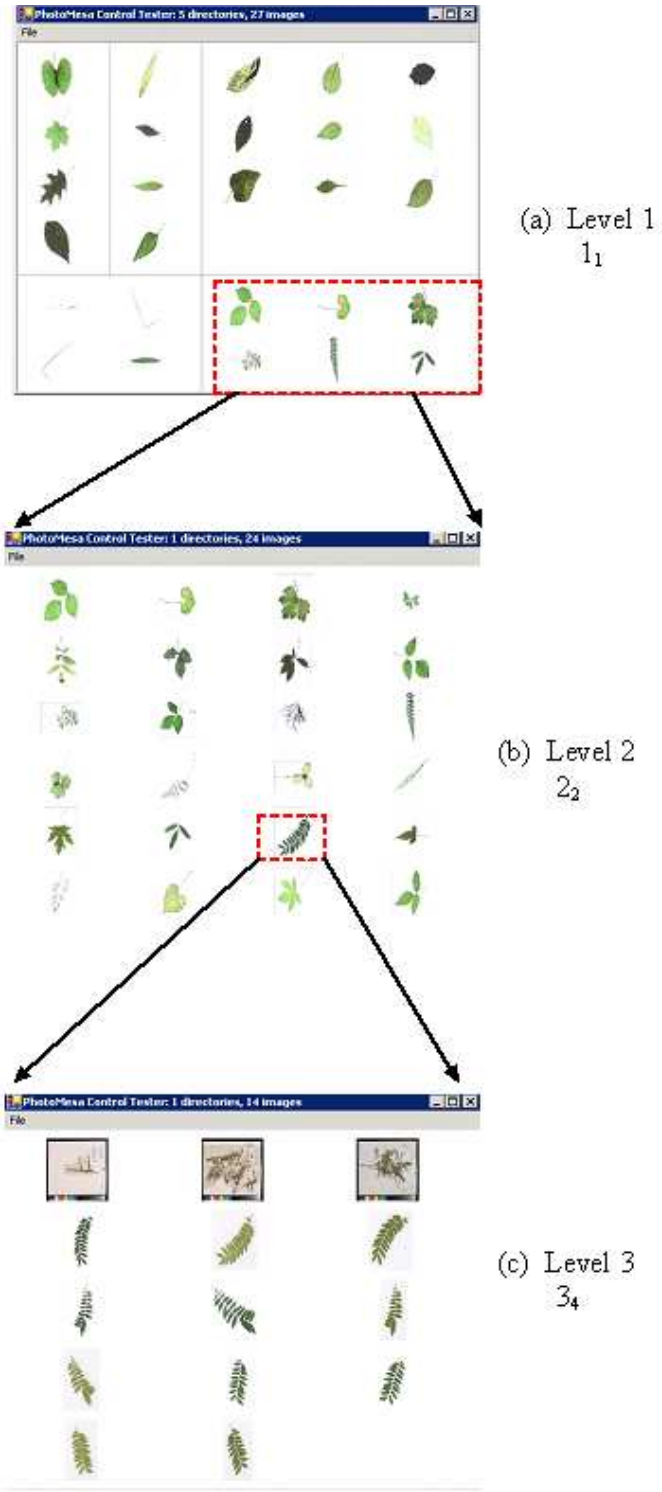


Figure 5.4: All the three levels for our database (Hierarchical placement). (a) Level 1. Initial placement (level one) with 27 images and 5 groups (b) Level 2. A group has been shown fully, 24 images. (c) Level 3. All images of a species have been shown, 14 images

1. Make the groups $G_i, i = 1....k$ (as done in Chapter 4)

2. Each group represents one circle of level 2 (see Figure 5.2 and Figure 5.4). For each group n_k, r_k are chosen using k-means such that the size of r_k represents the size of n_k i.e.

$$\frac{sizeof(n_k)}{sizeof(r_k)} = \text{constant} \quad \forall\ k \tag{5.1}$$

These r_k represents the level 1 in our case (as shown in Figure 5.3 and Figure 5.4). 1_1 is 5 and contains 27 images as shown in Figure 5.3. Figure 5.4 shows all the three levels of the hierarchical placement using the above algorithm. Using this placement scheme, more than 1500 images are available in a systematic way.

This placement scheme (called L2 in the user study) is used for the user study with reference to the random placement(called L1 in the user study). The screen size for 130 images has been kept 500×400 as compared to the usual 800×600 for the same number of species. The screen size is 2.4 times less and the thumbnails are the same size as with a 800×600 but more than 300 images. For this placement also the images in L1 are not clear which signifies the need for hierarchical placement. The results of the user study and the response of the users are discussed in Sec 6.

Chapter 6

User Study

6.1 Introduction

The system is targeted to assist botanists and amateurs in the identification of leaves. For a query image of an unknown species of a plant, this system provides a software interface to the leaf image database, to help users identify the matching leaf. In the present system, Computer Vision based techniques have been used to modify placement schemes of images on the screen and to provide the top-k matches for a query image. In case of a large database, a hierarchical placement based on similarity is used.

We are interested in knowing the answer to the following questions:

- Is the system useful to botanist/novice users?
- Are they comfortable with such a system?
- Are they able to find the correct answer in a limited time?
- Is the computer vision based technique better than the random placement?
- Do we have a better solution for large databases?

The time taken to find the correct match and the accuracy of the match for the query image will determine the usefulness of the placement techniques (of database

images on the screen).

6.2 Technology

A Zoomable User Interface (ZUI) has been used for this project. The interface can be fully controlled by computer mouse. The screen size of the interface is 800 x 600. The images of isolated leaves are placed on this screen and the user can zoom in and out using the left and right mouse click. Using left double click, more images of a particular species could be seen. Right double click brings back the initial placement screen.

Two 1.5Ghz (Intel Centrino) Windows XP laptops with atleast 512 MB RAM, with ordinary mouses were used. The screen resolution was 1024 x 764 pixels for both the laptops. The correctness of the answer and the time taken to find the answer was recorded. Participants were given 120 seconds to select the correct answer on each trial.

6.3 Database

The database (Section 3.3) consists of over 1500 images of isolated leaves of 130 species of plants. There are multiple images of each species. 40 images (of different species) were used as query images. These images were not used in the interface window.

6.4 Various methods

We conducted a controlled user study and compared the performance of various organizations of images for a specific query image (these methods are explained in earlier sections). For the first part of the study the placements were (Set1):

1. Random placement (method A)
2. Clustered placements (method B)
3. Top-k matches (method C)

For the second part of the study the placements were (Set2):

1. Random placement (method L1)
2. Hierarchical placement (method L2)

The difference between A and L1 is that in L1 the screen size has been kept smaller (500x400 instead of 800x600) keeping the number of images displayed the same. This simulates the effect of more images on a fixed size screen.

6.5 Participants

There were 21 volunteers for the User Study (3 male and 18 female, 18 to 60 years old). All volunteers were related to the Botany department of the Smithsonian Institute (employees or interns). 14 volunteers were botanists or had training in botany. All volunteers had experience with computers. For Set 1 data from all 21 subjects have been used. For Set 2 data from 19 subjects have been used. One

subject had problems using method L1 because of the small size of images. Another subject got confused in the hierarchical placement of data in L2.

6.6 Procedure

Each session lasted 30-40 minutes. Each user filled out a survey to determine their computer usage background and their experience with leaves. After explaining the interface controls, users were given hands-on experience on the controls of the interface. An animated demo showed the exact way the task has to be executed followed by more hands-on experience of the actual task. The query images used in training were not used for the test. The training was repeated if the user was not comfortable with the system.

6.7 Tasks

The task was to find the best match for the query image using the ZUI. The study was divided into two parts. In the first part, 15 query images were shown one after another. For each query image, the placement used was either A, B or C. In the second part, 10 query images were shown one after the other and the placements were L1 and L2. The placements (A, B, C) or (L1, L2) were shown in random order. A particular query image was repeated once for each method A to L2 (5 times) for 5 consecutive users. This ensures that all the query images were subjected to same number of different placements. For a user 25 different query images were shown (ensuring that no user saw same query image twice). There

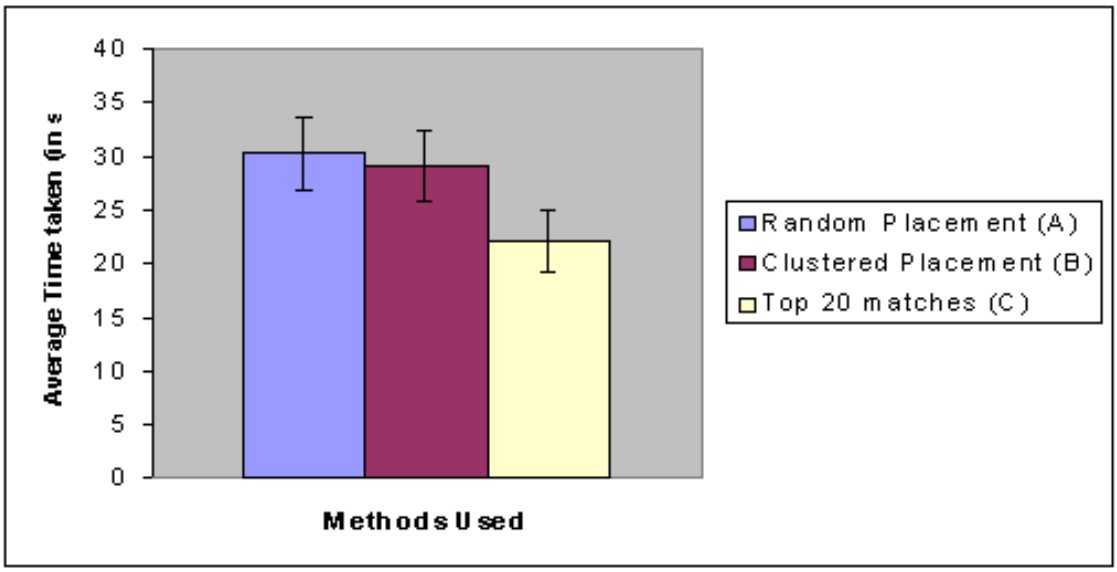


Figure 6.1: Timing Comparison for A (random), B (grouped), C (top-k). Clearly C outperforms A and B

were total 40 query images. Using this rotation method for 20 users, would ensure that each query image has been used for equal number of organization schemes (A to L2).

6.8 Results

The independent variables are the methods used A, B, C (Set1) or L1, L2 (Set2). The dependent variables are the time taken and the accuracy.

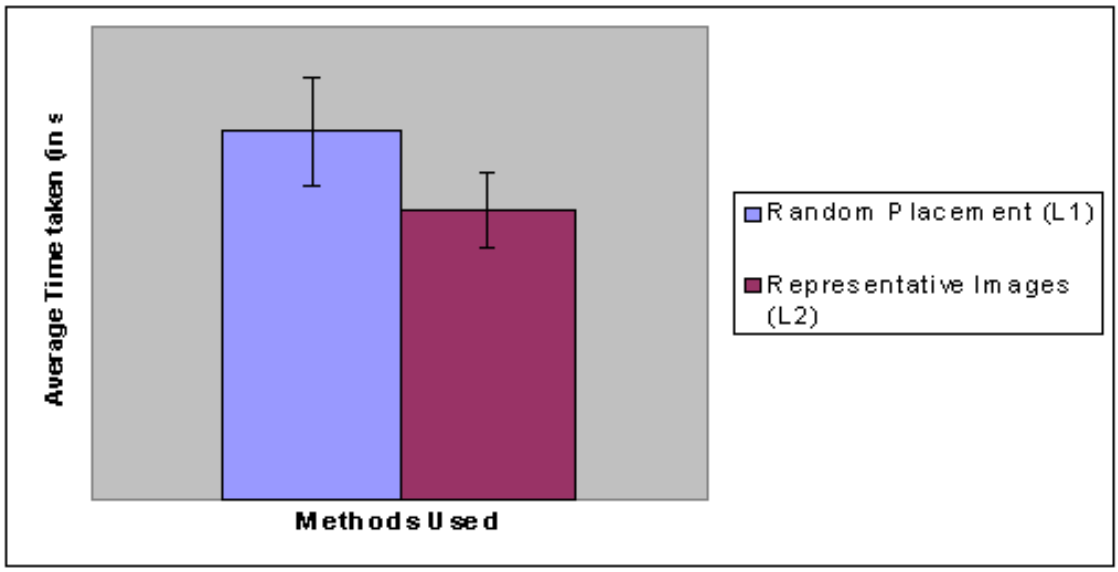


Figure 6.2: Time comparison for L1(random) and L2(hierarchical). L2 is better than L1

Figure 6.3: Accuracy plot for A (random), B (grouped), C (top-k). B and C outperforms A

6.8.1 Timing Comparison

Set1 (random, clustered and top-k)

Figure 6.1 shows the timing results for Set1 (A, B, C). The timing information for only the correct matches (in 120 seconds) have been considered for this comparison. Top-k (k=20) matches outperforms the random and clustered placement. Statistically significant results are obtained for Set1. One way ANOVA shows that the null hypothesis (A=B=C) can be rejected (F=7.63, p=0.001). Tukey's post-hoc analysis shows that C is statistically significant compared to A and B.

Though Clustered placement is not statistically significant from random placement, on an average it performs better.

Set2 (random, hierarchical)

Figure 6.2 shows the timing results for Set2 (L1, L2). Hierarchical placement outperforms the Random placement. Statistically significant results are obtained for Set2. Paired T-test showed that the time taken in L1 and L2 are statistically different with p=0.026.

6.8.2 Accuracy

Figure 6.3 shows the accuracy for A, B, C. For Set 1, 252 answers out of 315 (80%) were correct. The individual percentage of correct matches for A, B and C are:

- Random placement (A): 71.43%
- Clustered placements (B): 83.81%
- Top-k matches (C): 84.76%

One way ANOVA shows that the null hypothesis (A=B=C) can be rejected (F=3.94, p=0.025). Tukey's post-hoc analysis shows that the accuracy of C (top-k) is statistically significantly different from A (random). The accuracy of B (clustering) is moderately significantly different from A (random) with p=0.06.

For Set 2, 143 out of 190 answers (75.26%) were correct. The individual percentages of correct matches for L1and L2 (the difference is not statistically significant) are:

- Random placement (L1): 72.63%
- Hierarchical placements (L2): 77.89%

6.9 Usability

All the 21 users answered positively about the user friendliness of the system. Users were comfortable with the system after training. Users liked the displaying of more images by left double click (one user said," Left double click gives more information...this is intuitive and in line with windows basic mouse controls"). When asked whether they would like to use the system on a laptop or a PDA, they were enthusiastic and willing to do so.

6.10 User Comments (Subjective)

Users were asked their preference (on a scale of 1-9), rating for each method. Apart from this there were question such as which method they found best and why. For Set 1, users preferred clustered (6.8/9.0) and top-k matches (7.9/9.0) over the random placement (4.6/9.0). One way ANOVA shows that the null hypothesis of the subjective rating (A=B=C) can be rejected (F=26.03, $p < 0.000$). Tukey's post-hoc analysis shows that A, B, C are all statistically different.

Users found top-k (C) good, as the size of the images were bigger (because of the fewer number of images displayed, 20 instead of 130) and the computer generated matches were usually accurate. Many users said they would prefer clustered placement over others as the clustered arrangement (B) allowed them to narrow down the group quickly and all the species were available (unlike C), in case they would like to see some other group. Particularly, they didn't like random placement (A) as the users found it difficult to look through the whole screen for the right match.

For Set 2, users preferred hierarchical (7.2/9.0) over the random placement (3.8/9.0). Paired T-test showed that the subjective rating of L1 and L2 are statistically different with $p < 0.000$. Users found the number of images for random placement "overwhelming" and the size of images "too small". They found that hierarchical placement "speeds up the filtering" and they have fewer images to navigate.

6.11 Discussion and Conclusion

Though many volunteers had training in botany, the task was primarily more of searching and shape matching. No significant difference was seen in the performance of botanists and non-botanists, so prior knowledge is not very useful in these tasks.

For Set1, top-k matches were found to be better than both random and clustered placements techniques both in terms of time and accuracy. This is likely because clustered placement (B) allows user to find the possible group thus narrowing down the possible option. Top-k matches displays only 20 images which are the best matches according to the shape matching algorithm. There are two advantages of this. First, the leaves are much bigger in size (as 20 instead of 130 are displayed), thus user need not zoom in a lot. Second, the narrowing down has already been done which helps in a speedy match. But all this is true if the matching algorithm is robust. As one user mentioned, top-k (C) is "hit or miss".

Though clustering is marginally better than random placement in terms of time taken, the accuracy improvement makes it an obvious choice over random placement.

For the final system, it looks reasonable to use a combination of clustered placement (B) and top-k matches (C).

6.12 Large Databases

For large databases, when the number of species is large, if all the images are displayed on the screen, it would be difficult to identify anything. In Set2, L1 tries

to simulate that effect. L2 presents the data in hierarchical way. The hierarchical method gives statistically significant better results than random placement. This is likely because the hierarchical method shows few large images for each group thus making the initial identification of each group easier. If the user is able to identify the right group, he would be able to find the right match.

Chapter 7

Field Test

7.1 Introduction

The primary purpose of the Electronic Field Guide (Section 3) is to help botanists identify a leaf in the field in a simple manner. The original system has leaves photographed in the lab in controlled conditions. Figure 7.1 shows the lab set up used to photograph the leaves. At present we have around 1500 leaf images from 130 species. All these leaves are from Plummers Island, a small, well-studied island in the Potomac River. Once the query leaf is photographed, k-means is used to classify foreground and background. The boundary of the foreground is the input to the contour matching algorithm which matches the input contour to the database and return the top 20 species (Figure 7.2).

7.2 Photography in the field

The challenge is to be able to handle the variability of photography conditions in the field, which makes the segmentation of the leaves difficult. To achieve this, lab like controlled conditions can be imposed in the field, but that would make the photography process cumbersome, thus hurting our efforts to keep the system simple.

To overcome this problem, we experimentally came up with moderate con-

Figure 7.1: The present database has been photographed under controlled lighting conditions. One H20 back on Hasselblad 502 with 80mm lens has been used to get images of resolution 3600x5000 pixels

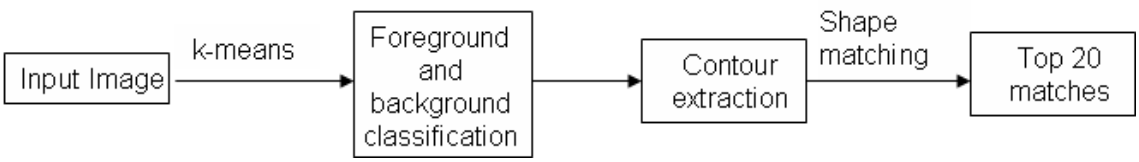


Figure 7.2: Various steps for finding the best match to the input image. First the foreground is extracted using k-means. The boundary of the foreground is used for shape matching to retrieve top 20 species from the database

trolled conditions which might give us uniform lighting thus aiding good segmentation. The following things are required for the present set up:

- Uniform colored background which is different from the foreground (green leaf). Suggested colors are light gray, yellow.
- Some pins to keep the surface of the leaf flat. Leaves could also be pressed for some time to flatten them.
- Any normal digital camera.

There are two precautions that should be taken care while photographing the leaves:

- The lighting should be as uniform as possible.
- The leaf should be flat.

Figure 7.3 shows an example. Note that the pins are completely inside the leaf and the lighting is more or less uniform. Also, the leaf is flat.

Using these simple conditions, we are able to get segmentation results that allows robust recognition. Figure 7.4 shows the foreground (in white) and Figure 7.5

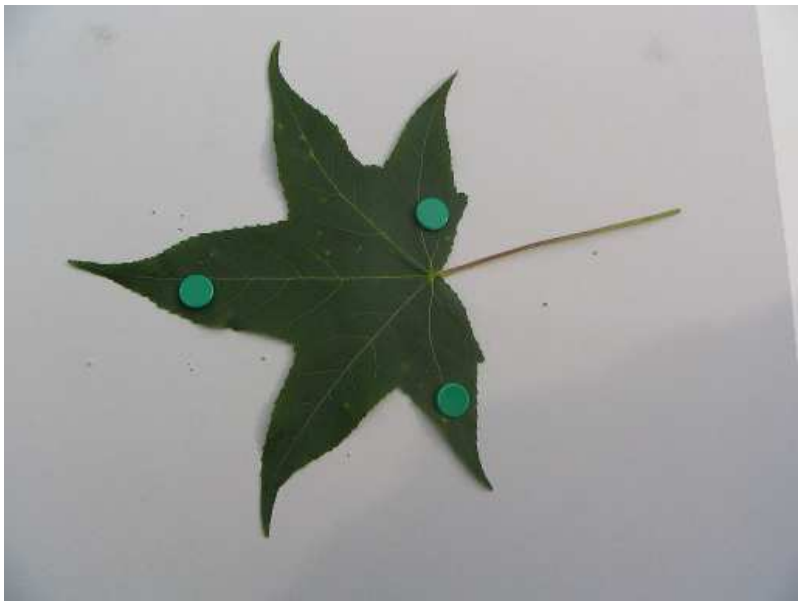


Figure 7.3: Image of a leaf photographed under moderatly controlled conditions. Note that the pins are completely inside the leaf and the lighting is more or less uniform

Figure 7.4: Results of thresholding and k-means to find background and foreground

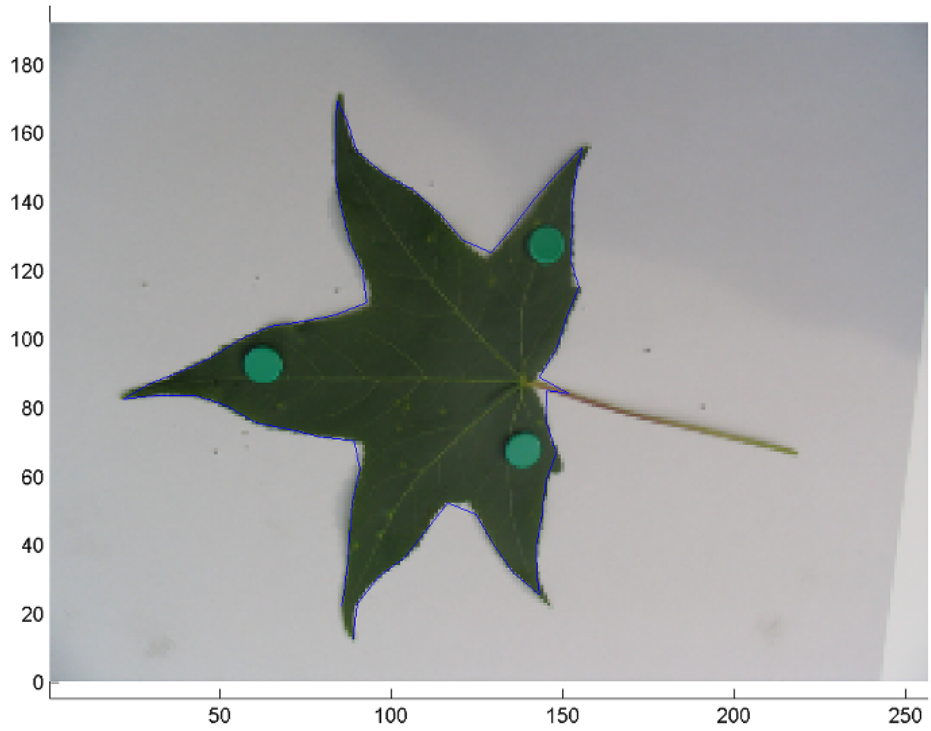


Figure 7.5: Once the foreground is found, the contour of the foreground is the required contour. The contour is shown in blue

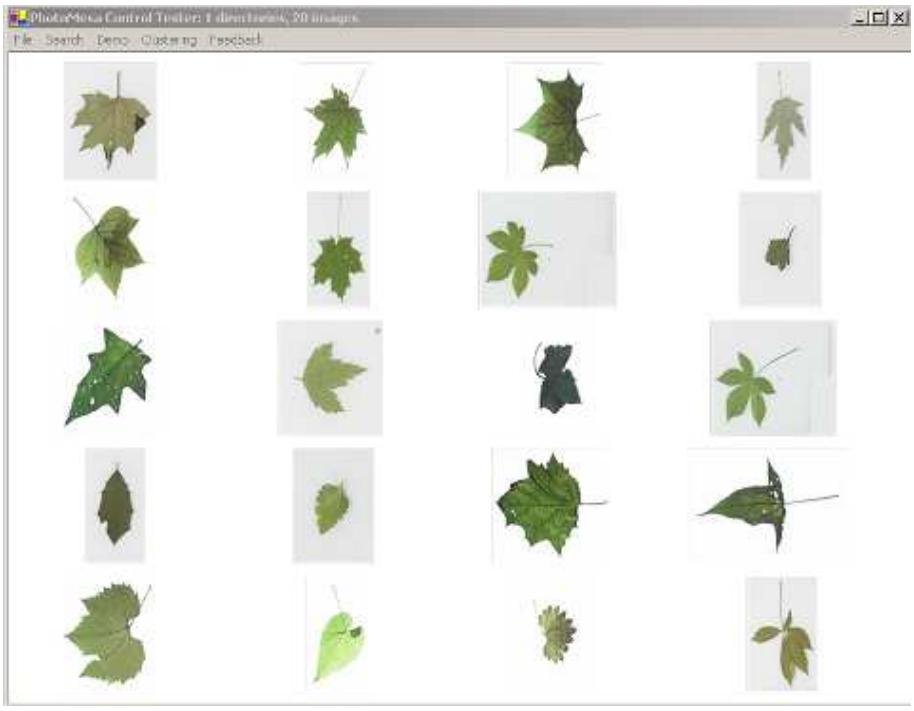


Figure 7.6: The top 20 retrieval results using the suggested photography conditions.

The species of this input image is not on the database, but the results are good

Figure 7.7: Leaves are being collected for the field test

shows the contour. Figure 7.6 shows the retrieval results. The input image is not in the database, but the retrieved species have similar shapes as that of the input image.

7.3 Test and Results

The suggested set-up has been field tested on Plummers Island. The team consisted of researchers and students from the Smithsonian Institute, University of Maryland and the Columbia University.

One set of leaves were collected from the island (Figure 7.7), arranged (Figure 7.9), photographed under moderately controlled conditions (Figure 7.10). Another set of leaf images were taken under no controlled conditions (Figure 7.11). In the second set, the leaves were not plucked. These images were entered in the sys-

Figure 7.8: Leaves after collection

Figure 7.9: A leaf is being made ready for photography

Figure 7.10: Leaves being photographed under moderately controlled conditions

Figure 7.11: Leaves being photographed under uncontrolled conditions

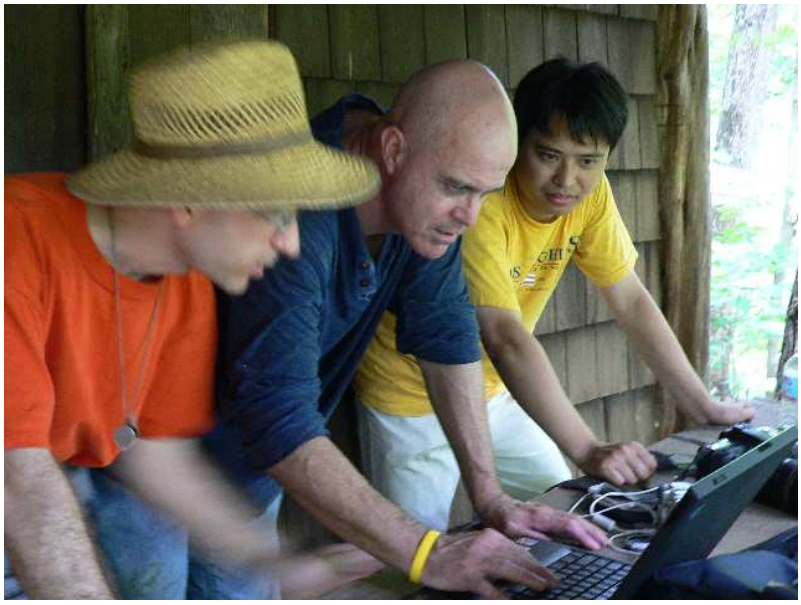


Figure 7.12: Online testing of the system

tem and the retrieval results were observed (Figure 7.12). Figure 7.13 - Figure 7.30 shows the results.

7.4 Discussion

The results are good overall. Figure 7.13 shows images of leaves photographed under moderately controlled lighting/boundary conditions. The foreground segmentation is usually good. Figure 7.14 - Figure 7.21 shows the retrieval results when the images are photographed in controlled condition. If the image is in the database, correct matches are retrieved, otherwise the shape of the retrieved images are similar to the query image. A better method to make the background system more robust is to take a photograph of the background without the leaf and then place the leaf and take another photograph. This would give us more accurate statistics

of the background thus resulting in better background subtraction.

Figure 7.22 shows images of leaves photographed under uncontrolled lighting/boundary conditions. In this case usually the background is complex. Most of the time the foreground extracted is noisy, though a portion of foreground reflects the actual shape. This portion helps in retrieving the similar shaped images. Figure 7.23 - Figure 7.27 shows the retrieval results when the images are photographed in uncontrolled condition. Though the results are good for uncontrolled conditions, its not clear if the system is robust for these type of scenarios.

The software environment used to do shape matching is MATLAB. The original system was developed and tested on version 7.0 R14 of MATLAB. The version used at Plummers Island was version 6.5 R13. Due to version change, the results were not as good as expected when tested on the field. This degradation in results is possibly due to implementation/thresholds changes in the in-built functions (from R13 to R14) in MATLAB. The results reported in this thesis are using the correct version of MATLAB (version 7.0 R14).

7.5 Conclusion

We conducted a Field test to check the utility of the system in real life conditions. The problem is challenging and we have shown that with some controlled conditions we can get good results. For completely uncontrolled conditions though we got retrieval results, the contours are noisy which questions the robustness of the system for these types of images. Robust background subtraction techniques are

required which can handle the variability of the lighting conditions and if possible of the complexity of the background.

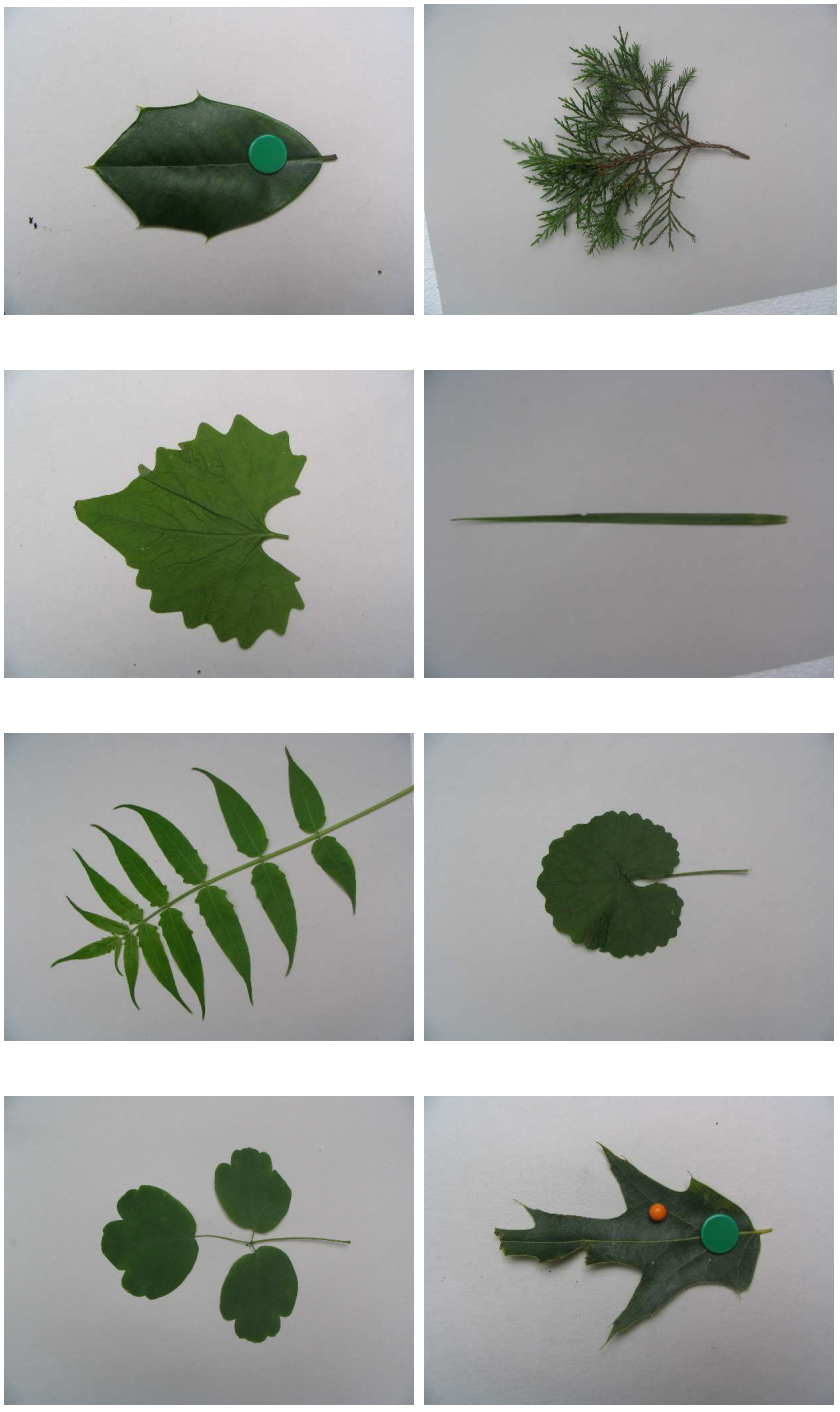
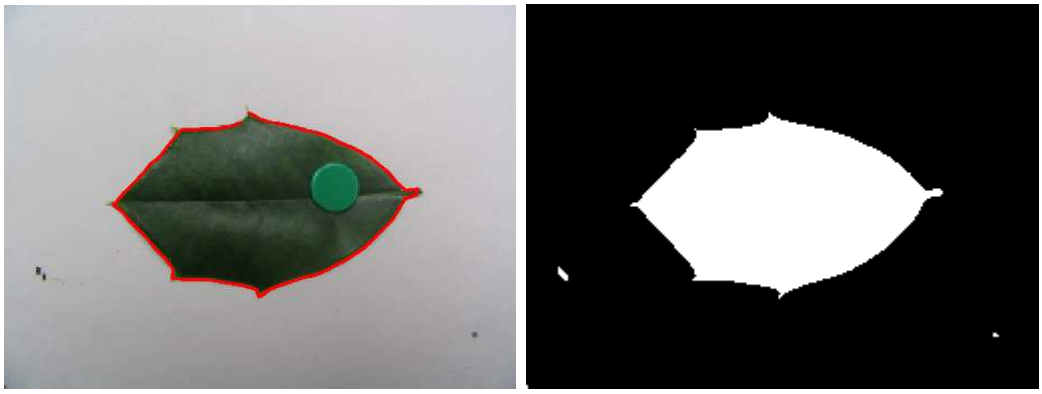
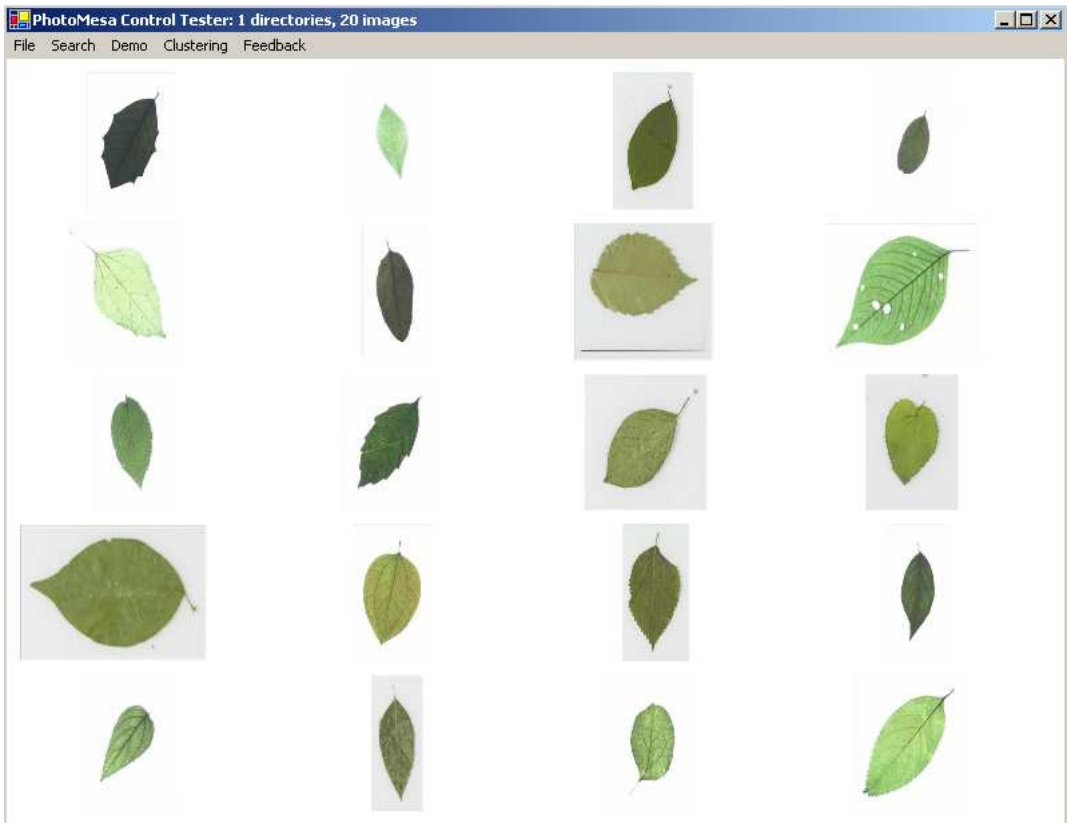


Figure 7.13: Input Images. Moderately controlled conditions

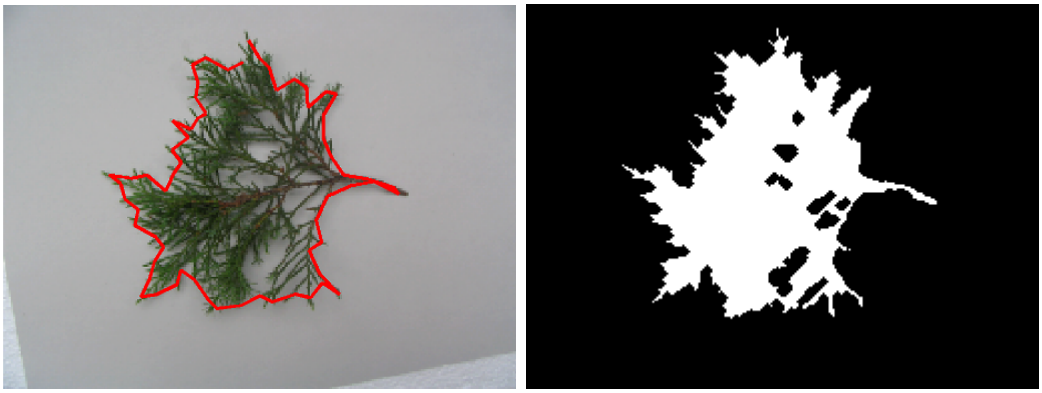


(a) Original Image with contour (b) Foreground and Background

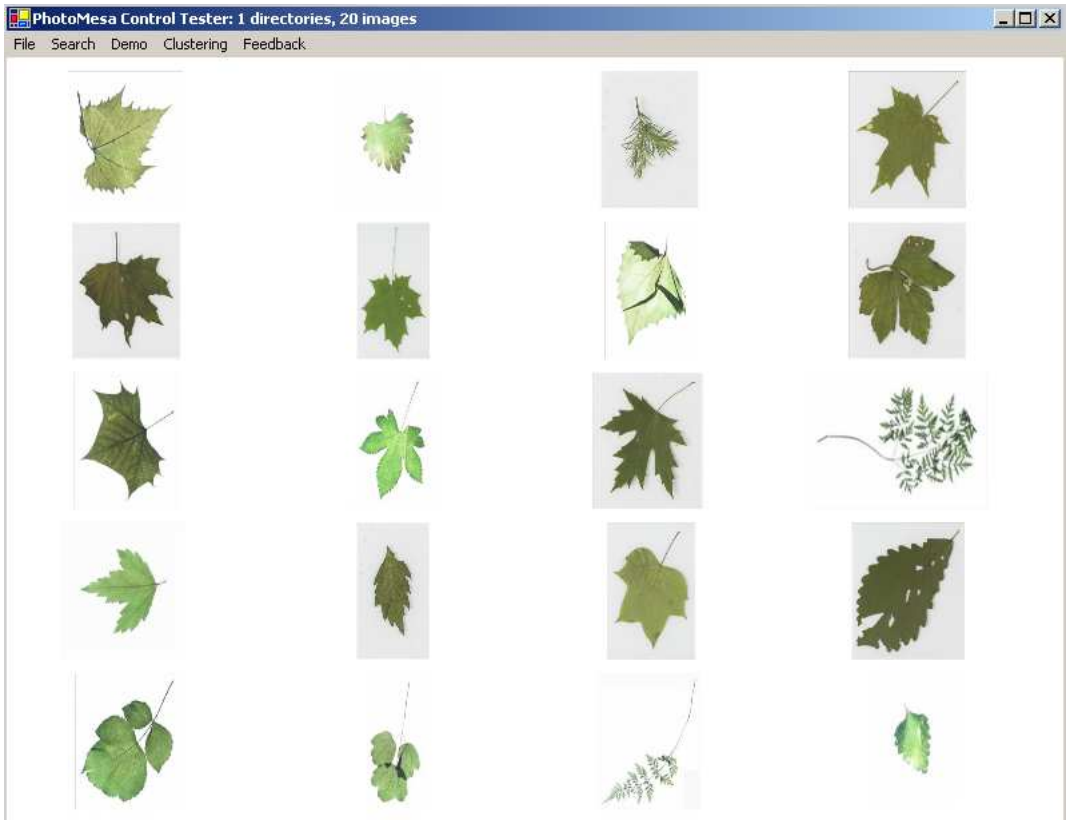


(c) The right match is the first image (1st row)

Figure 7.14: The search results using EFG. The correct match is in top 20



(a) Original Image with contour (b) Foreground and Background

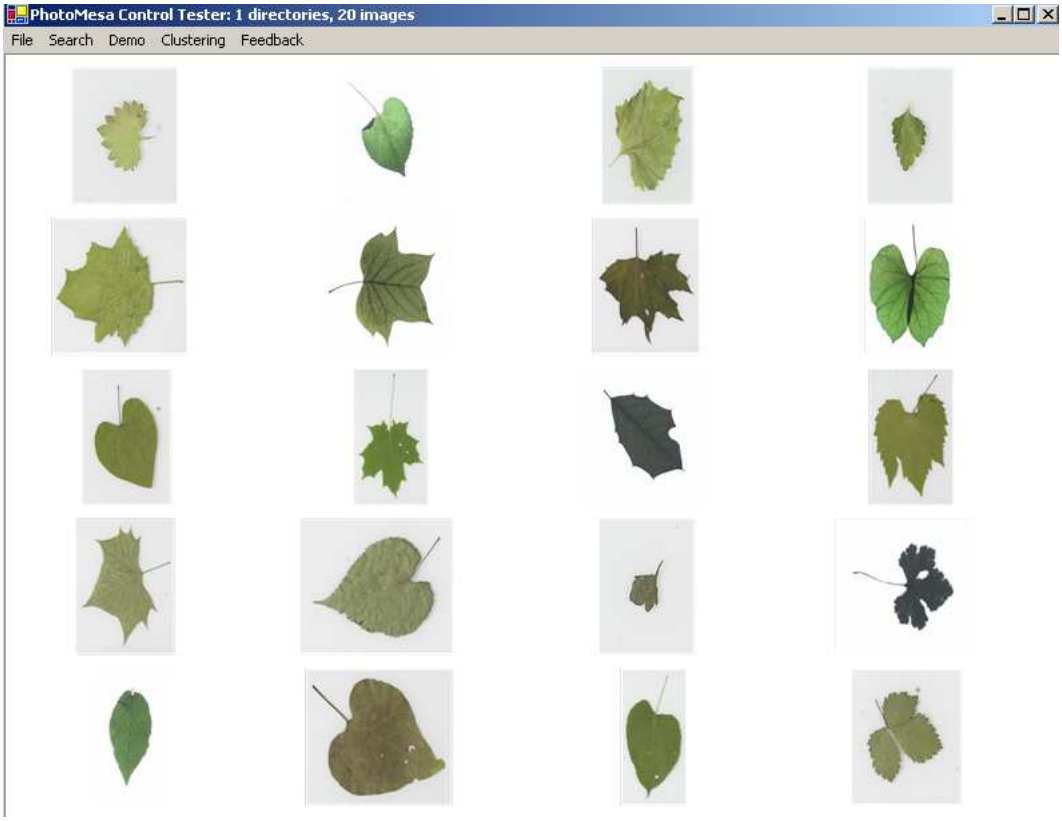


(c) The right match is the third image (1st row)

Figure 7.15: The search results using EFG. The right match is the third image (1st row)

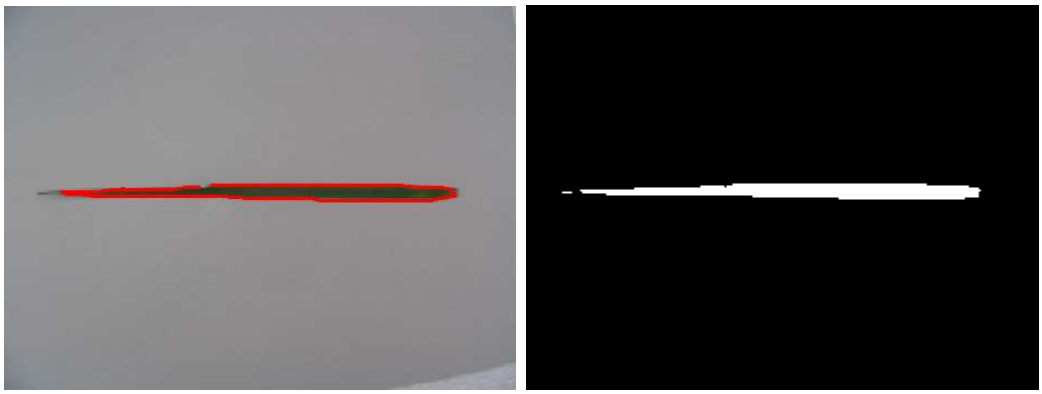
(a) Original Image with contour

(b) Foreground and Background

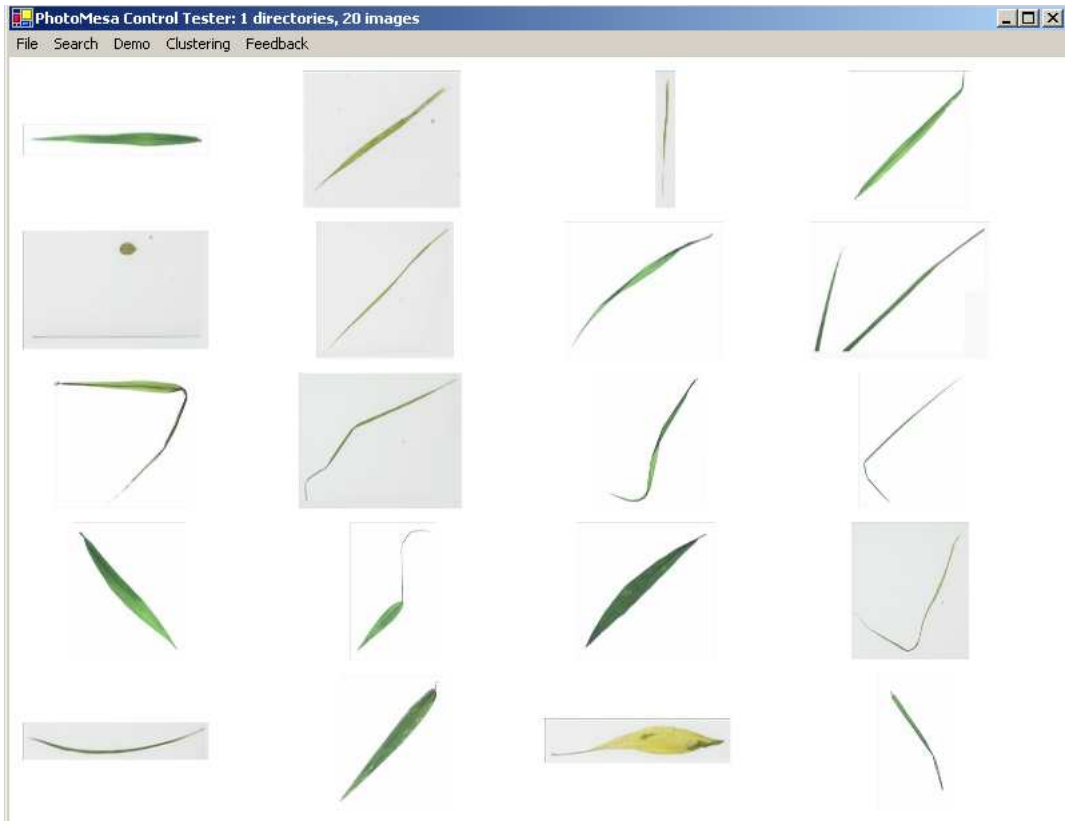


(c) The right match is the third image (1st row)

Figure 7.16: The search results using EFG. The correct match is in top 20

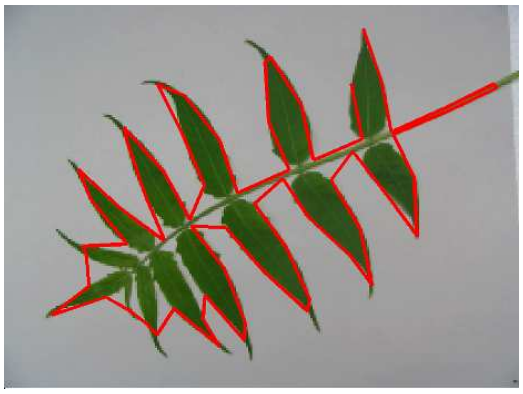


(a) Original Image with contour (b) Foreground and Background

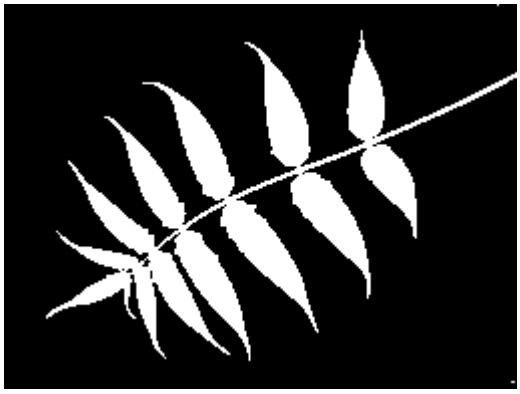


(c) The results retrieved are similar to the input query image

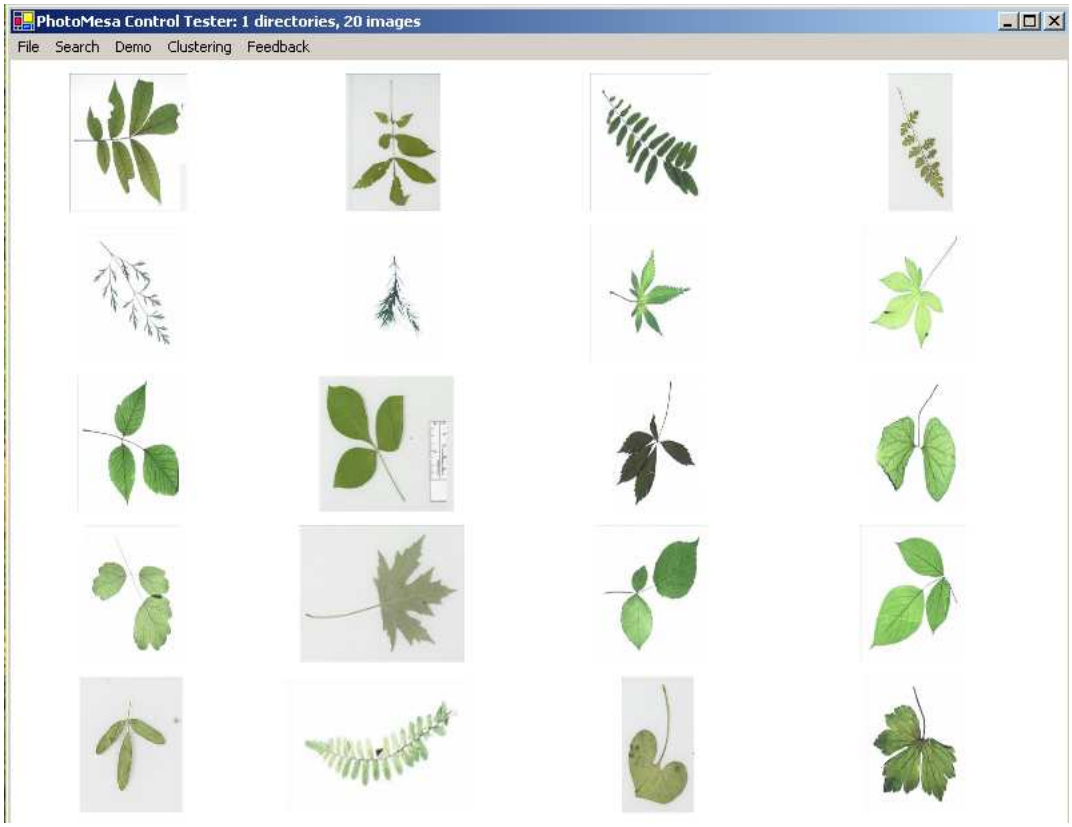
Figure 7.17: The search results using EFG



(a) Original Image with contour

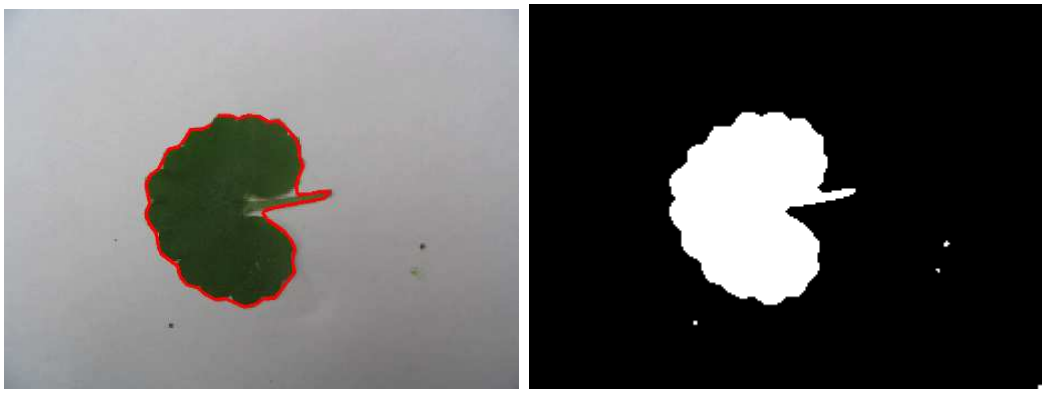


(b) Foreground and Background

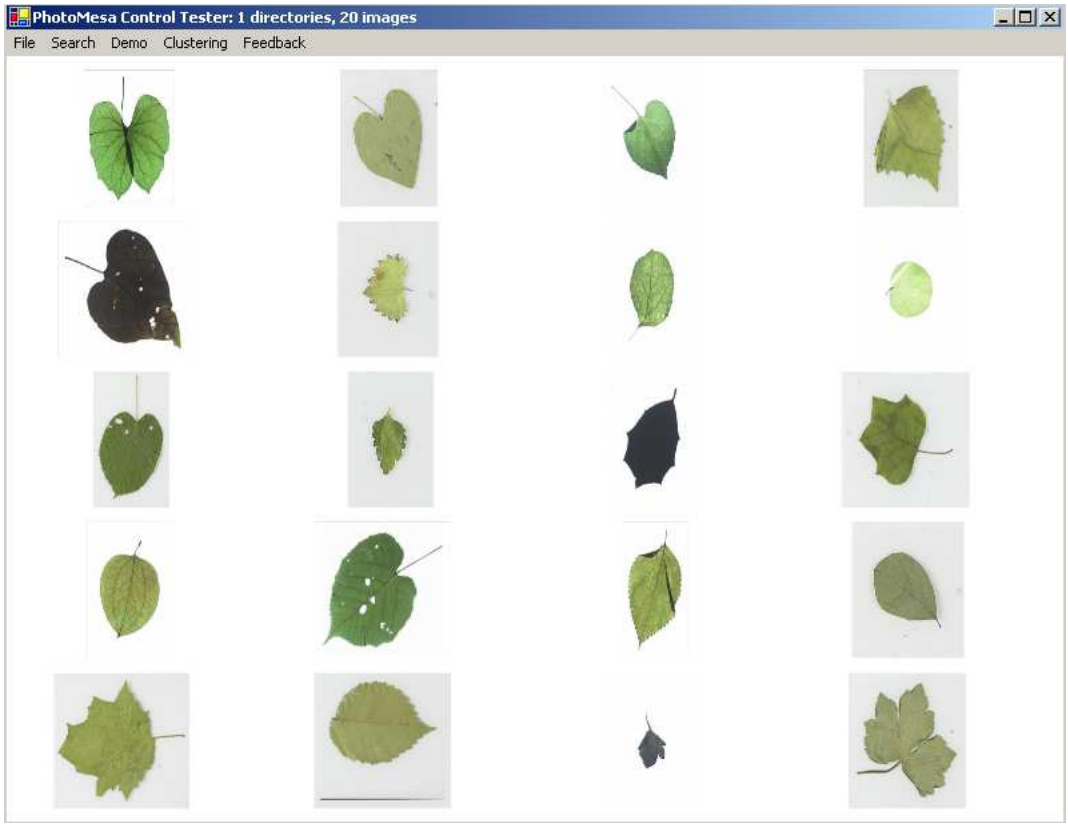


(c) The results retrieved are similar to the input query image

Figure 7.18: The search results are good

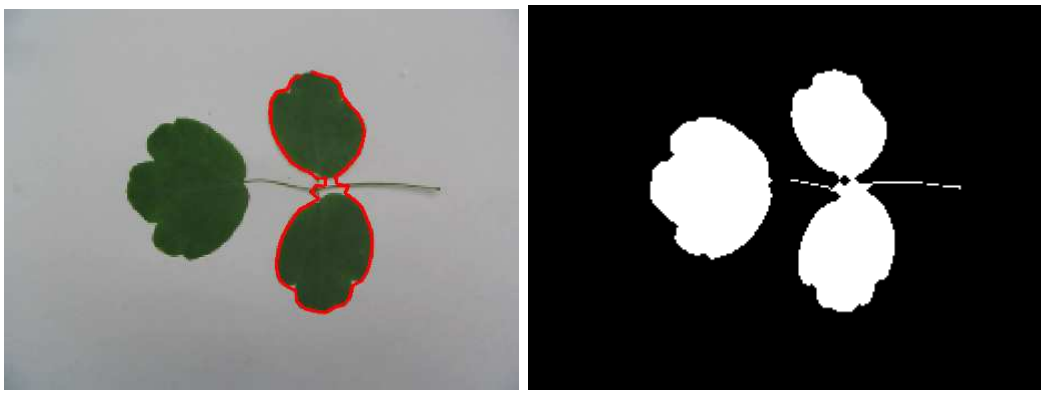


(a) Original Image with contour (b) Foreground and Background

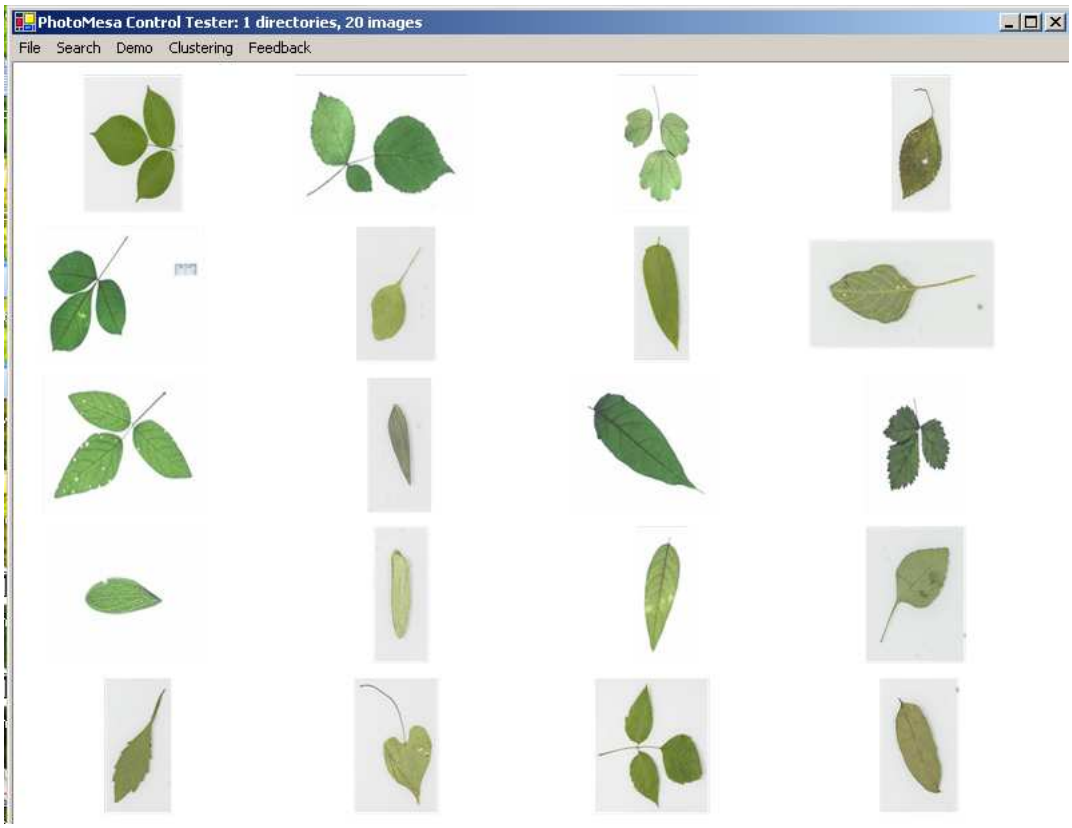


(c) The right match is not in the database but the closest matches are shown.

Figure 7.19: The system is able to retrieve similar shaped leaves



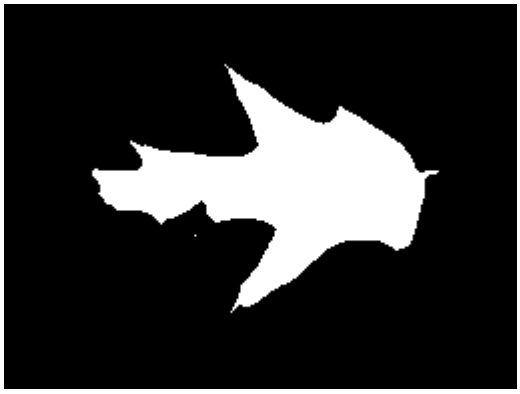
(a) Original Image with contour (b) Foreground and Background



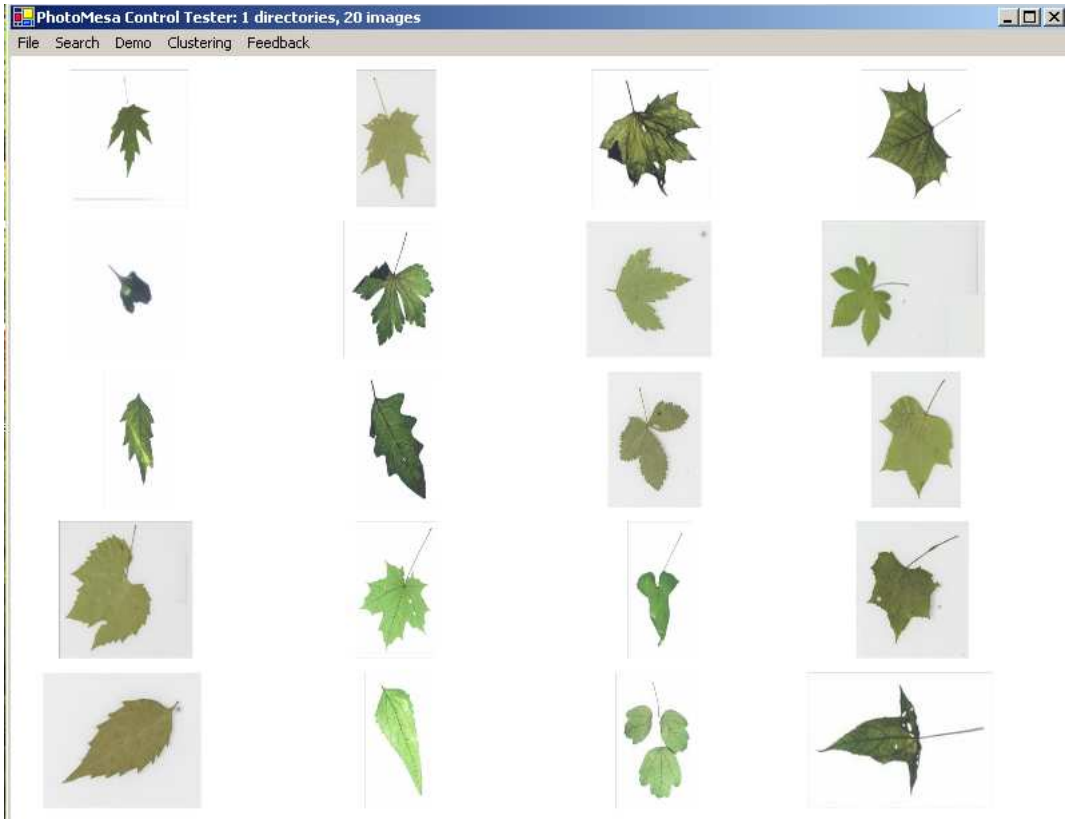
(c) The right match is the third image (1st row)

Figure 7.20: Only two leaves are in the contour as the foreground is broken. These are due to uneven lighting conditions

(a) This leaf is damaged.



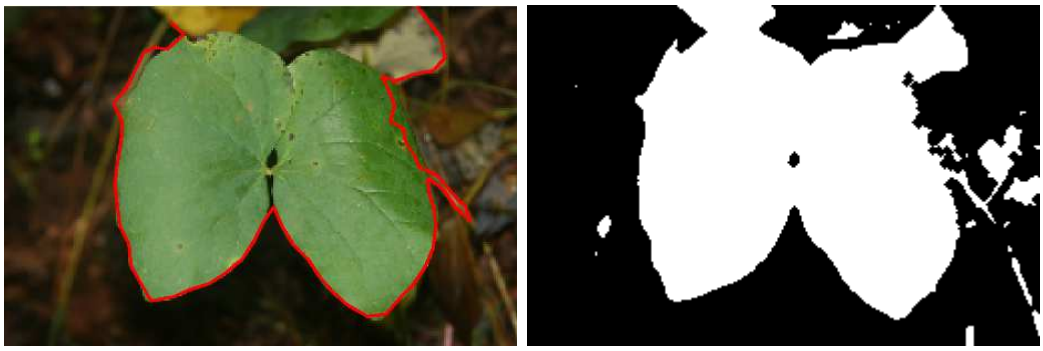
(b) Foreground and Background



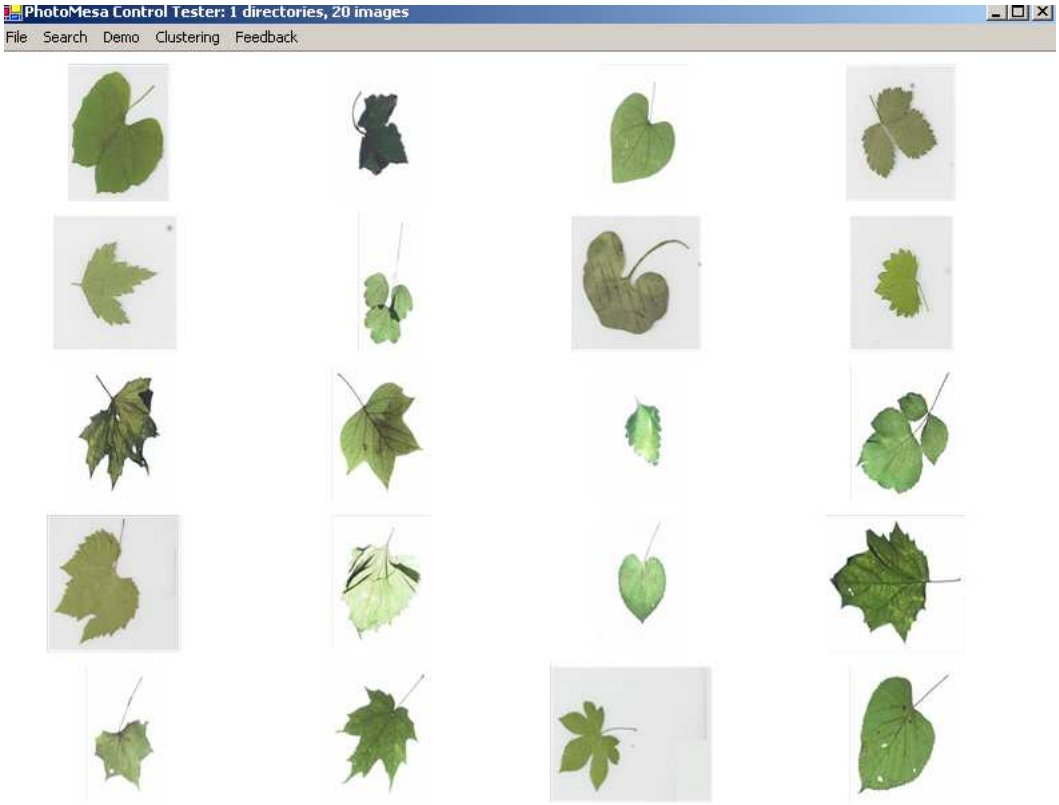
(c) The right match is not in top 20. This shows that system fails in case of damaged leaves.

Figure 7.21: The search results using EFG. Example when the leaf is damaged

Figure 7.22: Input Images. No controlled conditions



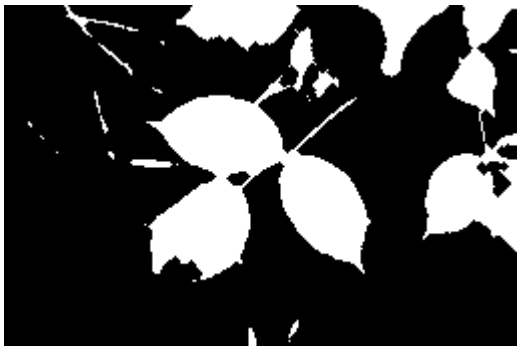
(a) Original Image with contour (b) Foreground and Background



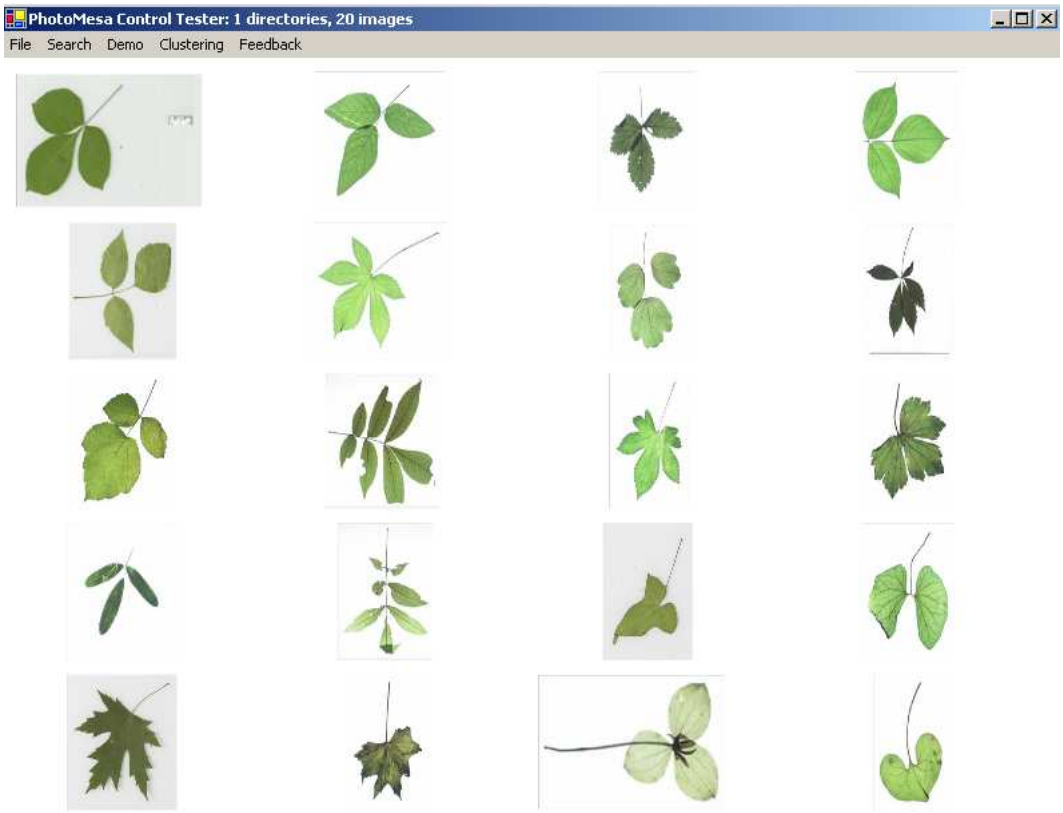
(c) Looks like the first image is the correct match

Figure 7.23: The search results using EFG for images with no controlled conditions. Looks like the first image is the correct match

(a) Original Image with contour

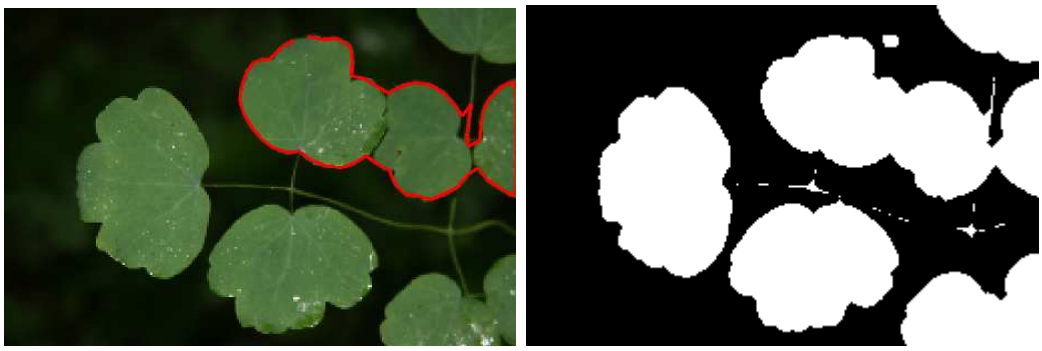


(b) Foreground and Background

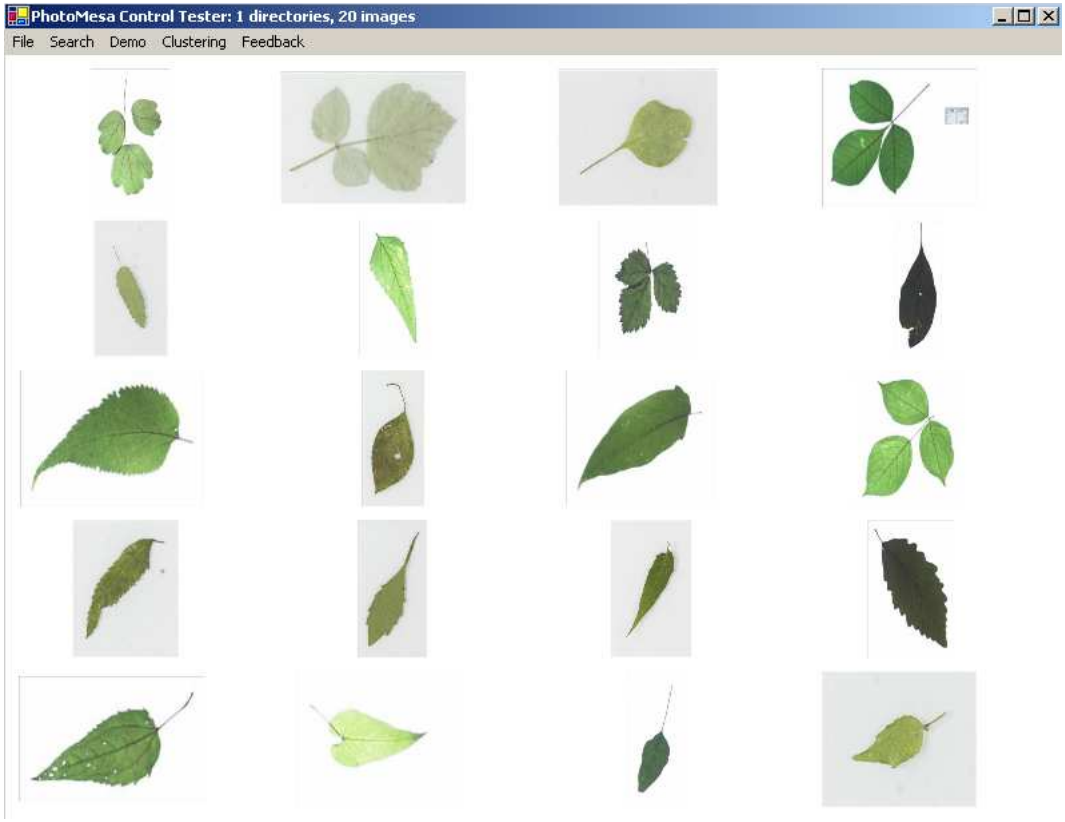


(c) Species with similar shape are retrieved

Figure 7.24: The search results using EFG for images with no controlled conditions. Species with similar shape are retrieved

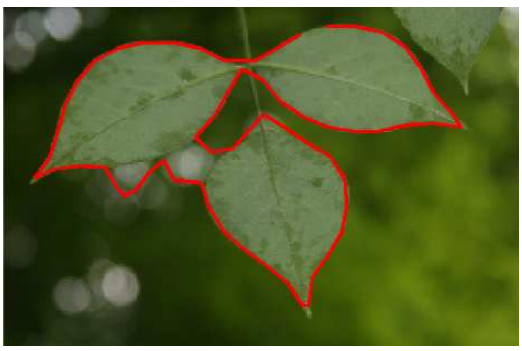


(a) Original Image with contour (b) Foreground and Background



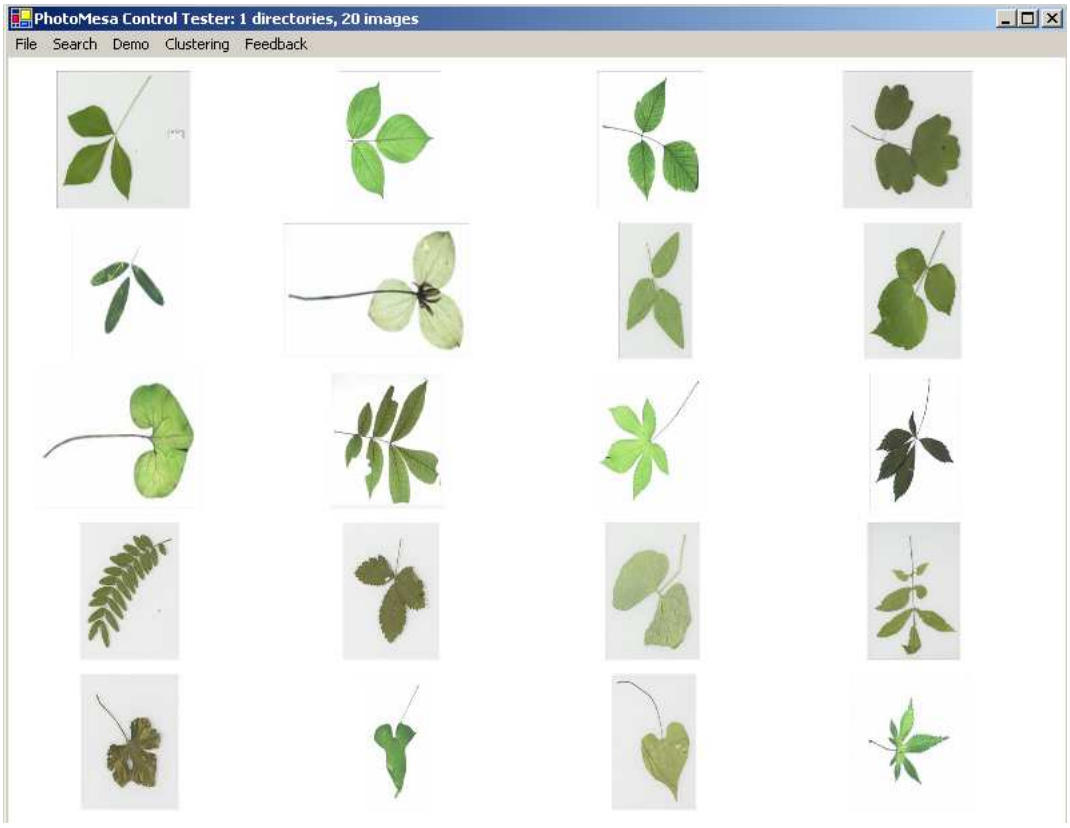
(c) Looks like the first image is the correct match

Figure 7.25: The search results using EFG for images with no controlled conditions. Looks like the first image is the correct match



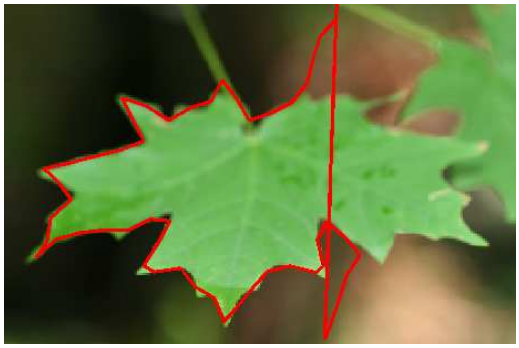
(a) Original Image with contour

(b) Foreground and Background



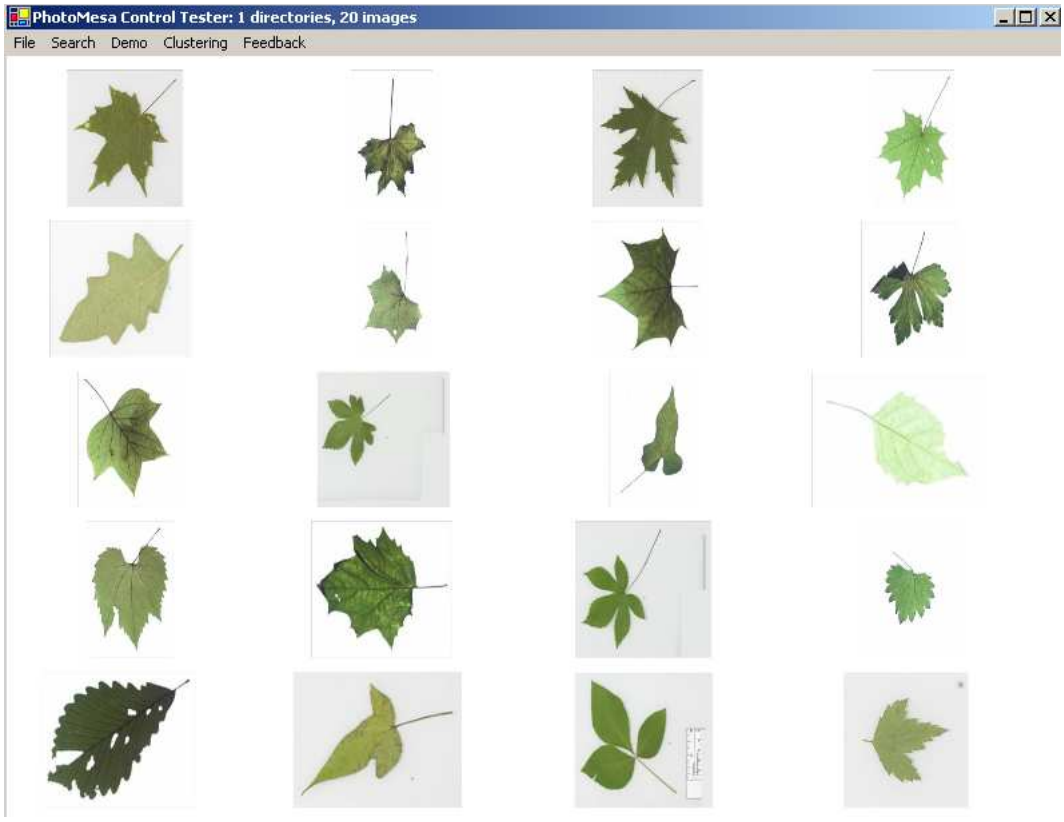
(c) The first image is the right match

Figure 7.26: The search results using EFG for images with no controlled conditions. The first image is the right match



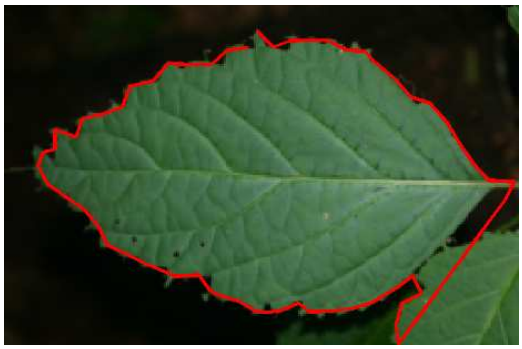
(a) Original Image with contour

(b) Foreground and Background



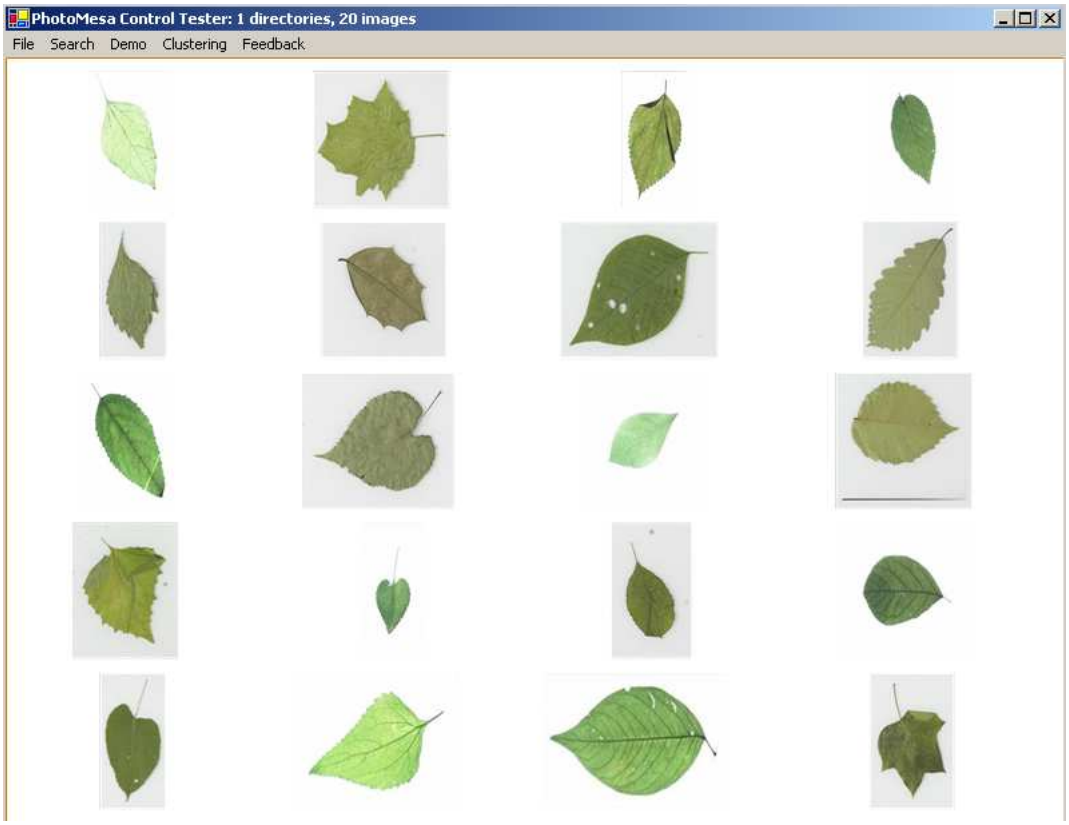
(c) Similar shaped leaves are retrieved

Figure 7.27: The search results using EFG for images with no controlled conditions. Similar shaped leaves are retrieved



(a) Original Image with contour

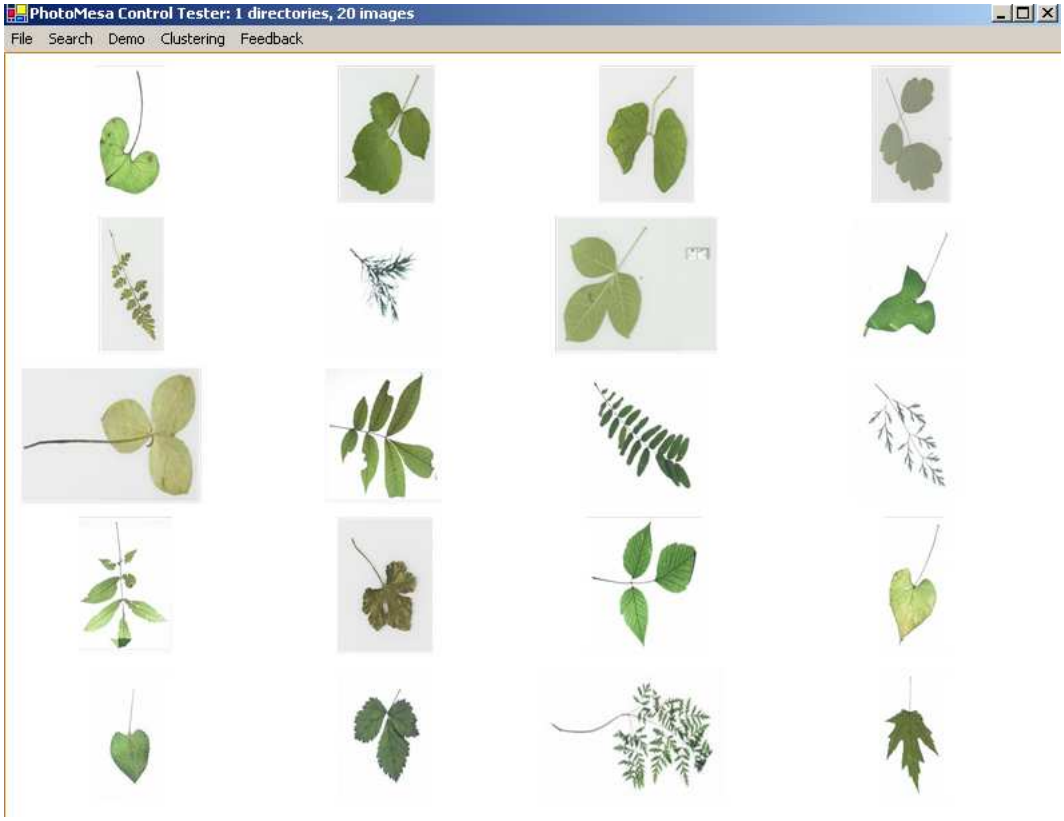
(b) Foreground and Background



(c) Similar shaped leaves are retrieved

Figure 7.28: The search results using EFG for images with no controlled conditions. Similar shaped leaves are retrieved

(a) Original Image with noisy Contour. (b) Foreground and Background

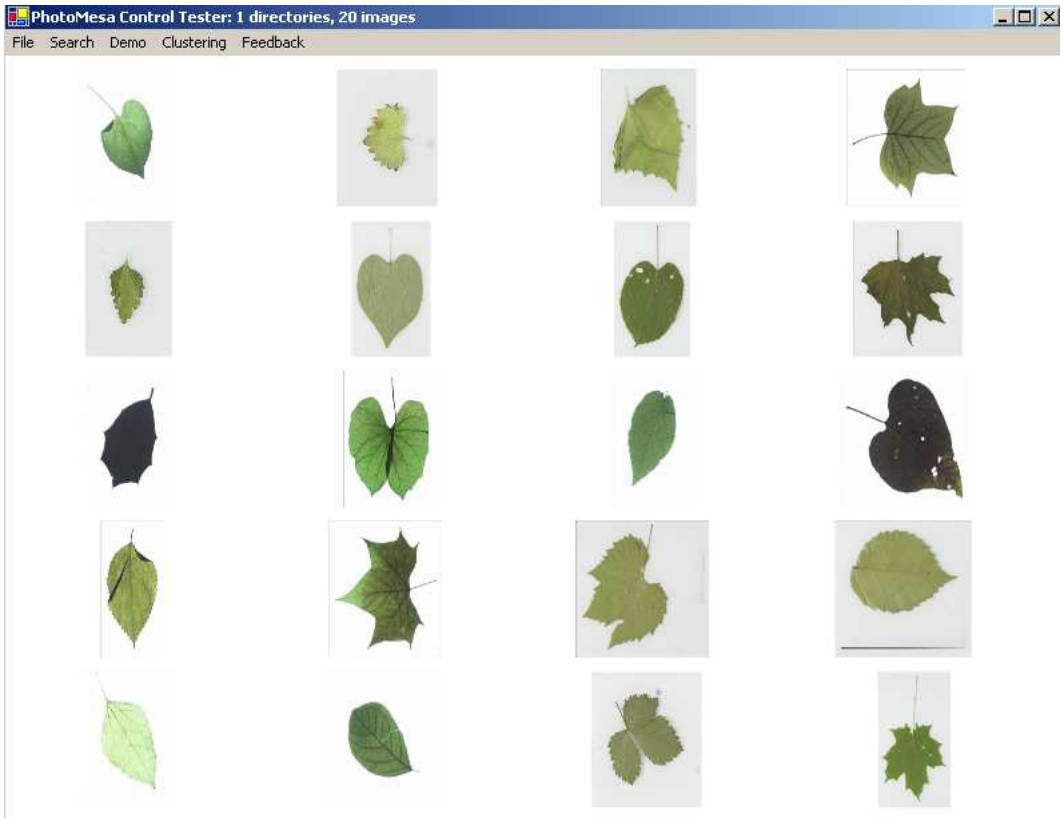


(c) Similar shaped leaves are retrieved.

Figure 7.29: The search results using EFG for images with no controlled conditions. Similar shaped leaves are retrieved

(a) Noisy Contour.

(b) Foreground and Background



(c) The results are not very good as the contour of the compound leaf is not clear

Figure 7.30: The search results using EFG for images with no controlled conditions. The results are not very good as the contour of the compound leaf is not clear

Chapter 8

Conclusion and Future Work

In this work, we suggest better ways to find saliency maps that indicate important regions in an image. We have developed a CBIR system and provided empirical evidence to some recently suggested methods of image browsing and navigation. We also conducted a field test to check the robustness of the system in varying photography conditions.

8.1 Saliency and Thumbnails

Given a set of images, their thumbnails can be shown to the user to give an overview of the contents. Usually thumbnails are downsampled versions of the original images. Cropping and downsampling has been found to produce better thumbnails [13] (where Itti's algorithm [1] have been used to find salient regions). We have developed computationally efficient methods to generate the saliency maps. We found that variance and wavelets based algorithms gives good saliency maps (as compared to Itti's algorithm) and are much simpler.

Now, we have methods that indicate the regions of saliency. These saliency maps are usually noisy and highlight some non-important regions as salient. New approaches that give better saliency maps might be helpful. Once we have good saliency maps, it might be possible to predict if an image has a distinct foreground

and subsequently other characteristics of the same.

8.2 Navigation and Browsing of Image databases

We developed a Content Based Image Retrieval (CBIR) system to provide empirical evidence for some suggested methods for browsing and navigation of image databases. We conducted a user study to show that for image retrieval, grouped and hierarchical placement of images based on similarity is better than random placement. Using an effective shape based similarity measure [32], we are able to conclude that visual search is helpful in such systems. Shape contour is the distinguishing feature in the leaf database images (our dataset). For general image databases, better similarity measures might be needed. We also conducted a field test to test the robustness of the system in real field conditions. Results are encouraging though better background subtraction techniques are needed to get good contours.

BIBLIOGRAPHY

[1] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[2] Philippe H. Gosselin and Matthieu Cord. A comparison of active classification methods for content-based image retrieval. In *CVDB '04: Proceedings of the 1st international workshop on Computer vision meets databases*, pages 51–58, New York, NY, USA, 2004. ACM Press.

[3] N Vasconcelos and M Kunt. Content-based retrieval from image databases: Current solutions and future directions. In *Proceedings of International Conference in Image Processing (ICIP'01)*, pages 6–9, 2001.

[4] Zoran Pecenovic, Minh N. Do, Martin Vetterli, and Pearl Pu. Integrated browsing and searching of large image collections. In *VISUAL*, pages 279–289, 2000.

[5] J-Y. Chen, C. A. Bouman, and J. C. Dalton. Hierarchical browsing and search of large image databases. *IEEE Transaction on Image Processing*, 9(3):442–455, 2000.

[6] Hao Liu, Xing Xie, Xiaoou Tang, Zhi-Wei Li, and Wei-Ying Ma. Effective browsing of web image search results. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 84–90, New York, NY, USA, 2004. ACM Press.

[7] Tammara T. A. Combs and Benjamin B. Bederson. Does zooming improve image browsing? In *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, pages 130–137, New York, NY, USA, 1999. ACM Press.

[8] Jonathan Ashley, Myron Flickner, James Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. The query by image content (qbic) system. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, page 475, New York, NY, USA, 1995. ACM Press.

[9] John P. Eakins, Jago M. Boardman, and Margaret E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998.

[10] Simone Santini and Ramesh Jain. Integrated browsing and querying for image databases. *IEEE MultiMedia*, 7(3):26–39, 2000.

[11] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. *Proceedings of the IEEE International Conference on Computer Vision*, pages 59–66, 1998.

[12] J. B. Kruskal. Multi-dimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika,*, 29:1–27, 1964.

[13] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 95–104, New York, NY, USA, 2003. ACM Press.

[14] Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. Picsom-content-based image retrieval with self-organizing maps. *Pattern Recogn. Lett.*, 21(13-14):1199–1207, 2000.

[15] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kenneth Wood. Does organisation by similarity assist image browsing? In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190–197, New York, NY, USA, 2001. ACM Press.

[16] Sara L. Su. Perceptual picture emphasis using texture power maps. Master's thesis, Massachusetts Institute of Technology, January 2005.

[17] Carsten Rother, Sanjiv Kumar, Vladimir Kolmogorov, and Andrew Blake. Digital tapestry. In *CVPR'05: IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[18] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, HongJiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia Syst.*, 9(4):353–364, 2003.

[19] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.

[20] E. Niebur and C. Koch. *The attentive brain*, chapter 9, page 163186. The MIT Press, Cambridge,MA, 1998.

[21] Ernst Niebur and Christof Koch. Control of selective visual attention: Modeling the where pathway. In *NIPS*, pages 802–808, 1995.

[22] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, May 2000.

[23] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1):107–23, Jan 2002.

[24] D. Parkhurst and E. Niebur. Scene content selected by active vision. *Spatial Vision*, 16(2):125–54, 2003.

[25] P. Reinagel and A. Zador. Natural scene statistics at the center of gaze. *Network: Computation in Neural Systems*, 10:1–10, 1999.

[26] G. Krieger, I. Rentschler, G. Hauske, K. Schill, and C. Zetzsche. Object and scene analysis by saccadic eye-movements: An investigation with higher-order statistics. *Spatial Vision*, 13(2-3):201–214, 2000.

[27] S. Mannan, K.H. Ruddock, and D.S. Wooding. The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images. *Spatial Vision*, 10(3):165–188, 1996.

[28] S. Mannan, K.H. Ruddock, and D.S. Wooding. Automatic control of saccadic eye movements made in visual inspection of briefly presented 2-d images. *Spatial Vision*, 9(3):363–386, 1995.

[29] Gaurav Agarwal, Alwin Anbu, and Aniruddha Sinha. A fast algorithm to find the region-of-interest in the compressed mpeg domain. In *ICME '03: Proceedings of the International Conference on Multimedia and Expo*, pages 133–6 vol 2, 2003.

[30] A. Sinha, G. Agarwal, and A. Anbu. Region-of-interest based compressed domain video transcoding scheme. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, 3:162–4, 2004.

[31] Benjamin B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80, New York, NY, USA, 2001. ACM Press.

[32] H. Ling and D.W. Jacobs. Using the inner-distance for classification of articulated shapes. In *CVPR'05: IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.