The Computer Journal Advance Access published April 9, 2010

doi:10.1093/comjnl/bxq037

Plant Image Retrieval Using Color, Shape and Texture Features

HANIFE KEBAPCI, BERRIN YANIKOGLU* AND GOZDE UNAL

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul 34956, Turkey
**Corresponding author: berrin@sabanciuniv.edu*

We present a content-based image retrieval system for plant image retrieval, intended especially for the house plant identification problem. A plant image consists of a collection of overlapping leaves and possibly flowers, which makes the problem challenging. We studied the suitability of various well-known color, shape and texture features for this problem, as well as introducing some new texture matching techniques and shape features. Feature extraction is applied after segmenting the plant region from the background using the max-flow min-cut technique. Results on a database of 380 plant images belonging to 78 different types of plants show promise of the proposed new techniques and the overall system: in 55% of the queries, the correct plant image is retrieved among the top-15 results. Furthermore, the accuracy goes up to 73% when a 132-image subset of well-segmented plant images are considered.

Keywords: image retrieval; plants; Gabor wavelets; SIFT

Received 4 October 2009; revised 9 March 2010
Handling editor: Hakki Toroslu

1. INTRODUCTION

Content-based image retrieval (CBIR) offers efficient search and retrieval of images based on their content. With the abundance and increasing number of images in digital libraries and the Internet in the last decades, CBIR has become an active research area. Two important query categories can be distinguished: (i) query by example (i.e. find examples similar to this one) and (ii) semantic retrieval using a description of the search concept (e.g. find images containing bicycles). Query by example is often executed by comparing images with respect to low-level features obtained from the whole image, such as color, texture or shape features. Semantic retrieval on the other hand requires higher-level understanding of the image contents. In this case, local features such as scale-invariant feature transform (SIFT) descriptors ([1, 2]) are used for this problem, in locating objects within complex scenes. These two broad categories can be further subdivided. For instance the query by example can be done by providing a sketch or a template instead of an image. Similarly, the semantic retrieval can be made at different levels of abstraction of the query concept (e.g. by providing an example image or a description of a bicycle) [3].

Examining images based on color is one of the most widely used techniques. The earliest image retrieval studies used color as the comparing feature between images [4–6]. A simple color similarity between two images can be measured by comparing their color histograms. The color histogram, which is a common color descriptor, indicates the occurrence frequencies of colors in the image [7]. More complex color features may involve looking into the spatial relationship of multiple colors or looking at the color histograms in automatically segmented regions of the image. For example, extending the idea of a histogram, a color co-occurrence matrix gives information about the color distribution in neighboring image pixels.

Other widely used features can be grouped as shape and texture features. Shape is one of the most important characteristics of an object. In CBIR, the problem often refers to finding a particular shape within the queried images, which requires image segmentation. Image segmentation is a difficult problem by itself; in fact, segmentation and recognition often form a chicken-or-egg problem. Once segmented, the shape of an object may be described and matched using a variety of shape features.

Two basic approaches to shape analysis exist: region-based and boundary-based (contour-based) [8]. Region-based systems typically use moment descriptors [9] that include geometrical moments, Zernike moments and Legendre moments [8]. Boundary-based systems use the contour of the objects and usually give better results for images that are distinguishable by their contours. Fourier descriptors [8, 9], curvature scale

space methods and deformable templates [10] are some of the common techniques used in contour-based shape recognition.

Texture can be described as the spatial patterns formed by the surface characteristics of an object that manifests itself as color or grayscale variations in the image. While each surface has a texture, some objects can be said to have distinguishing textures (e.g. skin or sand). Texture analysis and matching can be done in the spatial or the frequency domain. Commonly used texture features are gray-level co-occurrence matrices, local binary patterns (LBP), Markov random fields and Gabor wavelets [11–15].

In this paper, we present a plant retrieval system which takes as input the image of a plant and returns the most similar images from a database. The system is intended to be used as a web service where users can send images of their house plants (which they often do not know by name) to find their Latin names and care instructions. The problem involves identification of the matching plant, as well as retrieval of related varieties which may also be of interest to the user.

We assume that a close snapshot of the plant is given as query, ideally showing the outer boundary of the plant. However, this assumption does not always hold even for our own database, as some of the pictures are close snapshots not showing the outer boundary of the plant. Sample plant images from our database are shown in Fig. 1 to illustrate the complexity of the problem, including illumination, pose and scale variations, as well as differences in shape, background and flower colors.

The paper is organized as follows. In Section 2, related work on CBIR is presented. In Section 3, the proposed system is described, followed by the description of the collected database and results, in Sections 4 and 5. Finally, a summary of the contributions and future work are discussed in Sections 6 and 7.

2. RELATED WORK

Research on CBIR has shown its first significant results with feature-based systems in the early 1990s [16–18]. Later research moved more toward semantic analysis of content [3, 19], as well as improving the usability where, for instance, relevance feedback was used to improve the results. Some of the latest research in this field can be found in [3, 5, 19, 20].

While there are some plant images in the commonly used image retrieval databases (e.g. the Corel database), we are not aware of a CBIR system geared specifically toward house plant retrieval. However, there are some related work in the areas of plant classification and identification that are developed for botanical or agricultural needs. In systems geared toward botanical applications [21–30], segmented leaf images are used to identify unknown plant varieties, using features obtained from the leaf contour. Yahiaoui *et al.* proposed an image retrieval system for identifying plant genes by using contour-based shape features in [21]. The extracted shape descriptors in this study include the length histogram of contour segments in different directions [22, 23]. Another work on plant image retrieval focused on the leaf image retrieval problem using features such as centroid-contour distance curve, eccentricity and angle code histograms. These features are extracted from the leaf edge contour after some preprocessing (e.g. scale normalization). Bruno *et al.* proposed a shape classification method [24, 25] which they applied to the tree leaf identification problem. The shape features used in these studies (fractal dimension), are extracted using either segmented leaf outline or leaf venation system to retrieve both internal and external leaf characteristics. In some recent work [26, 27], the retrieval algorithm is supported with machine-learning techniques. In [26], plant leaves are classified based on their texture features, using LBP and Gabor wavelets together. The extracted texture features

FIGURE 1. Example images of two plants showing different types of variations among images of the same plant. Above plant: Hyacinth, Below: Calathea.

(spatial histograms) are then fed to a support vector machine (SVM) classifier. The study in [27] combined color and texture features (color moments and wavelet transform) after rotating each leaf so as to align its central axis with the horizontal. An SVM classifier is trained with the extracted color and texture features in order to recognize the plants.

Systems geared toward agricultural applications include detecting weeds in the field [31], detecting position of specific plants [32] and deciding whether or not a plant is damaged by a specified illness [33]. In [31], color and shape information is used to detect weeds in the field. Sena's work [33] on identifying damaged maize plants proposes a segmentation step to be used first. Leaf segmentation is done by thresholding the monochrome images that are converted from RGB using a transformation called the normalized excess green index ($2g - r - b$, where g, r and b are intensity values on corresponding RGB color channels) to distinguish weeds from soil regions. In [32], the center of a maize plant in a field is located by intersecting the main vein lines of leaves, that are in turn detected by using reflectance difference of veins and leaves.

While they attack a very similar problem, the above-mentioned plant identification systems differ from ours in an important way: they all use a well-segmented image of the plant leaf, while we use an image of the whole plant which shows variations in illumination, pose and scale, as well as overall plant shape, background and flower colors, if any.

3. METHODOLOGY

Given a query image, the system first uses an interactive segmentation step to segment the plant from the background. While we assume that the query pictures will be close snapshots of the plant, the pictures cannot be assumed to be free of background (e.g. pot, table, background wall). The segmentation step ensures the extraction of relevant features, by focusing on the plant region rather than the background. In the current system, the segmentation is run semi-automatically: for the database images, the foreground and background regions are marked manually until a satisfactory segmentation result is obtained. In the actual use scenario, the user would have to mark a few regions to identify some background and foreground regions; but once those regions are selected, further operations including segmentation are performed by the system automatically without needing any additional information. Using segmented plant image rather than the original image brings an important gain in the system performance.

The segmented region is used for feature extraction which consists of well-known color, shape and texture features, along with some modifications to existing texture matching techniques and introducing some new shape features. We studied the suitability of these features, both individually and in combination, in order to find the best combination for the given problem. The feature extraction and matching steps are explained in the following subsections.

3.1. Image segmentation

We use the max-flow min-cut (MFMC) graph cut method [34] to segment the plant images from the background. The MFMC technique has recently become one of the most popular segmentation approaches in computer vision; it efficiently minimizes an energy defined on a graph constructed over the image, based on the image descriptors. In the basic graph-cut approach, an image is represented as a graph where the graph nodes are the pixels and the graph edges are formed between the neighboring pixels in the image. The algorithm requires seed foreground and background pixels (source and sink, respectively) to be specified. It then splits the graph into two disjoint sets S (source) and T (sink), minimizing a cost functional. The selected seeds form the initial values of the sets S and T. The assignment of the image pixels into two disjoint sets corresponds to a binary labeling of the image with foreground and background regions. The functional is based on two values: (i) a spatial smoothness term which measures the cost of assigning the same label (e.g. foreground or background) to adjacent pixels and (ii) an observed data term that measures the cost of assigning a label to each pixel.

Once an energy or cost functional is defined as described above, one can resort to efficient optimization methods that exist in the algorithms literature. For solving MFMC problem on directed weighted graphs, the *augmenting path* algorithm of Ford and Fulkerson [35], the *push-relabel* method and a modified version of the augmenting path method by Boykov and Kolmogorov [34] were introduced. In our work, Boykov and Kolmogorov's technique is utilized since it is efficient and widely used. There is no parameter expected from the user for the segmentation; both the query and the database images are segmented using the same settings. The most important variable for the graph-cut algorithm is the variance estimator (σ^2) that is used to calculate the capacity between adjacent pixels (of edges) in order to maximize the flow. We used the default value for this variable that is determined according to the image intensity values. Specifically, σ^2 is set according to the variance of the foreground seed region points and the maximum intensity of the image.

Currently, the seed and background selection is carried out manually, using a MATLAB GUI program that we have implemented. For selecting the seed points more efficiently, the user selects seed *regions* by indicating the vertices of polygonal regions. Defining five polygonal seed regions requires the selection of 15 points, on average. Figure 2 shows sample segmentation results on two plant images from our database. While the segmentation step of the current system is relatively easy, we plan on adopting a more user friendly interactive segmentation technique, such as the GrabCut algorithm [36], in the future.

FIGURE 2. Segmentation examples from our database: the input image, the seed points shown as white (background) and red (foreground) regions and the segmented images are shown in a row.

3.2. Feature extraction

In the developed system, images are analyzed using various color, texture and shape features. The use of color in plant retrieval is more complicated compared with most other CBIR applications, since most plants have green tones as their main color. Furthermore, the color of the flowers also poses a challenge: two flowering plants should be matched despite differences in flower colors. For instance, given an hyacinth of a certain color, ideally one should find its exact match from the database, as well as other hyacinth plants with different flower colors like the ones in Fig. 1. We currently use some basic color features consisting of color histograms and color co-occurrence matrices obtained from the segmented image, to represent the color information.

Probably the most important aspect of an object is its shape, and the same applies to plants as well. In the plant identification problem, both the leaf shape and the overall shape of the plant are important. We use the SIFT features to extract the local shape features of the plant and some newly proposed features extracted from the plant's outer contour, to describe the overall plant shape.

In addition to color and shape, the third core characteristic of an object is its texture. The texture of a plant, formed by the color and vein patterns, is also important in plant identification. In our problem, texture features are extracted using Gabor wavelets. In the remainder of this section, we give details of the proposed feature extraction process for extracting color, shape and texture features.

3.2.1. Color features

As in many other studies [6, 37–39], we used color histograms and color co-occurrence matrices to assess the similarity between two images. If the overall color or color pair distributions of two images are close, they are matched as similar in terms of their colors.

Three different color spaces are used to produce color histograms; namely RGB, normalized RGB (nRGB) and HSI color spaces. In the RGB color space, each color is represented as a combination of the three primary color channels (Red, Green and Blue). While commonly used, the RGB color space has an important shortcoming which is the sensitivity to illumination changes. In fact, different color spaces may be suitable in different applications. For instance, the nRGB and the HSI color spaces are often used in order to obtain robustness against illumination differences. The nRGB color model is a derivation of the RGB model in which each channel value is normalized with the total intensity of all channels. The normalization process effectively normalizes for different illumination conditions. The colors are represented by three normalized color values (nR, nG, nB), which indicate the red, green and blue color ratio in a specific pixel. The normalization computation for red and green channels is formulated as follows: $\mathrm{nR} = R/(R+G+B)$ and $\mathrm{nG} = G/(R+G+B)$. In the HSI color model, color is represented using its Hue, Saturation and Intensity values. The important feature of this color space is the separation of the intensity from the chromaticity.

In order to obtain a histogram robust to normal variations in plant images, the 24-bit RGB information is quantized into a 9-bit representation (for a total of 512 bins, using 3 bits for each color channel). For the nRGB representation, one of the channels can be deduced from the normalized value of the other two ($\mathrm{nR} + \mathrm{nG} + \mathrm{nB} = 1$); therefore, we compute the nRGB color histogram using only the values of two normalized channels, which affords more bins (for a total of 256 bins, using 4 bit for each of the nR and nG values). In the HSI space, the

360 different hue values are quantized to 10, 30 or 90 bins. The intensity value is intentionally discarded, while the saturation component is unused in the current work, for simplicity. Prior to histogram matching, we smooth the computed histograms by taking weighted averages of the consecutive bin values, so as to obtain some robustness against quantization problems.

Although color co-occurrence is generally mentioned as a texture analysis method, it primarily indicates the distribution of color pairs. We use a 30×30 co-occurrence matrix computed from the HSI color space, where $C[i][j]$ stores the number of neighboring image pixels having the hue values i and j. We generate the co-occurrence matrix using three different methods: (i) considering only four neighboring pixels (i.e. top, bottom, right and left neighbors); (ii) considering all eight neighboring pixels; and (iii) using 8-neighbors but ignoring the diagonal elements of the co-occurrence matrix. Diagonal elements store the number of neighboring pixels that have the same quantized color and dominate the matching process since they correspond to large uniform color regions in the image. Thus, this last method aims to concentrate on areas where colors change, rather than the uniform areas.

3.2.2. *Shape features*

We use two types of shape features in this work: (i) SIFT features to capture local characteristics of the plant and (ii) newly proposed global shape features describing the overall plant shape.

SIFT features have been successfully used in many recognition and retrieval problems [1, 40–42]. In this approach, scale-invariant keypoints (or interest points) in an image are located and the descriptors measured at these keypoints are compared with those obtained in another image. A high number of matching keypoints indicate similarity between two images. The algorithm can be summarized as follows. Candidate keypoints on the image are first detected by checking the scale space extrema obtained from the difference of Gaussian images. After eliminating some of these candidate keypoints (such as those aligned along the edges), the dominant orientation is selected for each keypoint by computing the weighted average of the gradient magnitudes of 4×4 neighboring pixels. Then, histograms of gradient orientations are calculated at each keypoint, relative to the dominant orientation, to provide rotational invariance. The resulting 128 histogram values consisting of 8-bin histograms, for a total of 4×4 neighboring pixels, constitute the SIFT descriptors. The number of matching keypoints, as measured by the similarity of their descriptors, is used as a measure of similarity of two images. A keypoint obtained in an image is generally matched to many keypoints in the second image; to eliminate the spurious matches, the algorithm checks to see if the match is significant. The significance is measured by considering the ratio of the match scores of the first and second best matching keypoints. If this ratio is more than the prefixed ratio threshold, the keypoint is discarded.

The SIFT features of segmented plant images converted to grayscale are extracted by the public domain SIFT algorithm of David Lowe [2]. The average number of matching keypoints between two images is around 20 in our problem, obtained using a distance ratio parameter of 0.7. Two plant images and the matched keypoints are shown in Fig. 3.

The SIFT algorithm is not as successful in the plant retrieval problem as it is in some other domains. For this reason and also because we consider the outline of a plant image as an important characteristic, we propose some new features extracted from the overall plant contour. These global shape features are evaluated separately and also in addition to the SIFT features.

In order to extract the global shape features, first the contour of the plant is extracted by tracing the segmented plant image. The contour is then represented as a *chain code*, which is a common representation of contour as a series of enumerated direction codes ([7, 43]). Next, high curvature points on the contour are extracted by analyzing direction changes in the chain code. These points are labeled as convex or concave depending on their position and direction (or curvature of the contour). Figure 4 shows the contour for a given segmented plant image, along with the high-curvature points, labeled according to their type. Finally, the following six features are extracted, using the extracted contour and the labeled high-curvature points of the contour:

(i) Number of Concave Points (e.g. A, C, E)
(ii) Number of Convex Points (e.g. B, D)
(iii) Leaf Base Distance ($\overline{\mathrm{AC}}$)
(iv) Leaf Tip Distance ($\overline{\mathrm{BD}}$)
(v) Leaf Arc Length ($\widehat{\mathrm{ABC}}$)
(vi) Normalized Leaf Arc Length ($n\widehat{\mathrm{ABC}} = \widehat{\mathrm{ABC}}/\overline{\mathrm{AC}}$)

The leaf arc length is measured as the arc distance between two consecutive concave points ($\widehat{\mathrm{ABC}}$), while the leaf base distance ($\overline{\mathrm{AC}}$) is measured as the straight line distance between

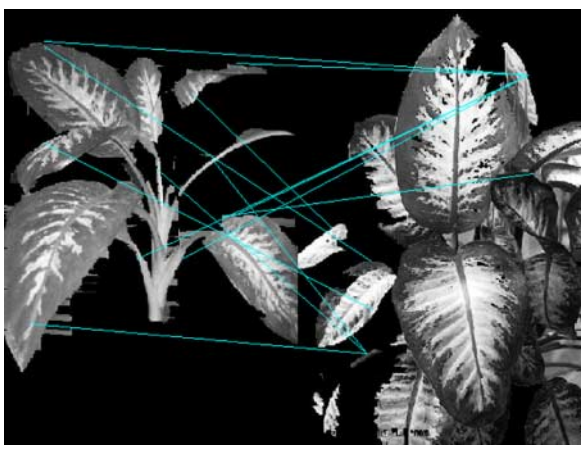

FIGURE 3. Matching SIFT keypoints between two plant images. The lines connect the matching keypoints.

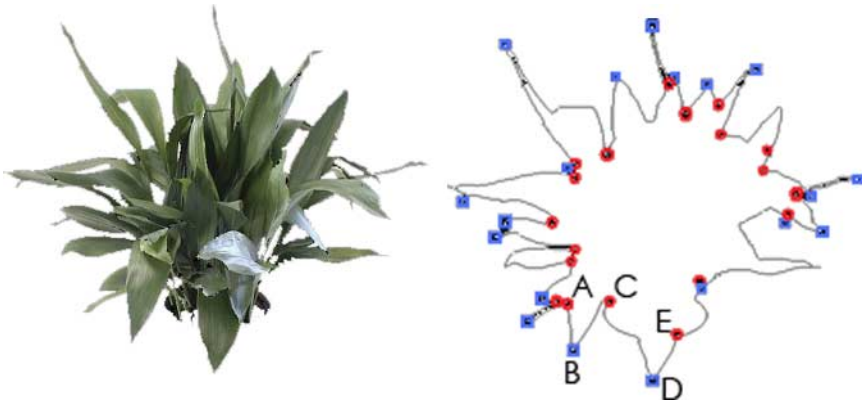

FIGURE 4. Traced plant contour of a plant, used to extract global shape characteristics. The squares and circles mark the convex and concave points on the contour.

them. Similarly, the leaf tip distance is measured as the distance between two consecutive convex points ($\overline{\text{BD}}$).

For robustness, we obtain the median value of these features over the whole plant region. Also, in order to obtain scale invariance in shape features, some features are normalized with the image size (leaf arc length, leaf base distance, leaf tip distance) or the contour length (number of high-curvature points), as appropriate.

3.2.3. *Texture features*

Texture of a plant may be due to having many veins in different directions or parallel lines of different colors. In addition to the texture of a single leaf, the whole plant also has a texture of its own that depends on the frequency of leaves, their orientation and curvature. We use Gabor wavelets to extract the plant texture. Gabor wavelets, which are commonly used for texture analysis [11, 14, 15], are obtained through the multiplication of a Gaussian function and a harmonic function. The basic 2D Gabor function can be stated as follows:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-1/2(x^2/\sigma_x^2 + y^2/\sigma_y^2)} \times e^{2\pi i f x}.$$

The first term in this equation is the normalized 2D Gaussian function with different standard deviations in two dimensions (σ_x and σ_y) and the remaining term is the complex sinusoid with frequency f, which can be expanded using the Euler's formula as follows:

$$e^{2\pi i f x} = \cos 2\pi f x + i \sin 2\pi f x.$$

Gabor filters in different orientations and scales are used to detect textures in different orientations and scales, respectively. In our case, we use the following family of Gabor functions generated at different orientations (u) and scales (s):

$$g_{su}(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x'^2 + y'^2)/\sigma^2} \times e^{2\pi i f x'/C},$$

where

$$x' = x\cos(\theta) + y\sin(\theta),$$
$$y' = -x\sin(\theta) + y\cos(\theta).$$

Here x and y indicate the coordinates in the image, while x' and y' indicate the coordinates rotated by θ; f is the spatial frequency of the sinusoid and C is the window size in x-dimension; s denotes the scale variable that determines the frequency as $f = 2s + 1$ and u indicates the chosen orientation that determines the rotation angle as $\theta = u\pi/4$. The orientation variable u ranges between 1 and K, where K is the number of orientations (typically 4) and the scale variable s ranges between 1 and S (4 in this work). Here, as a special case of the general formula, we use $\sigma = \sigma_x = \sigma_y$ and set it according to f.

The response of a Gabor filter on an image $I(x, y)$ is the convolution of the image ($I(x, y)$) and the Gabor filter. The convolution is expressed below:

$$R_{su}(x, y) = \sum_{m=-C/2}^{C/2} \sum_{n=-C/2}^{C/2} I(x - m, y - n) g_{su}(m, n),$$

where $g_{su}(m, n)$ denotes the Gabor function at scale s and orientation u; and m and n are variables that are use to sum the response over the Gabor filter window's size C. The energy response of a Gabor filter in a certain scale and orientation indicates the concurrence of the filter with the image texture.

Wavelets in four different orientations and four different scales are used in this work. With four scales (f_{1-4}) and four orientations (u_{1-4}), a total of 16 Gabor wavelets are applied to each image, resulting in 16 different Gabor response images. We use the mean (μ_i) and standard deviation (σ_i) of these maps in comparing the texture differences between two response images.

When comparing the texture similarity of two images, often the comparison is done using the Gabor responses in all scales.

We call this the *default* texture matching method. An alternative is to do the comparison only in the most dominant scale for each image, with the intention of ignoring scale differences across images of the same plant. We call this the *max-scale* texture matching method. Finally, we introduced a new, third approach which is called *patch-based*, to provide rotational invariance on a leaf level and scale invariance, as explained below.

Patch-based texture extraction When a uniformly textured object (e.g. straw or fabric) is rotated, its Gabor response maps in different orientations on the same scale are circularly shifted. For instance, when an object with a dominant texture along the X-axis (0°) is rotated 45°, the response of the rotated image is dominant on the 45° Gabor response. Hence, if we represent the feature vector starting with the angle with the maximum response and in increasing angular order, we can match the corresponding maps to obtain rotation invariance. In this example, the initial texture feature vector

$$\{(\mu_0, \sigma_0), (\mu_{45}, \sigma_{45}), (\mu_{90}, \sigma_{90}), (\mu_{135}, \sigma_{135})\}$$

would be matched to the circularly shifted feature vector of the rotated image

$$\{(\mu_{45}, \sigma_{45}), (\mu_{90}, \sigma_{90}), (\mu_{135}, \sigma_{135}), (\mu_0, \sigma_0)\}.$$

The rotation invariance problem is more complex in plant images than in general CBIR problems. Even if the texture in a plant shows little variations across the leaves of the plant, the fact that the leaves are often oriented in different directions makes the above rotation invariant method inapplicable (see Fig. 5 for an example). For this, the ultimate solution is to go down to the leaf level and compare the texture responses of individual leaves. In this work, we attempt to approximate this by using a patch-based approach where we divide the image into uniformly distributed patches and rotate the texture responses of individual patches to a canonical order (increasing angular order starting with the most dominant response), constructing the overall response images R_{su} from these rotated responses.

Feature extraction for the patch-based method is performed as follows. First, the Gabor response images calculated in $K = 4$ directions is divided into 20×20 patches and mean intensity and standard deviation values (μ, σ) are calculated for each of these patches. Then, for each patch, the Gabor texture features $\{(\mu_k$ and $\sigma_k)\}$ are circularly shifted to a canonical order, so as to have the maximum response in the first element. At this point, Gabor responses in $K = 4$ directions can be reconstructed from the responses of individual patches, as if the patches were rotated in the original image; however, it is sufficient to calculate the Gabor features (mean and standard deviation) for the whole image, from the responses of individual patches. The patch-based texture analysis aims to normalize the rotation of individual leaves so that their Gabor responses are in the same direction; however, since we do not have leaf level segmentation, patches are used to approximate the leaf. While this method is not guaranteed to provide full rotation invariance, the experimental results show that it does help with the texture analysis.

We implemented the patch-based method considering Gabor response images in one scale only since the patch-based method provides invariance to scale variations. As each image is divided into a constant number of patches, a patch will contain a certain proportion of the plant area, regardless of the image resolution. Thus, there are 4×2 patch-based texture features with $K = 4$ and $S = 1$.

3.3. Matching

The dissimilarity between a query image Q and a database image I is assessed according to the extracted feature(s). The metrics used in matching different features are explained in this section.

3.3.1. *Color matching*

The RGB color dissimilarity score of two images Q and I is calculated using the Kullback–Leibler divergence (*KL-divergence*) measure of the corresponding histograms h_Q and h_I:

$$\delta_{\mathrm{RGB}}(Q, I) = -\sum_{i=1}^{512} h_Q(i) \log h_I(i) + \sum_{i=1}^{512} h_Q(i) \log h_Q(i),$$

where $h_Q(i)$ and $h_I(i)$ are the values of ith bin of Q's and I's histograms, respectively. The KL-divergence of two histograms can be expressed using the concept of entropy; specifically how many bits are needed to represent the histogram of I by using the histogram of Q as the reference:

$$\delta_{\mathrm{RGB}}(Q, I) = H(h_Q, h_I) - H(h_Q).$$

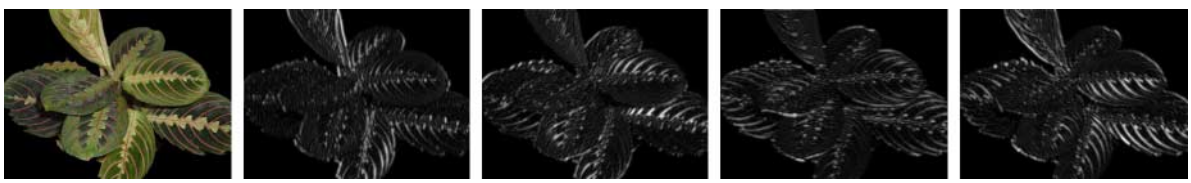

FIGURE 5. Four different maps show the texture energy in different orientations (0, 45, 90, 135 from vertical, from left to right). Note that the texture in different leaves of this plant are captured in different orientations.

Here $H(h_Q, h_I)$ is called cross entropy of h_Q and h_I, while $H(h_Q)$ is the entropy of Q's histogram.

The other color features, namely nRGB and hue histograms and the HSI-based color co-occurrence matrix are matched similarly. In those cases, the summation is done over 256, 10/30/90 and 900 (30×30) elements, respectively. Prior to the matching of the hue histograms, we smooth the histograms so as to obtain some robustness against quantization problems. This is done by taking weighted averages of the consecutive bin values and brings a very small improvement to the performance. For the color co-occurrence matrix, which is basically a 2-dimensional histogram, we again used the KL-divergence, with or without considering the diagonal elements, as explained in Section 3.2.1.

3.3.2. *Shape matching*

When using SIFT features, the similarity of two images is measured by the number of matching SIFT keypoints [2]. We use the following normalized SIFT dissimilarity score, for two images Q and I:

$$\delta_{\text{SIFT}}(Q, I) = 1 - \alpha(\log_{10}(m + 1)),$$

where m is the number of matching SIFT keypoints, α is a normalization constant with a current value of 0.25 and a logarithmic scale is used since the number of matching points range between 0 and possibly several hundreds. Note that using the unnormalized number of matching SIFT points is sufficient for retrieval, if SIFT features are used alone. Normalization is necessary when combining the SIFT dissimilarity with the dissimilarity scores obtained from the other features.

Contour-based shape dissimilarity is measured using the L_1-distance. For instance, the number of high-curvature (convex and concave) points is matched as follows:

$$\delta_{\text{HighCurv.}}(Q, I) = \left|\text{conc}_Q - \text{conc}_I\right| + \left|\text{conv}_Q - \text{conv}_I\right|.$$

When using multiple shape features, the shape dissimilarity of two images, Q and I, is calculated as the average of the individual shape feature dissimilarities.

3.3.3. *Texture matching*

The *default* texture dissimilarity of two images is calculated by taking the dissimilarity of each of the $s \times u$ Gabor response features as displayed in the following two equations. Here s denotes the scale, u denotes the orientation and $\delta_{su}(Q, I)$ denotes the dissimilarity in the given scale and orientation (s, u):

$$\delta_{\text{Gabor}}(Q, I) = \sum_{s=1,\ldots,4} \sum_{u=1,\ldots,4} \delta_{su}(Q, I), \tag{1}$$

where

$$\delta^2_{su}(Q, I) = w_1(\mu_{su}(Q) - \mu_{su}(I))^2 + w_2(\sigma_{su}(Q) - \sigma_{su}(I))^2.$$

Here w_1 and w_2 are weights of mean and standard deviation distance squares. We tested a few different values for w_i, giving more weight to the distance of the means (e.g. $w_1 = 0.8$, $w_2 = 0.2$); however, the overall results did not improve. As a result, we use the weights as $w_1 = 0.5$, $w_2 = 0.5$.

We also compare texture features of two images using two other approaches. One is called the *max-scale* method, where the texture of two images is matched in the dominant scale, as follows:

$$\delta_{\text{maxGabor}}(Q, I) = \max_{s=1,\ldots,4} \sum_{u=1,\ldots,4} \delta_{su}(Q, I).$$

In the newly proposed *patch-based* method, we obtain Gabor response images by rotating patches of the image (see Section 3.2.3). While the texture feature computation using the patch-based method is different, the dissimilarity score is computed similar to the default method. The only difference is that patch-based textures are measured in only one scale; hence, Equation (1) is computed with $s = 1$.

In addition to these similarity measures involving single features, we also experimented with combined methods, as done in various other studies [44, 45]. In this case, the overall dissimilarity score is the average of the individual scores. The outcome of various feature combinations are presented in Section 5.

4. DATABASE

Currently, we have 380 plant images from 78 different plant types in our database, collected mainly from the Web, but also by taking pictures of available house plants. The number of images for each plant type varies from type to type, ranging from 3 to 14 images, while the average number of images per plant is about 5. All the original images in the database are semi-automatically segmented to remove the background. The created house plant image database is publicly available.[1] The data collection is ongoing, with the aim of extending the variety to a minimum of 100 different plants as part of future work.

In the database the plant images are named following a standard notation: a number prefix indicating the plant type, its Latin name, a sample number and a postfix of "segm" to indicate the images that are segmented. A general gallery view of our implemented system is depicted in Fig. 6, displaying some of the images in the database after a particular query.

Some of the images in this database are not very suitable for shape analysis: close-up shots of the plant that do not show the outer contour and images with highly textured backgrounds that are badly segmented, resulting in a jagged contour. For that reason, we identified a *clean* subset of the database, consisting of 132 images of 32 different plants, in which each plant type has at least three images with smooth outer contours. This subset is used in evaluating shape related results.

Besides the plant image database collection, we use the MySQL database to index the images. The information of each

[1] *http://people.sabanciuniv.edu/berrin/SUPlants/*

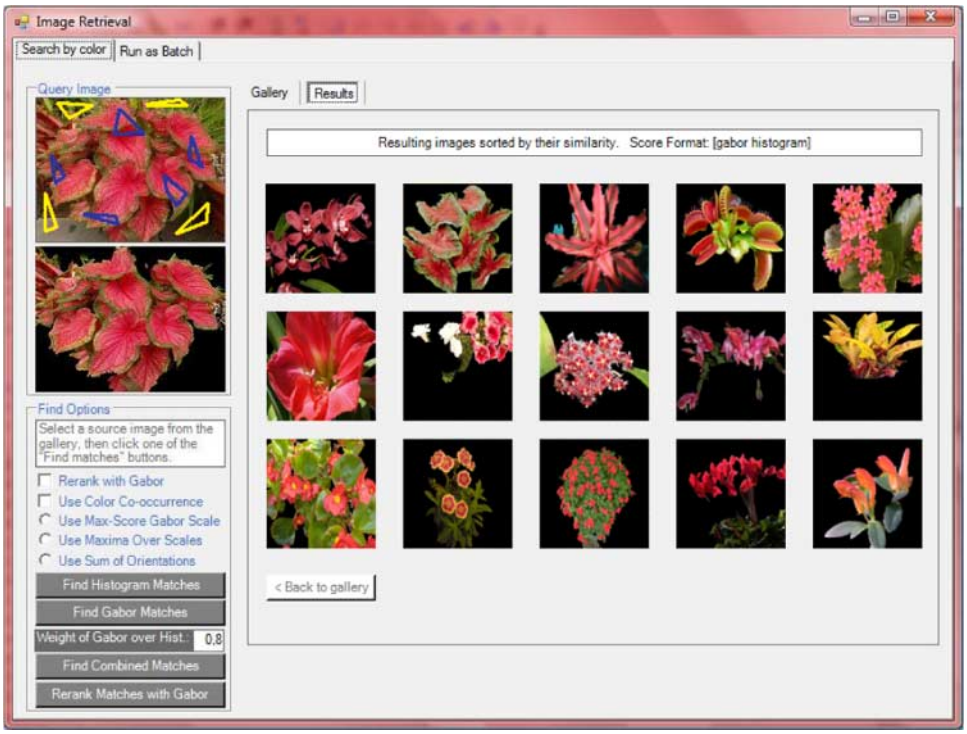

FIGURE 6. The GUI of the implemented system. The image in the top left corner is the original query, overlaid with blue and yellow regions indicating seed region selections. The lower left image is the segmented query image used for feature extraction. The images on the right are the top-15 similar plant images among the whole plant image database (ordered left-to-right and then top-to-bottom).

plant is stored in a table including filename, width, height, RGB histogram with 512 bins, Hue histogram with 90 bins, color co-occurrence matrix of 30×30 bins, 32 Gabor features consisting of 16μ and 16σ values and 6×3 global shape features (in three scales). Since the SIFT computation is time consuming and produces a variable number of keypoints, for each image in the database, we store the precomputed SIFT similarity scores to each of the $n - 1$ images, where n is the number of images in the database.

During indexing, the features of each image are extracted and inserted into the database. When a query image is uploaded to the system, the features extracted from the query image are compared with the stored feature values of the database images.

5. EXPERIMENTAL RESULTS

The performance of the system is evaluated by running tests over our plant image database. Each test is done as a *one-versus-the-rest* test, by querying each image in the database against the remainder. Unless otherwise indicated, the full database is used in the tests. The clean database is used in shape related tests since certain shape features require a clear outer boundary of plants.

The main metric used in assessing performance is the top-N retrieval rates indicating whether the correct plant type is among the top N returned images. We used top-10 and top-15 retrieval rates, assuming that a user can easily and quickly identify the correct image among 10–15 returned images. In addition, we present the average minimum rank value which indicates the rank of the best matching correct plant, averaged over all queries.

All three feature classes (color, shape and texture) are tested with all possible parameters and retrieval methods that we proposed. In addition to these individual tests, several combined methods are tested as well. In summary, test results show that the most useful individual feature class is color, followed by texture and global shape, and that the performance of the system is increased when combining several features. Since the performance of the proposed method is still relatively low, we also include the performance of a dummy engine, which randomly selects the retrieved images, in the overall result tables. The following sections present test results with comments on the performance of the retrieval methods.

5.1. Results using color features

Results obtained using various color features are given in Table 1. Among distinct color feature analysis approaches, the nRGB color histogram provides the best top-10 and top-15 rates of 41 and 48%, respectively, while the second best performing color feature is the hue-based, modified color co-occurrence

TABLE 1. Color similarity results (full database).

Method	Top-10%	Top-15%	AvgMinRank
RGB hist. (256-bin)	0.25	0.32	71.9
nRGB hist.(512-bin)	**0.41**	**0.48**	**35.9**
Hue hist. (10-bin)	0.37	0.46	39.7
Hue hist. (30-bin)	0.36	0.44	38.9
Hue hist. (90-bin)	0.36	0.44	39.4
Color co-occ.	0.39	0.44	39.4
Color co-occ. (off-diag)	0.40	0.48	37.1
Random	0.15	0.23	74.0

matrix. The higher results obtained by nRGB and hue-based histograms, in comparison with RGB, may be attributed to the illumination resistance of nRGB and HSI color spaces. The modified color co-occurrence method, which ignores diagonal entries in the co-occurrence matrix, outperforms the conventional color co-occurrence method (48 versus 44% top-15 accuracy). On the other hand, the effect of smoothing the histograms were insignificant.

5.2. Results using shape features

Results obtained according to various global shape features are given in Table 2. The number of high-curvature points feature is more effective than other features with 28% retrieval rate at top-15. Combining all of the global shape features achieves a 26% top-10 and 31% top-15 rate.

To evaluate the full potential of the global shape descriptors, shape methods are also tested on the clean database of 132 images from 32 plant types. The results of this test, shown in Table 3, indicate that the best performing individual feature is the number of high-curvature points, as was also the case with the full database. Again as with the full database, combining all shape features provided the best performance, with 48 and 61% of top-10 and top-15 accuracies, respectively.

The results of the system using only SIFT features are displayed in Table 4, along with results of the combination of SIFT and global shape features. As can be seen in this table, the

TABLE 2. Global shape similarity results (full database).

Feature	Top-10%	Top-15%	AvgMinRank
No. of high curv. pts	0.22	0.28	75.9
Leaf arc length	0.15	0.21	86.9
Leaf base dist.	0.18	0.22	86.7
Leaf tip dist.	0.18	0.22	86.9
Norm. leaf arc length	0.15	0.21	88.1
All features	**0.26**	**0.31**	**79.9**
Random	0.15	0.23	74.0

TABLE 3. Global shape similarity results (clean database).

Feature	Top-10%	Top-15%	AvgMinRank
No. of high curv. pts	0.44	0.56	25.1
Leaf base dist.	0.45	0.55	25.2
Leaf tip dist.	0.43	0.55	28.6
Leaf arc length	0.34	0.50	28.1
Norm. leaf arc length	0.30	0.42	33.1
All features	**0.48**	**0.61**	**23.4**
Random	0.27	0.39	45.7

TABLE 4. SIFT + Global Shape Similarity Results.

Method	Top-10%	Top-15%	AvgMinRank
SIFT (full DB)	0.16	0.22	72.0
SIFT (clean DB)	0.29	0.38	27.9
SIFT + global	**0.44**	**0.56**	**22.0**
Random (full)	0.15	0.23	74.0
Random (Clean)	0.27	0.39	45.7

SIFT method by itself is not successful at all; in fact they are roughly the same as those of the dummy retrieval engine. In the clean database where images all have cleaner outer contours, the results are better (29 and 38% of top-10 and top-15 accuracies, respectively), but they are still very low and significantly lower compared with the results obtained using global shape features.

The main problem using the SIFT features is the lack of strongly defined keypoints on grayscale images of plants, since some plants only show a smooth variation in color. Using color SIFT features may address this problem to some extent. Part of the lower accuracy may also be attributed to the following. When using SIFT features, a Hough transform may be used in finding the global affine transformation that aligns two images [1]; this can be used to improve the similarity assessment, by ignoring erroneously matching SIFT features that are not in agreement with the global transformation. However, in our problem, an affine transformation does not model the mapping between snapshots of different instances of a given plant. The number, orientations and size of the leaves vary due to within-class variations of a given plant type. In addition, the lighting conditions in the environment vary from one image to another. Therefore, in the plant matching problem, the SIFT feature matching algorithm performs below expectations and the retrieval performance is degraded.

Using the combination of SIFT and global shape features does not improve the results of global shape feature results either. In fact, the top-10 and the top-15 rates drop (from 48 to 44% and 61 to 58%, respectively), while the average minimum rank improves by a small amount. We believe that this is due to the difficulty in setting the relative weights of the two

dissimilarity scores (SIFT and global shape). The normalization of SIFT-based dissimilarity is not straightforward, because the number of matching keypoints shows a very wide range from a few points to several hundreds for matching images in the database, while the typical number ranges around 20. On the other hand, using SIFT in combination with all other features (global shape, color, and texture) improve the accuracy, as shown in Table 7. Further research is needed to find the most effective normalization and weight parameters, when using SIFT features.

TABLE 5. Texture analysis results (full database).

Method	Top-10%	Top-15%	AvgMinRank
Default	0.27	0.36	53.2
Max-scale	0.19	0.28	61.8
Patch-based	**0.27**	**0.36**	**50.0**
Random	0.15	0.23	74.0

5.3. Results using texture features

Considering texture features, we compared the results of (i) the *default approach* where two images are compared in all scales, (ii) *max-scale* approach where two images are compared using only the scale with the highest energy and (iii) *patch-based* approach where individual patches are rotated to a canonical rotation, before obtaining Gabor responses. These three methods are explained in Section 3.2.3.

The results given in Table 5 show that the default and patch-based approaches give better and very similar results, compared with the max-scale approach. Both methods achieved 27% at

TABLE 6. Shape + color + texture analysis results (full database).

Shape	Color	Texture	Top-10%	Top-15%	AvgMinRank
All Global	RGB	Default	0.31	0.40	53.7
		Max scale	0.30	0.41	58.0
		Patch based	0.34	0.41	53.5
	nRGB	Default	0.42	0.51	39.9
		Max scale	0.41	0.50	42.5
		Patch based	0.41	0.49	40.7
	HSI (10-bin)	Default	0.39	0.45	42.8
		Max scale	0.34	0.44	46.7
		Patch based	0.41	0.47	43.9
SIFT	**RGB**	Default	0.46	0.54	30.6
		Max scale	0.45	0.54	32.9
		Patch based	**0.50**	**0.55**	**31.4**
	nRGB	Default	0.44	0.51	31.6
		Max scale	0.40	0.51	33.3
		Patch based	0.46	0.54	29.8
	HSI(10-bin)	Default	0.40	0.49	35.3
		Max scale	0.41	0.50	36.6
		Patch based	0.42	0.51	34.7
All global + SIFT	RGB	Default	0.45	0.53	38.8
		Max scale	0.44	0.53	40.9
		Patch based	0.46	0.53	38.2
	nRGB	Default	0.44	0.51	37.5
		Max scale	0.42	0.50	39.6
		Patch based	0.43	0.50	36.5
	HSI(10-bin)	Default	0.41	0.51	40.2
		Max scale	0.41	0.51	41.3
		Patch based	0.39	0.53	40.1
	Random		**0.15**	**0.23**	**74.0**

top-10 and 36% at top-15 retrieval accuracy, however, the patch-based approach retrieves the correct plants in a slightly higher rank (50 rather than 53.2). When used in combination with other features, the patch-based approach was found to be the most useful texture analysis method, as shown in Tables 6 and 7.

5.4. Results using color, texture, shape combinations

In order to see the full performance of the proposed system, we combined features from all three classes: color, shape and texture. Combined methods taking into account color, texture and shape-based features obtained improved results, as expected. The results including all three feature classes are shown in Table 6 for the full database. In this table, we also include the performance of a dummy (random) retrieval engine, in order to show the improvement of the proposed system over random retrieval. The most successful approach is to combine the SIFT + RGB + Patch-Based features, resulting in 50 and 55% as the top-10 and top-15 accuracy rates, respectively.

The combined results given in Table 6 do now show much improvements over the performance of individual features, but this is mostly due to the fact that the full database is not suitable to be used with the global shape features. In order to evaluate the newly proposed shape features, and their combination with SIFT features, new combined tests are run on the clean database consisting of well-segmented images. The results of these tests, given in Table 7, are the highest accuracy rates among all results and they show a performance gain obtained when using all three groups of features. Furthermore, they also show that the combination of SIFT and global shape features improves the overall system performance, supporting the claim of their complementarity. While almost each color, shape and texture feature combinations are tested, only outstanding methods are given here. With respect to the top-15 accuracy rates, the best-performing

TABLE 7. Shape + color + texture analysis results (clean database).

Shape	Color	Texture	Top-10%	Top-15%	AvgMinRank
All global	RGB	Default	0.52	0.61	21.3
		Max scale	0.51	0.59	22.4
		Patch based	0.53	0.61	20.7
	nRGB	Default	0.63	0.69	18.8
		Max scale	0.60	0.68	19.6
		Patch based	**0.63**	**0.71**	**18.6**
	HSI(10-bin)	Default	0.54	0.63	20.7
		Max scale	0.54	0.63	21.2
		Patch based	0.59	0.71	18.8
SIFT	RGB	Default	0.50	0.57	20.1
		Max scale	0.44	0.55	21.2
		Patch based	0.50	0.60	19.3
	nRGB	Default	0.53	0.66	18.2
		Max scale	0.54	0.62	19.0
		Patch based	0.53	0.66	17.7
	HSI(10-bin)	Default	0.44	0.59	20.2
		Max scale	0.47	0.57	21.1
		Patch based	0.50	0.57	18.9
All global + SIFT	RGB	Default	0.51	0.62	19.5
		Max scale	0.54	0.63	20.9
		Patch based	0.60	0.66	19.7
	nRGB	Default	0.64	0.72	18.0
		Max scale	0.61	0.69	19.1
		Patch based	**0.60**	**0.73**	**18.6**
	HSI(10-bin)	Default	0.59	0.68	19.7
		Max scale	0.57	0.66	19.9
		Patch based	0.58	0.68	18.7
	Random		**0.27**	**0.39**	**45.7**

method is the Global-shape + SIFT + nRGB + Patch-based method, with **63**% top-10 and **71**% top-15 rates.

The results of CBIR studies are normally evaluated based on *precision* and *recall* rates. Precision is defined to be the ratio of the relevant images in all of the returned images. Recall is defined to be the ratio of retrieved relevant images to all relevant images. These metrics are not very suitable in identification problems where higher precision values are desired, but sufficient to have only one relevant image among a small number of returned images. For identification problems, top-N results are often reported. Our problem is mostly an identification problem, considering that we are primarily interested in the correct match. However, retrieving similar plants are also useful, for instance, to find different varieties of a particular plant, or similar plants. Therefore, the problem can be seen as a retrieval problem, as well. We report top-N rates since assessing how similar or related other plant varieties are is out of the scope of this paper.

To get a better understanding of the system performance, we plot the probability of having a relevant result in the top-N results for changing N values, in Figs 7 and 8. In Fig. 7, the accuracy is plotted for a subset of the individual color, shape and texture methods, using the full database. Here, we observe that color methods are most successful (nRGB, hue and color co-occurrence), followed by patch-based Gabor texture, and then the global shape features. The patch-based approach is significantly more accurate than the default texture method , especially for larger N.

In Fig. 8, we plot the probability of having a relevant result in the top-N for changing N values, for outstanding individual and combined methods, using the clean database. As can be seen in this figure, the accuracy curves of two combined methods (nRGB + patch + shape and nRGB + default + shape) are indistinguishable and outperform the others.

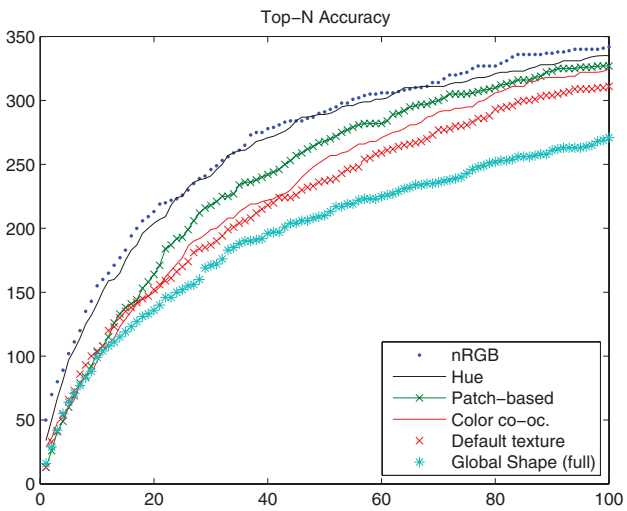

FIGURE 7. Accuracies of individual color, shape (global) and texture methods on the full database.

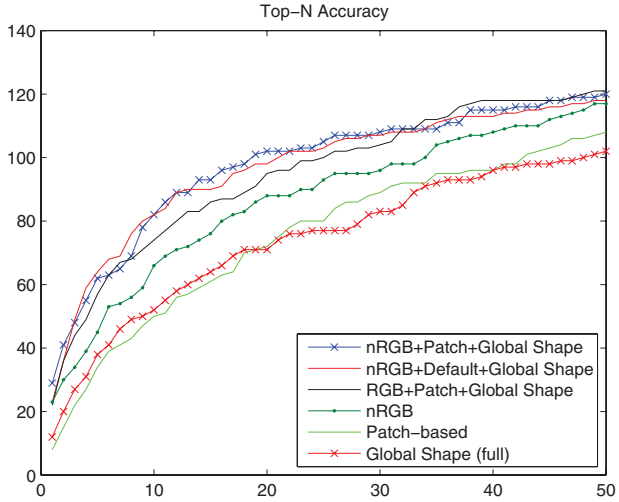

FIGURE 8. Accuracies of outstanding color, shape (global), texture and combined methods on the clean database.

The experiments are performed on a 1.60 GHz Celeron M CPU with 2 GB RAM. We developed the application on Microsoft Visual Studio 2005 and used the Windows Vista platform. Retrieval time for a segmented query image (with a medium size of 600 × 800) is 125 ms on average: 16 Gabor response images of the segmented image are produced in about 10–15 and the remaining feature extraction operations take 10 s. Segmentation is the longest process with an average of 90, 40–45 s of which is spent for the seed region selection.

5.5. Challenges

The primary challenge we have encountered in color analysis is caused indirectly by bad segmentation results. When the background region is not cleaned up smoothly, these regions effect and bias the generated color histograms. Additionally using hue histograms, there is the well-known problem of undefined saturation and hue values. Such values are obtained in two situations: (i) singularity problems cause zero saturation and undefined hue values; (ii) very dark and very bright points have saturation values of 0 and 1, respectively, while their hue values vary widely. To avoid undefined hue and saturation values, the system may be enhanced with additional controls on singularity points, and very dark or bright points. For instance RGB, or intensity values may be used as a color feature in such cases as proposed in [46]. In fact, we implemented a modification to ignore pixels with undefined or problematic values, but this attempt was not very successful, mainly due to ignoring white areas inside the plants.

The accuracy of image segmentation is the most important factor directly affecting the success of shape-based features, as well. Bad plant contours resulting from inaccurate segmentation (often due to textured background) is a common problem that leads to faulty feature extraction. Figure 9 depicts a typical case

FIGURE 9. Example for a bad (jagged) plant contour caused by inadequate segmentation. Left: Segmented image, Middle: Segmentation map to see the segmentation faults, Right: Traced contour of the image with spurious high-curvature points.

where the extracted jagged contour makes it look like the plant has very small leaves, while, in fact, the plant has a few long and smooth leaves.

The main challenge in texture analysis is due to the variance in image size and resolution, which cause the texture to appear in different scales using Gabor filters. Our texture analysis methods propose two alternative solutions: max-scale and patch-based methods which aim to overcome orientation and scale variance problems. In particular, in the patch-based approach, images are partitioned into fixed number of patches. This approach overcomes the image resolution variance problem by using a fixed number of patches per image, effectively dealing with the resolution problem. Moreover, this method partially overcomes the rotation variance problem of leaves, by rotating all patches to a canonical orientation.

Note that while we assume that the plant images clearly indicate the general structure or the outline of the plant, not all the pictures in the database are of this type. If a constraint is defined to regulate the plant size in the image (e.g. the image shows only a minimal amount of background to clearly show the outer contour of the plant), scale problems will only depend on image resolution and can be handled more easily.

6. SUMMARY AND CONCLUSION

We present a plant image retrieval system, combining various CBIR approaches with a segmentation preprocessing step. Extracting plant regions from images by the MFMC segmentation technique has given us an opportunity to focus solely on the plant, which increased consistency of the retrieved global features. Furthermore, combining different color, shape and texture features extracted from the images enhance the accuracy of the system.

Common techniques are used in color and texture feature extraction steps: color histograms, color co-occurrence matrices and Gabor filters. However, with a modification on Gabor texture analysis, we were able to focus on the local features of the approximated leaf regions obtained by our patch-based approach. The novelty of this technique is that rotating the approximated leaf patches to a canonical direction provides the effect of rotating the plant leaves to the same orientation. The proposed patch-based approach appears as the best texture method in the overall results in Table 7. For shape-based retrieval, we used SIFT features that capture local characteristics of the plant, as well as newly proposed global shape descriptors that are based on the outer contour of the plant. Both the modified Gabor texture method and new global shape descriptors provided improvements over the existing methods.

While there is clearly room for improvement, the proposed approach got promising results for the plant retrieval problem. Using color, shape and texture features in combination have improved the system performance. The highest top-15 identification rate obtained on the clean database (73%) is a combined method of nRGB histogram, local and global shape features and patch-based texture methods. Furthermore, we see that the average rank of the top matching image is 18.6, which also looks promising.

7. DISCUSSION AND FUTURE WORK

While a direct comparison is not meaningful, the fact many CBIR problems report precision values around 50–60% [19, 20, 44, 47], can give an idea of the difficulty of the CBIR problem. Shape-based retrieval in botanical collections [21], which is the closest study to our problem, reports precision rates of 0.92 at top-5 and 0.88 at top-10 retrieval ranks. These results are obtained on the public Swedish tree leaves database, which consists of 1125 isolated leaves from 15 different Swedish trees species. In the current problem, on the one hand we have more information (the whole plant image) which would help with the problem. On the other hand, the problem is more difficult in terms of the difficulty of extracting a single leaf from the full image.

Future studies will include expanding the database in order to make it useful as a practical application; extending SIFT implementation to color SIFT to improve its effectiveness; and evaluating new features such as spatial color histograms to address the particular problem. How to combine different features is also an important problem that we plan to study further.

8. FUNDING

The research was conducted at Sabanci University, Istanbul, Turkey.

REFERENCES

[1] Lowe, D.G. (1999) Object Recognition from Local Scale-Invariant Features. *ICCV*, pp. 1150–1157.

[2] Lowe, D.G. (2004) Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, **60**, 91–110.

[3] Eakins, J.P. (2002) Towards intelligent image retrieval. *Pattern Recognit.*, **35**, 3–14.

[4] Venters, C.C. and Cooper, D.M. (2000) A Review of Content-Based Image Retrieval Systems. Technical Report, Manchester Visualization Centre, Manchester Computing, University of Manchester, Manchester, UK.

[5] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A. and Jain, R. (2000) Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**, 1349–1380.

[6] Tico, M., Haverinen, T. and Kuosmanen, P. (2000) A Method of Color Histogram Creation for Image Retrieval. *Proc. Nordic Signal Processing Symp. (NORSIG'2000), Kolmarden, Sweden*, June, pp. 157–160.

[7] Gonzalez, R.C. and Woods, R.E. (2001) *Digital Image Processing*. Addison-Wesley, Boston, MA, USA.

[8] El-ghazal, A., Basir, O.A. and Belkasim, S. (2007) Shape-based Image Retrieval Using Pair-wise Candidate Co-ranking. In Kamel, M.S. and Campilho, A.C. (eds), *ICIAR*, pp. 650–661. Lecture Notes in Computer Science, 4633. Springer.

[9] Park, J.S. and Kim, T.-Y. (2004) Shape-based Image Retrieval Using Invariant Features. In Aizawa, K., Nakamura, Y. and Satoh, S. (eds) *Advances in Multimedia Information Processing—PCM 2004*, pp. 146–153. Lecture Notes in Computer Science. Springer, Berlin.

[10] Abbasi, S., Mokhtarian, F. and Kittler, J. (1999) Curvature scale space image in shape similarity retrieval. *Multimedia Syst.*, **7**, 467–476.

[11] Ma, W. and Manjunath, B. (1996) Texture Features and Learning Similarity. *Proc. Conf. Computer Vis. Pattern Recognition (CVPR)*, June, pp. 425–430.

[12] Han, J. and Ma, K.-K. (2007) Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image Vis. Comput.*, **25**, 1474–1481.

[13] Dengsheng Zhang, M.I., Aylwin, W. and Lu, G. (2000) Content Based Image Retrieval Using Gabor Texture Features. *Proc. 1st IEEE Pacific Rim Conf. Multimedia* (*PCM'00*), December, pp. 392–395.

[14] Manjunath, B. and Ma, W. (1996) Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**, 837–842.

[15] Liu, C. and Wechsler, H. (2001) A Gabor Feature Classifier for Face Recognition. *ICCV*, Vol. 2, pp. 270–275.

[16] Haralick, R.M. and Shapiro, L.G. (1992) *Computer and Robot Vision*. Addison-Wesley, Boston, MA, USA.

[17] Flickner, M. *et al.* (1995) Query by image and video content: the QBIC system. *IEEE Comput.*, **28**, 23–32.

[18] Bach, J.R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R.C. and Shu, C.F. (1996) Virage Image Search Engine: An Open Framework for Image Management. In Sethi, I.K. and Jain, R.C. (eds), *Storage and Retrieval for Still Image and Video Databases IV*, pp. 76–87. SPIE.

[19] Lew, M.S., Sebe, N., Djeraba, C. and Jain, R. (2006) Content-based multimedia information retrieval: state of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, **2**, 1–19.

[20] Veltkamp, R.C. and Tanase, M. (2000). Content-based image retrieval systems: a survey.

[21] Yahiaoui, I., Hervé, N. and Boujemaa, N. (2006) Shape-Based Image Retrieval in Botanical Collections. *PCM*, pp. 357–364.

[22] Wang, Z., Chi, Z. and Feng, D. (2003) Shape based leaf image retrieval. *IEE Proc.—Vis., Image and Signal Process.*, **150**, 34–43.

[23] Wang, Z., Chi, Z., Feng, D. and Wang, Q. (2000) Leaf Image Retrieval with Shape Features. *VISUAL'00: Proc. 4th Int. Conf. Advances in Visual Information Systems*, London, UK, pp. 477–487. Springer.

[24] Backes, A.R. and Bruno, O.M. (2010) Shape classification using complex network and multi-scale fractal dimension. *Pattern Recognit. Lett.*, **31**, 44–51.

[25] Bruno, O.M., Plotze, R.D.O., Falvo, M. and de Castro, M. (2008) Fractal dimension applied to plant identification. *Inf. Sci.*, **178**, 2722–2733.

[26] Lin, F.-Y., Zheng, C.-H., Wang, X.-F. and Man, Q.-K. (2008) Multiple Classification of Plant Leaves Based on Gabor Transform and LBP Operator. *ICIC*, Vol. 3, pp. 432–439.

[27] Man, Q.-K., Zheng, C.-H., Wang, X.-F. and Lin, F.-Y. (2008) Recognition of Plant Leaves Using Support Vector Machine. *ICIC*, Vol. 3, pp. 192–199.

[28] Woebbecke, D.M., Meyer, G.E., Bargen, K.V. and Mortensen, D.A. (1993) Plant Species Identification, Size, and Enumeration Using Machine Vision Techniques on Near-Binary Images. *Optics in Agriculture and Forestry*, pp. 208–219. SPIE.

[29] Yonekawa, S., Sakai, N. and Kitani, O. (1996) Identification of idealized leaf types using simple dimensionless shape factors by image analysis. *Trans. ASAE*, **39**, 1525–2533.

[30] Abbasi, S., Mokhtarian, F. and Kittler, J. (1997) Reliable Classification of Chrysanthemum Leaves Through Curvature Scale Space. *SCALE-SPACE'97: Proc. 1st Int. Conf. Scale-Space Theory in Computer Vision*, London, UK, pp. 284–295. Springer.

[31] Pérez, A.J., López, F., Benlloch, J.V. and Christensen, S. (2000) Colour and shape analysis techniques for weed detection in cereal fields. *Comput. Electron. Agric.*, **25**, 197–212.

[32] Jia, J. and Krutz, G. (1992) Location of the maize plant with machine vision. *J. Agric. Eng. Res.*, **52**, 169–181.

[33] D.G. Sena Jr, Pinto, F.A.C., Queiroz, D.M. and Viana, P.A. (2003) Fall armyworm damaged maize plant identification using digital images. *Biosyst. Eng.*, **85**, 449–454.

[34] Boykov, Y. and Kolmogorov, V. (2004) An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**, 1124–1137.

[35] Ford, L.R.F. and Fulkerson, D.R. (1956) Maximal flow through a network. *Canad. J. Math.*, **8**, 399–404.

[36] Rother, C., Kolmogorov, V. and Blake, A. (2004) "Grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, **23**, 309–314.

[37] Zarit, B., Super, B. and Quek, F. (1999) Comparison of Five Color Models in Skin Pixel Classification. *Proc. Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 58–63.

[38] Sural, S., Qian, G. and Pramanik, S. (2002) Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval. *ICIP*, Vol. 2, pp. II-589–II-592.

[39] Shim, S.-O. and Choi, T.-S. (2003) Image Indexing by Modified Color Co-occurrence Matrix. *ICIP*, September, Vol. 2, pp. III-493–III-496.

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

[41] Ledwich, L. and Williams, S. (2004) Reduced SIFT Features for Image Retrieval and Indoor Localisation. *Australian Conf. Robotics and Automation*.

[42] Wang, J., Zha, H. and Cipolla, R. (2005) Combining Interest Points and Edges for Content-Based Image Retrieval. *IEEE Int. Conf. Image Processing, 2005. ICIP 2005*, September, pp. 1256–1259.

[43] Šonka, M., Hlaváč, V. and Boyle, R.D. (1993) *Image Processing, Analysis and Machine Vision* (1st edn). Chapman & Hall, London, UK.

[44] Iqbal, Q. and Aggarwal, J.K. (2002) Combining Structure, Color and Texture for Image Retrireval: A Performance Evaluation. *16th Int. Conf. Pattern Recognition (ICPR)*, August, pp. 438–443.

[45] Arevalillo-Herráez, M., Domingo, J. and Ferri, F.J. (2008) Combining similarity measures in content-based image retrieval. *Pattern Recognit. Lett.*, **29**, 2174–2181.

[46] Plataniotis, K.N. and Venetsanopoulos, A.N. (2000) *Color Image Processing and Applications*. Springer New York, USA.

[47] Wong, Y.M., Hoi, S. and Lyu, M. (2007) An Empirical Study on Large-Scale Content-based Image Retrieval. *IEEE Int. Conf. Multimedia and Expo*, July, pp. 2206–2209.