Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 71–82
The Journal of Systems and Software
www.elsevier.com/locate/jss

Utilizing venation features for efficient leaf image retrieval

JinKyu Park [a], EenJun Hwang [a,*], Yunyoung Nam [b]

[a] School of Electrical Engineering, Korea University, Anam-dong, Seongbuk-Gu, Seoul, Republic of Korea
[b] Graduate School of Information and Communication, Ajou University, Wonchon-dong, Youngtong-Gu, Suwon, Kyunggi-Do, Republic of Korea

Received 20 December 2006; received in revised form 4 May 2007; accepted 7 May 2007
Available online 22 May 2007

Abstract

Most Content-Based Image Retrieval systems use image features such as textures, colors, and shapes. However, in the case of a leaf image, it is not appropriate to rely on color or texture features only as such features are very similar in most leaves. In this paper, we propose a new and effective leaf image retrieval scheme. In this scheme, we first analyze leaf venation which we use for leaf categorization. We then extract and utilize leaf shape features to find similar leaves from the already categorized group in a leaf database. The venation of a leaf corresponds to the blood vessels in organisms. Leaf venations are represented using points selected by a curvature scale scope corner detection method on the venation image. The selected points are then categorized by calculating the density of feature points using a non-parametric estimation density. We show this technique's effectiveness by performing several experiments on a prototype system.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Leaf image retrieval; CBIR; Classification; Venation; Parzen window

1. Introduction

In the last few years, due to the development of digital devices, computers, and network technologies, generating, processing and sharing digital images have become very popular. Consequently, huge amounts of digital images are available. As the number of digital images has increased, the need for sophisticated image retrieval has become extremely desirable. Traditional image retrieval applications have relied on textual information such as file names or keywords describing the image. However, as digital images are becoming so voluminous in number, attaching and memorizing such text information is not manageable any more by human beings. To help alleviate this problem, researchers are working on Content-Based Image Retrieval (CBIR) systems. For example, a CBIR can detect the main objects in an image and then generate automatically some useful information describing those objects including shapes, textures, and colors.

CBIR techniques have many diverse applications. Because of the vast popularity of hand-held devices with any where any place applications, CBIR techniques running on these devices can contribute to the strong trend of ubiquitous information retrieval. For example, during a field trip or visit to a botanical garden, people may encounter some unfamiliar plant. Instead of having to wait until they return home to look up the plant in a botanical book for detailed information or instead of having to carry a set of books with them, these people can get specific information on the spot by either drawing or taking a picture of it with their hand-held device and then using the drawing or picture as input to a query to a remote database by means of their PDA's wireless connection (Kim et al., 2005). As another example, if someone is on a fishing trip and wants to know immediately about the identification of some fish and some specific information about this fish that he has just caught, then he may rely on some CBIR technique by describing the fish's features to an application on his PDA. This application will then, based on the

* Corresponding author. Tel./fax: +82 2 32903256.
E-mail addresses: saanin@korea.ac.kr (J. Park), ehwang04@korea.ac.kr (E. Hwang), youngman@ajou.ac.kr (Y. Nam).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.05.001

supplied information of features, provide him with information about the fish (Sonobe et al., 2004).

For effective Content-Based Image Retrieval, the CBIR system needs to figure out and then represent most effective feature points from an image and based on those features, provide a match with any similar images from the database. Content-Based Image Retrieval typically uses images' features such as textures, colors, or shapes.

In the case of a leaf image, it may not be appropriate to rely on color or texture features because such features are similar in almost all leaves of many different varieties. Instead, leaf shapes can provide some clue to finding a fewer number of similar varieties. In addition, if we examine the leaf venation, we can easily figure out that specific varieties of leaves have distinct venation patterns. The leaf's venation corresponds to patterns of blood vessels of organisms. In this paper, we propose an original leaf image retrieval scheme which first analyzes the image for the leaf's venation for a first level of leaf categorization and then extracts and utilizes shape features from the image to find similar leaf images from the corresponding categorized group in the database. Leaf venations are represented using points which are selected by the curvature scale scope corner detection method on the leaf's venation image. Categorization is performed by calculating the density of feature points using a non-parametric estimation density.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 describes a venation-based leaf image categorization scheme which collects the leaf venation's feature points and then calculates their distribution using the Parzen window (Parzen, 1962). Based on this distribution, the type of leaf venation is identified. Section 4 describes several experiments. The last section concludes the paper and discusses future work.

2. Related work

Well-designed image retrieval tools enable people to make more efficient uses of collections of digital images. Typical image retrieval systems in the late 1970s were mainly based on keyword annotation. This approach suffered from many difficulties including the vast amounts of human labor required to provide the annotations and the challenging problem of maintaining annotation consistency among images in large databases. In order to overcome these difficulties, there has been extensive research in the last decade on Content-Based Image Retrieval (CBIR). Examples of some of the prominent systems using this approach are VIRAGE (Hamrapur et al., 1997), QBIC (Niblcack et al., 1993), Photobook (Pentland et al., 1994), and VisualSEEk (Smith and Chang, 1996).

In many CBIR systems, an image is represented by low level features such as color, texture, and shape. Relevant images are retrieved based on the similarity of their image features. A color histogram of an image is frequently used as a color feature. This approach is simple and quick to process. However, this feature represents only a global property of an image. Generally, it is hard to describe a local property such as the shape and direction of an object in an image. Chan and Chen (2004) proposed a color-complexity feature and a color-spatial feature to solve these problems. Texture features can be obtained by applying a Wavelet Transform to an image. This technique is used in several image analysis applications including texture classification and segmentation, image recognition, image registration, and motion tracking (Manjunath and Ma, 1996). In particular, shape-based image retrieval is regarded as an efficient and interesting approach. For example, shape recognition methods have been proposed and implemented for use in face recognition, iris recognition, and fingerprint recognition. In cases where images show similar color or texture, shape-based retrieval can be more effective than other approaches using color or texture. For instance, leaves of most plants are green or brown; but the leaf shapes, themselves, are usually distinctive and can thus be used for identification.

There are shape representation schemes which extract feature values or vectors from an image. These schemes identify an object from an image by extracting its contour and then representing the contour as feature vectors. In these schemes, we should concentrate on some conditions. For example, if we take pictures of an object varying the position of the camera, objects appear a little different from each other in the pictures. The scale of an object in the picture depends on the distance between the camera and the object. The object may be translated or rotated based on the direction from the camera to the object. Though an image is translated, rotated, or scaled, it is good to keep the feature values unchanged in order to identify the object consistently. In other words, the feature values should be invariant to image translation, rotation, and scaling. In this section, we introduce some shape representation schemes, chain-codes, the Fourier descriptor, the Center–Contour Distance Curve (CCD), the Minimum Perimeter Polygon (MPP), and the CSS.

Chain-codes represent a contour as a connected sequence of line segments. The line segments have predefined lengths and directions. There are two types of chain-codes according to the number of directions, four-directional chain-codes and eight-directional chain-codes (Gonzalez and Woods, 2002). Fig. 1 shows these two types of chain-codes. They use numbers to represent directions. There is some research involving representing a leaf contour with eight-directional chain-codes. One of the weaknesses of chain-codes is that a noise in a contour cause changes in the code. A leaf has small teeth at its boundary. These teeth can be considered as noises in shape representation. Gouveia et al. (1997) use a heuristic way: if the angle of the apex is less then 155°, then the apex point is a tooth and is removed from the chain-codes sequence. In this step, chain-codes representation presents an easy angle measuring scheme.

Lee et al. (2003) select Fourier descriptors as a feature vector of an image. They (Lee et al., 2003) approximate a

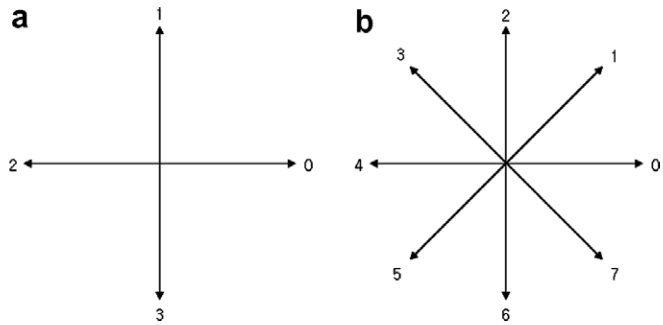

Fig. 1. (a) Four-directional chain-codes and (b) eight-directional chain-codes.

contour as a polygon and use bend angles to represent the polygon. This method represents a contour as $T(l)$ and the bend angle as a function of normalized accumulated length l. Fig. 2 shows $T(l)$.

The detailed procedure is as follows. To reduce complexity, Lee et al. (2003) use a curve evolution which removes irrelevant points of the contour while maintaining the characteristics of the contour. By this method, the contour is approximated to a polygon as shown in Fig. 3. To obtain $T(l)$, this technique involves calculating the bend angles tracing the polygon and normalizing them by tracing the length for the scaling invariant. By applying a Fourier transform to $T(l)$, the Fourier descriptor can be calculated as feature vectors of the image. Lee et al. (2003) propose a similarity measuring method. This method involves calculating the power spectrum from the descriptors and then getting the Euclidean distance between the two power spectrums. This representation is invariant to translation, rotation, and scaling. $T(l)$ does not contain any specific position values of points. It only deals with relations between the contour points. So, even if an image is translated, the feature vectors are not changed. Because this representation does not contain orientation information, this representation meets the rotation invariant requirement. For scaling invariance, $T(l)$ is normalized.

The Center–Contour Distance Curve (CCD) generates a center distance curve of a contour. An example of this curve is shown in Fig. 4. First of all, this method calculates the center of the contour and measures distances between the center and points of the contour. These distances can be expressed as a curve representing the contour. Wang et al. (2000) use the CCD to represent leaf images. For the first step, their method extracts a center distance curve function $f(i)$, the distance between the center and ith contour point. Then, this method calculates the dissimilarity by accumulating differences of two curve functions. Based on this dissimilarity, Wang et al.'s method (Wang et al., 2000) retrieves similar leaf images from a database. Because the CCD has a native invariance to image translation, the distance curve does not depend on specific x–y coordinate values. For scaling invariance, $f(i)$ is normalized

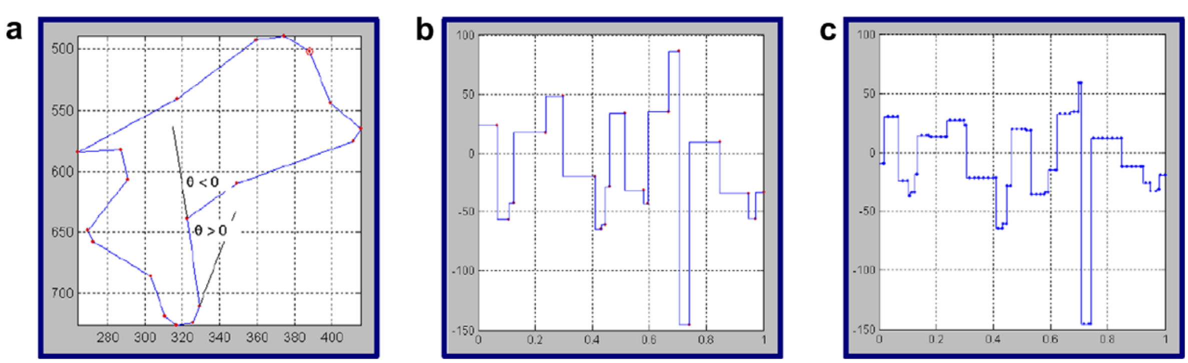

Fig. 2. (a) Bend angle, (b) bend angle vs. normalized length, (c) data samples for similarity measurement (Lee et al., 2003).

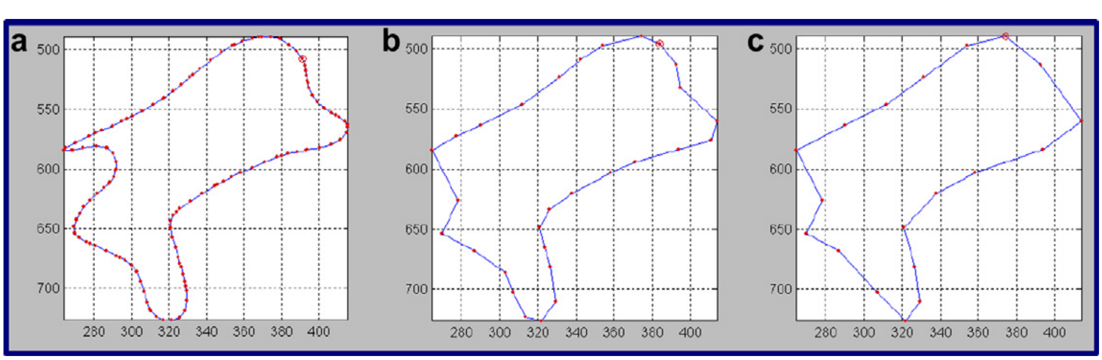

Fig. 3. (a) The original contour with 172 data points reduced to (b) 30 points and (c) 20 points by curve evolution (Lee et al., 2003).

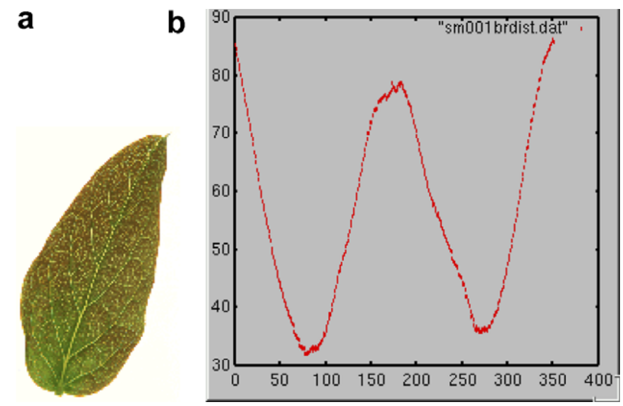

Fig. 4. (a) A sample leaf image and (b) the center distance curve (Wang et al., 2000).

as a contour length to the range [0, 1]. In the dissimilarity measurement, this method calculates n-dissimilarities by shifting one curve n times, where n is the number of points. We select their minimum as the dissimilarity of two images. This guarantees rotation invariance.

A Minimum Perimeter Polygon (MPP) (Sklansky et al., 1972) can be used to approximate a leaf contour into a polygon. We enclose the contour by a set of concatenated cells. We think of the contour as a rubber band contained within the walls. If this rubber band shrinks, it produces a polygon of minimum perimeter (Gonzalez and Woods, 2002). Fig. 5 shows an example of a MPP. Vertices of this polygon are considered as feature points. A set of x–y coordinates of vertices of the polygon values is used as feature vectors (Nam and Hwang, 2005). The dissimilarity is measured as a sum of distances between two sets of feature points. Nam and Hwang (2005) use a rotational adjustment algorithm for rotation invariance that detects left, right, top, and bottom points for scaling invariance. Im et al. (1998) use a set of line segments connecting two adjacent vertexes and angles at vertexes as feature vectors.

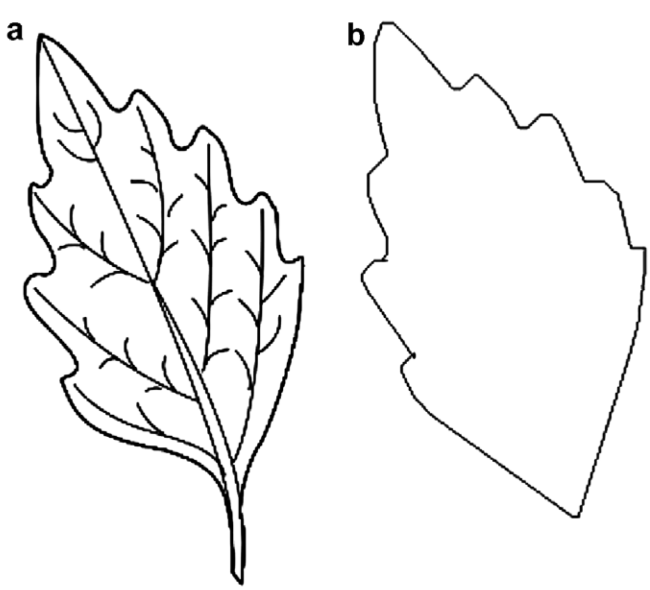

Fig. 5. (a) The original leaf image and (b) the MPP result (Nam and Hwang, 2005).

Mokhtarian et al. (1996) use the maxima of curvature zero_crossing contours of the Curvature Scale Space (CSS) image as a feature vector to represent the shapes of object boundary contours (see Fig. 6). Because values of these feature vectors do not depend on the specific x–y coordinate values, this representation has translation invariance. For scaling invariance, this method normalizes the horizontal coordinate in a range of [0, 1]. The shape queries using image database (SQUID) (http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html) is an example of shape-based image retrieval using CSS. This searches for similar images from a Database (DB) which has 1100 fish images based on CSS.

For the last example, a wavelet transform can also be used to represent a shape. Lee and Kim (2001) convert a full color image into a gray-scale image and then extract edge information. Then, they apply a discrete wavelet transform to the edge information. They get wavelet coefficients. Finally, they obtain a feature vector from these coefficients.

We have reviewed some shape representation schemes. Some of them are used in the leaf image retrieval area (Gouveia et al., 1997; Wang et al., 2000; Nam and Hwang, 2005; Im et al., 1998). In addition to them, there have been studies about leaf image retrieval, classification, and adjustment for effectiveness (Lin et al., 2004; Wang et al., 2000; Im et al., 1999; Mokhtarian and Abbasi, 2004). Most of these studies use only leaf contours, the outer shape features. On the other hand, a leaf venation represents its distinct vein pattern as inner shape feature and thus may be used to categorize leaves. One of the benefits from this venation-based categorization is that we can reduce the search space for matching by just looking into those leaves with same venation feature. In this paper, we propose an automatic categorization scheme for leaf images based on the venation feature. This scheme can be appended to most existing content-based leaf image retrieval systems and improve their retrieval accuracy and efficiency.

For the venation-based categorization, venation feature should be extracted from leaf images. Fu and Chi (2003) used the intensity histogram of vein pixels and trained artificial neural network (ANN) classifier to extract leaf veins. Other known extraction methods include active contours (Li et al., 2005) and b-spline (Kirchgessner et al., 2002). In this paper, we do not cover the venation feature extraction process but focus on categorizing extracted venation samples.

3. Venation-based categorization

In this section, we will categorize a leaf image using leaf venation which expresses the inside features of a leaf. Leaf venation is an arrangement of veins and veinlets. Leaf veins

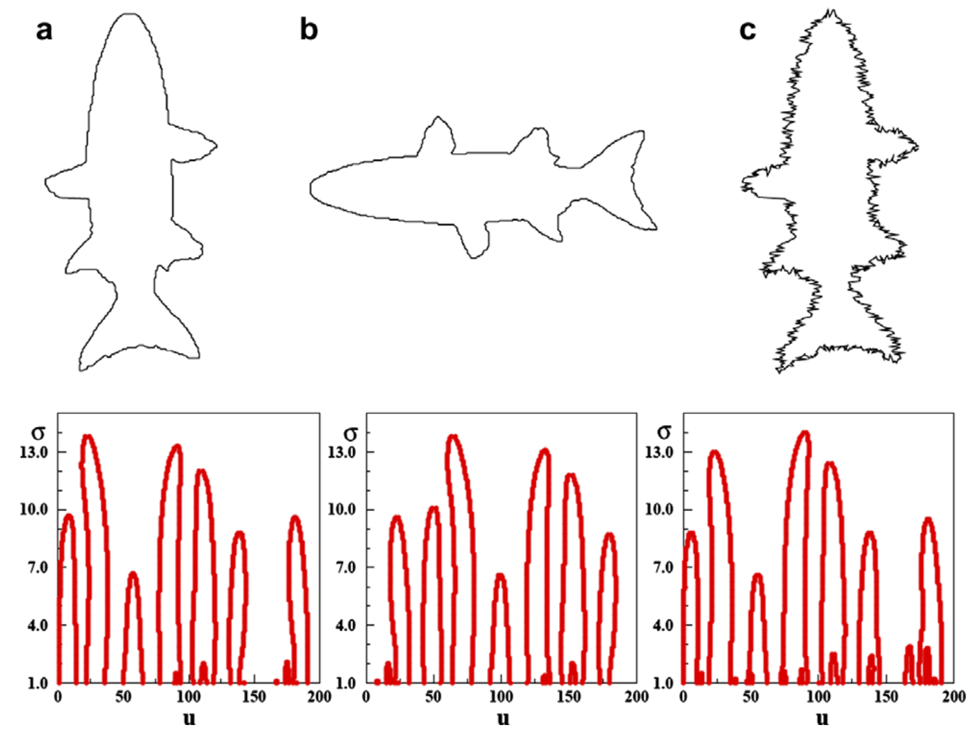

Fig. 6. (a) A boundary and its CSS image, (b) the change in orientation causes a circular shift in the CSS images and (c) noise creates small contours in the CSS image (Mokhtarian et al., 1996).

correspond to the blood vessels of organisms that carry minerals and oxygen. We can see the venation on a leaf surface. First of all, we explain the types and characteristics of leaves that are generally observed from leaves. Then, we show how to select feature points for the representation of venation. Finally, we describe how to identify the type of leaf venation (categorization) based on the distribution of those selected feature points.

3.1. Venation types of leaves

Fig. 7 shows four typical venation types (Lee, 1999) used for categorization in this paper. The reason for this is that these four venation types, which are taken from the illustrated plant book for plants in Korea (Lee, 1999) are representative ones among domestic aquatic plants and give good performance with the outer shape feature considered together.

From Fig. 7, type (a) is called a pinnate venation. It has a large primary vein and several secondary veins. These veins look like a bird's feather and are split from the primary vein. Types (b) and (c) are called parallel venation. In type (b), we can observe many veins which are parallel up to the end of the leaf and split from the petiole. The secondary veins are not clear or ladder-like. Type (c) has a primary vein with several secondary veins that are parallel up to the end of the leaf. For the last type, type (d) is a palmate venation in which three or more primary veins are split from the petiole and form a palm-like shape.

The key characteristic of pinnate venation is that there exists one primary vein with secondary veins that are split from the primary vein. Therefore, for this type of categorization, we need to find out the points where the secondary veins get split. We then check the distribution of those points. If the observed distribution is in the line-type, then this line is considered as a primary vein. The points along this line are where the secondary veins get split.

The key characteristic of parallel venation is the fact that all the veins merge at the end of the leaf. In this case of parallel venation, we check the distribution of the points where the vein is ended. If there are heavy densities at the top of the leaf, then the leaf is considered to have parallel

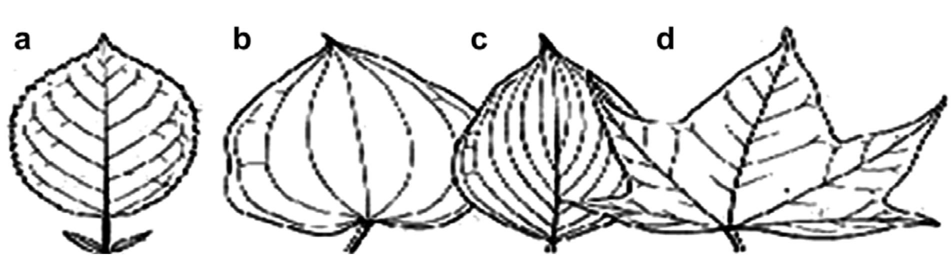

Fig. 7. Four types of venation.

venation. The most efficient way to separate types (b) and (c) would be to check locations with heavy density where a vein gets split. Another method to separate them would be to check whether the leaf has a line-type distribution or not.

In the case of a palmate venation, we can observe that the vein gets split at the bottom of the image. So the palmate venation can be distinguished by checking the distribution of points where the vein gets split.

3.2. Leaf feature extraction

As mentioned earlier, the feature points of leaves are where the venation gets branched and ended. By checking the distribution of such feature points, we can categorize a given leaf into one of the four different venation types. If the leaves in the database were also categorized by the venation type, then we can greatly reduce the number of images to consider for matching. This section explains the Curvature Scale Space Corner Detection algorithm (Mokhtarian and Suomela, 1998) in order to extract those feature points. This algorithm calculates curvature from the points on curves, and extracts some points as corners which have the maximum value of curvature. In the case of venation, the feature points are the maximum curvature points.

3.2.1. Edge detection

Before we apply the CSS algorithm to an image, we first perform Canny Edge Detection (Canny, 1986) to detect the shape of the venation. In Fig. 8, the original venation image is on the left side. On the right side is the image which results from an application of the Canny Edge Detection algorithm to the original image.

During this step, if the leaf venation is somehow broken or the vein is too thin, the leaf image is recognized as having several curves. An ideal case is when the shape of the venation is detected as a single closed loop. If we have several curves resulting from the reasons already mentioned, the order of feature points extracted will be mixed up and cause a problem which we explain later. This problem can be solved by adding an image pre-processing step. That is, before the Edge Detection process, making the venation thicker and increasing the contrast of the image could solve the broken vein problem.

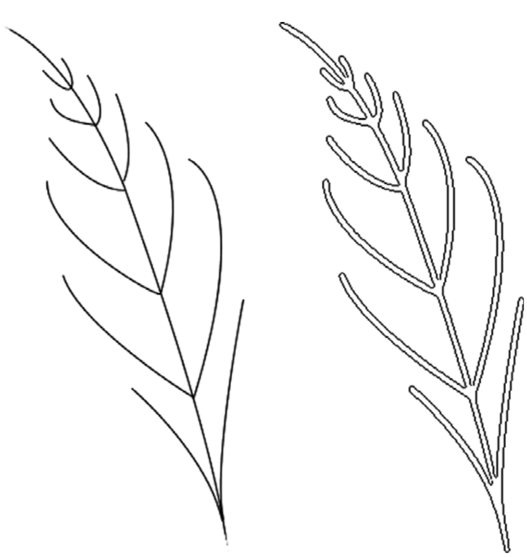

Fig. 8. Venation (left) and edge detection (right).

3.2.2. Feature points detection

Feature points can be obtained from the curves in the previous step. The CSS algorithm can now be used at this point. Applying the algorithm to the image on the right side of Fig. 8 gives the venation feature points_where the curvature gets maximum values. At these points, the venation gets branched and ended.

However, as you can see from Fig. 9, these two points are detected where the venation gets branched. This detection is because they both have a maximum curvature value. In order to solve this problem, we need to perform the following process.

Two feature points do not necessarily represent one position. Therefore, we first calculate the angles of the maximum value points to the curve for these two points. We then detect a point which has less than a 90 degree angle value. In the case of Fig. 9, the black point, which is located below, will be ignored. The white point will be selected as a feature point. The feature points after applying the CSS algorithm to the image of Fig. 8 are shown in Fig. 10.

3.2.3. The Branching Points/Ending Points distinction

As you might already notice, extracted feature points can be classified either as Branching Points (BP) or as Ending Points (EP). For example, for the ending point ① in

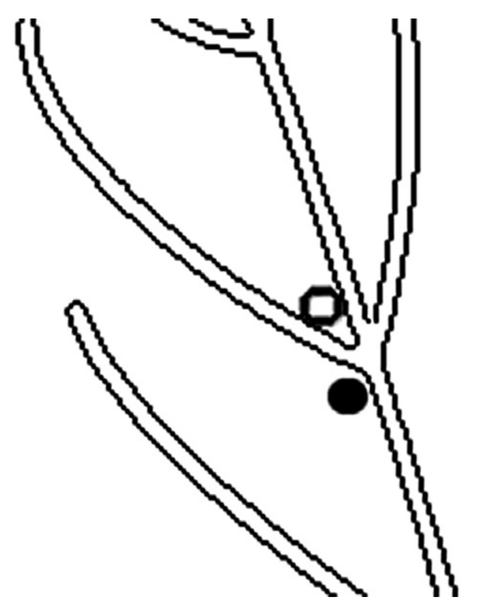

Fig. 9. Maximum curvature positions.

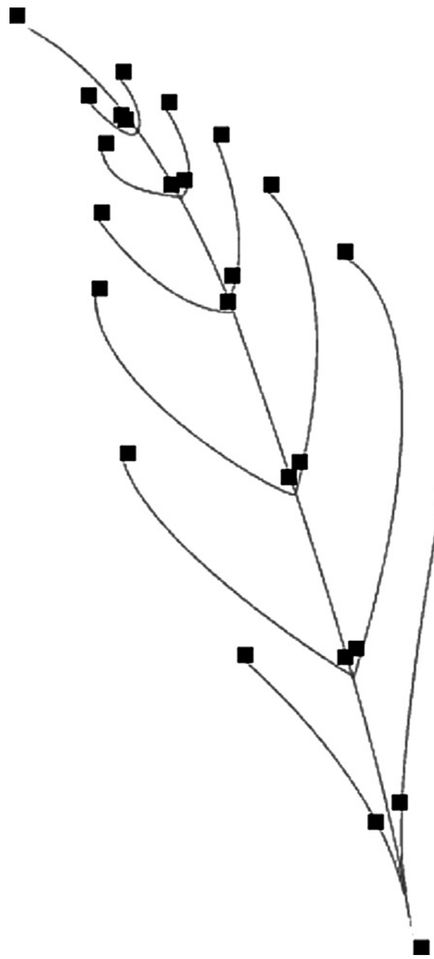

Fig. 10. Feature points from the CSS algorithm.

Fig. 11b, the direction changes from the ending point to the left side and changes to the right side at the branching point ②. As in Fig. 11c, when there are three connected points (C_1, C_2, C_3) along the proceeding direction, making a decision of BP/EP for the middle point C_2 is based on the previous proceeding direction of $\overrightarrow{C_1C_2}$. This means checking the location of the point C_3, (whether C_3 is located at an upper position or a bottom position) will be the decision maker. If C_3 is located at the upper position, then the proceeding direction will be changed to the left and C_2 will be the EP based on the previous proceeding direction. If it is the other way, then C_2 will be the BP. If the angle between the line $\overrightarrow{C_1C_2}$ to the x-axis is θ, then rotating C_3, which has C_2 as a reference point and checking the y-coordinate of C_3 which is rotated up to $-\theta$ will be enough. As in Fig. 11a, the proceeding direction is counter-clockwise and if the y-coordinate of C_3 has positive value, then it is an Ending Point; if the coordinate has a negative value, then it is a Branching Point. The detailed algorithm is showed in Table 1.

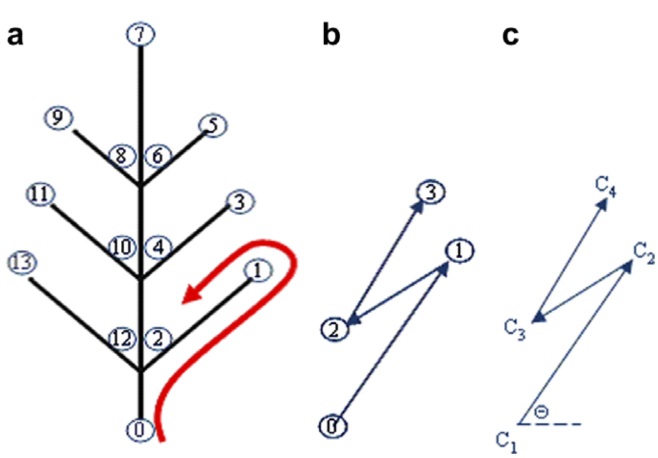

Fig. 11. (a) Feature points, (b) direction and (c) feature point distinction.

To judge the preceding direction, we set the bottom venation starting point as a base point and check whether the direction is clockwise or counter-clockwise as compared to the x-coordinates of a previous point and later point.

Fig. 12 shows the result of applying this method to the feature points in Fig. 10. In Fig. 12, grey points are where the venation gets branched (Branching Point) and black points are where the venation gets ended (Ending Point).

3.3. The density distribution of feature points

To classify the venation, we need to decide whether the feature points are distributed along a line or around one point. The density of Branching Points and Ending Points can be calculated by the Parzen window method (Parzen, 1962) which can estimate a non-parametric density. This method estimates the density function from limited data. We first explain how to get the standard line that is necessary to calculate the density of the feature points and then how to decide the distribution type by calculating the density of distances from this standard line.

3.3.1. The pseudo primary vein

To check whether the feature points have a line-type distribution or point-type distribution, we need to obtain the density of feature points by calculating the distance

Table 1
An algorithm for distinguishing feature points

function CornerDistinct (C_1, C_2, C_3, *direction*)
{
  $\theta \leftarrow$ an angle between vector $\overrightarrow{C_1C_2}$ and x-axis
  $C_3' \leftarrow$ rotation of C_3 around C_2 at $-\theta$
  if $C_3' \cdot y > 0$
    state ← Ending Point
  else
    state ← Branching Point
  end if
  if *direction* is counter-clockwise
    return state
  else
    return !state
  end if
}

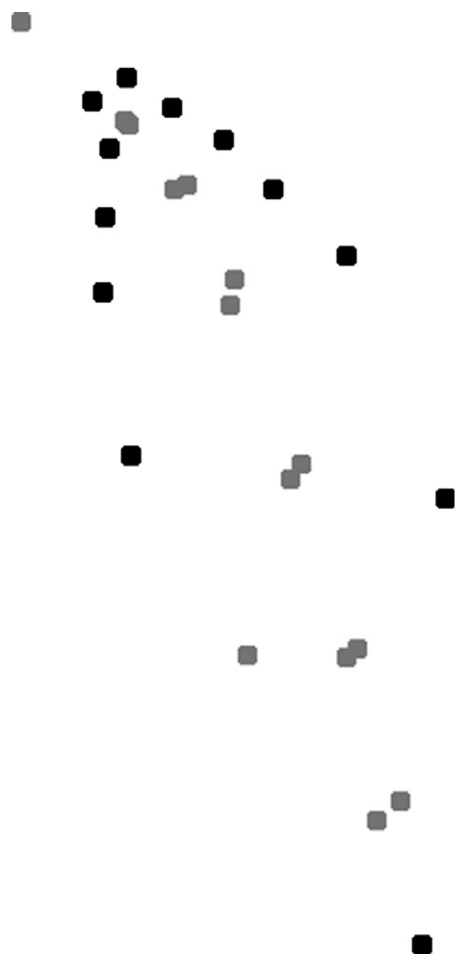

Fig. 12. The BP (gray-dot) and the EP (black-dot).

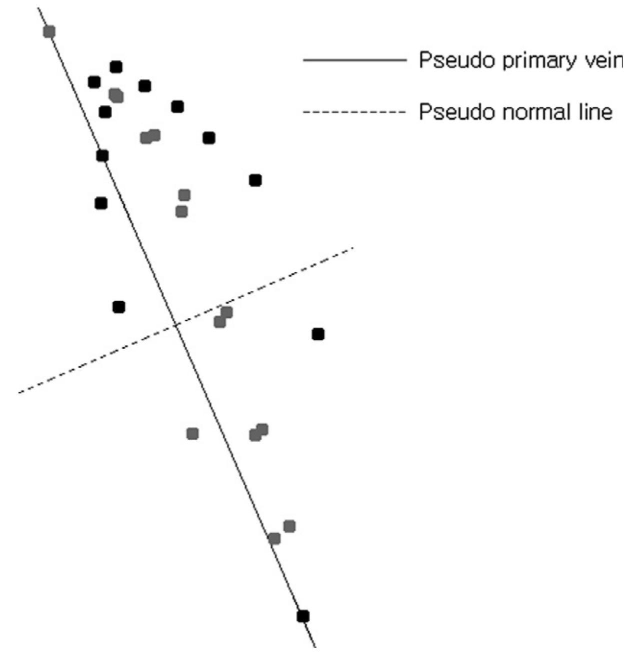

Fig. 13. A pseudo primary vein and pseudo normal line.

between some line and feature points. The lines to be used here are the pseudo primary vein and the pseudo normal line (see Fig. 13). The pseudo primary vein will be the line that connects from the top point of the venation to the bottom point. The calculated perpendicular line is used as the pseudo normal line. We can check the row distribution by calculating the density between the BP and imaginary venation line and find where the density gets a maximum value. Similarly to this, by calculating the density between the BP or other feature points and the pseudo normal line, we can possibly check the column distribution.

3.3.2. Parallel venation verification

As mentioned before, the characteristic of parallel venation is that there are points densely dispersed at the end of the leaf. To check this type of distribution, we have to see the column distribution of feature points. Table 2 shows the algorithm for calculating the density of distances between a line and feature points. The *w_size* of the density function is a Parzen window size. From an experiment, the size, 70 pixels, is selected as an optimal window size for the parallel venation verification. This experiment will be explained in Section 4.

Table 2
The algorithm for calculating density

```
function Density (distances, w_size)
{// distances is an array of distances of corners from a line
  minDist ← min(distances)
  maxDist ← max(distances)
  for each r such that minDist ⇐ r ⇐ maxDist
    sum ← 0
    for each d in distances
      if |r − d|/w_size < 0.5
        sum++
      end if
    end for each
    kde[r] ← sum
  end for each
return kde
}
```

In order to get the distance where the density gets a maximum value, we need to check the number of maximum values. If the number of maximum values is one and the average distance of feature points that is close to this maximum value is located above the venation, then we may consider it a parallel venation. If the number of maximum values is two and the average distances of feature points on each maximum value are distributed above and below the venation, then we may consider it a parallel venation too. Fig. 14 shows a sample graph from calculating the density of feature points in the parallel venation.

3.3.3. Relationships of Branching Points

In this step, we calculate the density of BPs to check whether the primary vein does exist or not and to decide whether it is a palmate venation or not.

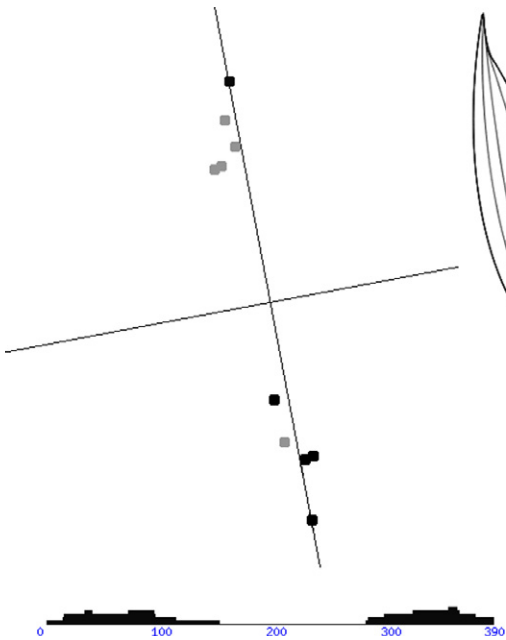

Fig. 14. Calculating the density.

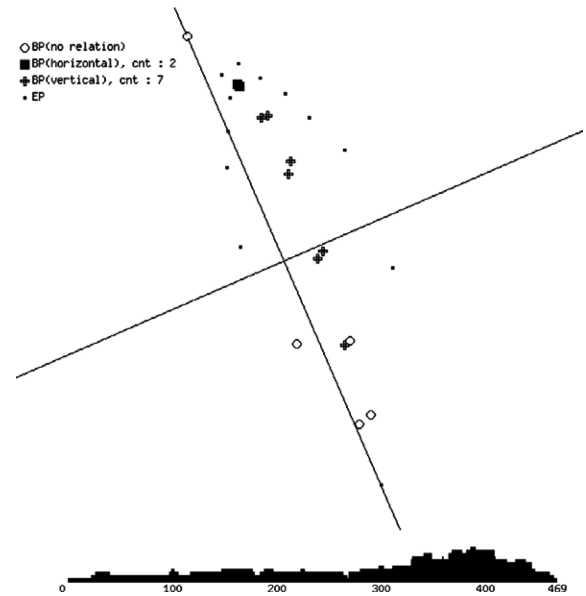

Fig. 15. Distribution of Branching Points.

To check whether the distribution of BPs is line-shaped or densely dispersed around one point, we need to calculate both the row density and the column density. We can find points where the density gets to a maximum value by calculating the row and column density. Then we need to check BPs whose distance is around this maximum value and decide whether they are related or not. Namely, we can get the vertically related BPs which have the pseudo primary vein as a standard line and the horizontally related BPs from the pseudo normal line. This process requires the density function again. However, the Parzen window size is not the same as before. Because the previous one is for ascertaining the positions of maximum densities, the size is large. But, to distinguish the highly concentrated BPs in each direction, the window size should be small. If it is large, some of unrelated BPs would be found as related ones. This may make an ambiguity between the pinnate and palmate venations. From an experiment, the size of 10 pixels is selected as an optimal window size.

If the leaf image contains a real primary vein, vertically related BPs form a line parallel to the pseudo primary vein. Like a palmate venation, if some BPs are gathered around one point, the BPs are related in both directions. Fig. 15 shows the relationships among BPs using this method. In the figure, small circles indicate BPs with no relationships. They are considered as secondary Branching Points or noise and are ignored in the vein classification process. In the figure, a black box indicates a horizontally related BP and a cross indicates a vertically related BP. Finally, a small black point is an EP.

3.3.4. Venation classification

Parallel venation is identified by a scheme described in Section 3.3.2. If a vertically related BP is dominant, a real primary vein is found and the BPs are on it. In this case, the leaf image can be classified as a pinnate venation or a parallel venation. If a maximum density value is found at the top of the venation, this leaf will be classified as a parallel venation. Otherwise, it is a pinnate venation. Dominant BPs of palmate venation have relations in both directions.

4. Experiments

In this section, we explain two types of experiments. First, we observe classification results varying the window size to get optimal Parzen window sizes. Second, we show the improvements of leaf image retrieval by applying this classification method.

4.1. The Parzen window size

As mentioned in Sections 3.3.2 and 3.3.3, this classification scheme needs two Parzen window sizes. We create an experiment to get an optimal window size which minimizes the number of misclassified images. By varying each Parzen window size, we observe numbers of well-classified and misclassified images.

Figs. 16 and 17 show numbers of misclassified images in the parallel venation verification process. From Fig. 16, parallel venations strongly depend on the Parzen window size of this process. From Figs. 16 and 17, we see that a size of 70 pixels is an optimal size.

The second Parzen window size is used in the relationship of branching points. Similar to the above, we find an optimal window size and the results are shown in Figs. 18 and 19. In Fig. 18, we find that the window size influences pinnate venation and palmate venation. In contrast to the above, the total number of misclassified images

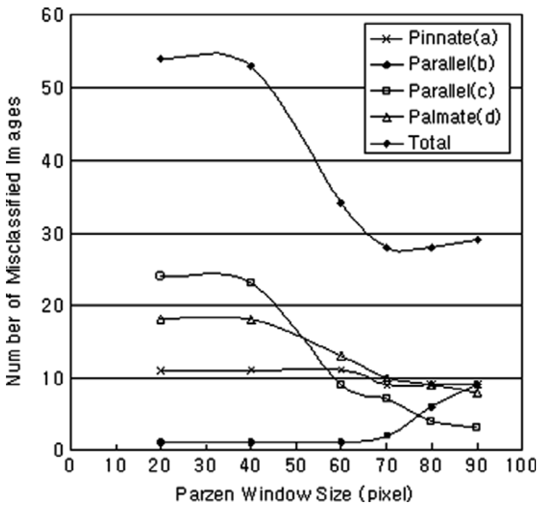

Fig. 16. The influence of Parzen window size in parallel venation verification.

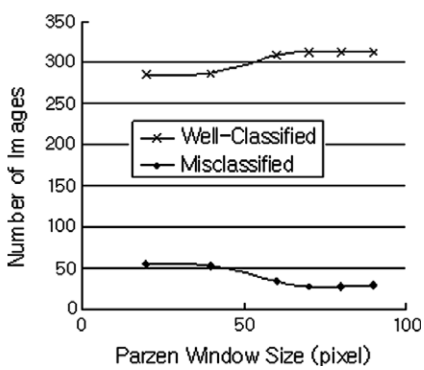

Fig. 17. Numbers of well-classified and misclassified images in parallel venation verification.

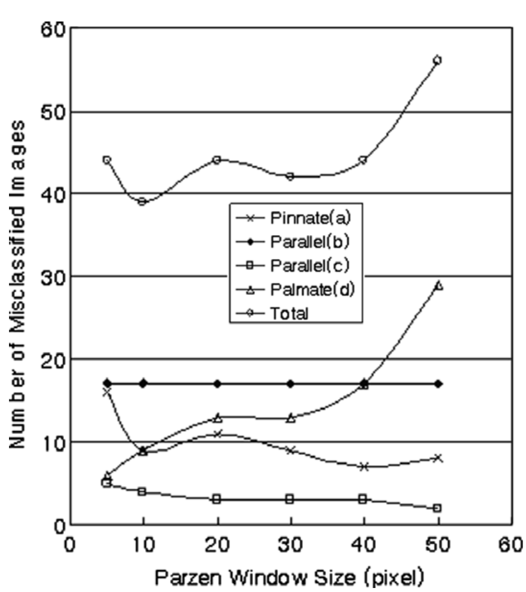

Fig. 18. The influence of the Parzen window size in relationship to Branching Points.

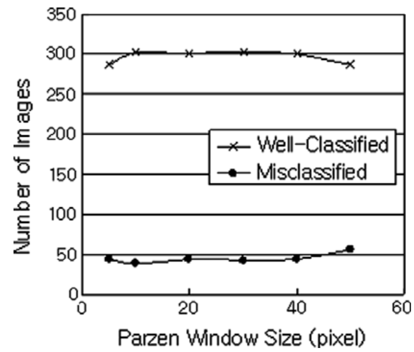

Fig. 19. The numbers of well-classified and misclassified images in relationship to Branching Points.

increases as a Parzen window size increases. As mentioned in Section 3.3.3, unrelated BPs may be found as related BPs when the size increases. They make an ambiguity between pinnate and palmate venations.

4.2. Leaf image retrieval

In order to measure the effects of the venation-based categorization in leaf image retrieval, we have used the CLOVER system (Nam et al., 2005) as our test-bed. CLOVER is a shape-based image retrieval system that we have built for retrieving domestic aqua-plants in Korea. The algorithms were tested on a PC with a Pentium 4 of 3.0 GHz for its CPU and a 1 GB RAM. In the experiment, we used 350 illustrated leaf images from *Illustrated Flora of Korea* (Lee, 1999), which contains pictures of native plants in Korea. MATLAB was used to calculate the feature points from the images. Also, we used PHP to categorize leaf venations from the feature points. Fig. 20 and Table 3 show the results.

In Fig. 20, a query image is drawn by a user. For the comparison, we have performed two types of retrievals: shape-based retrieval only and shape-based retrieval with venation categorization which has been described in this paper. Each retrieval scheme calculates image similarities between the query image and images in the database. In the shape-based retrieval with venation categorization, we categorize the query image first and then calculate the similarity against database images that belong to the same category as the query image. Based on these steps, Fig. 20 shows the 10 most similar images in the database.

Table 3 shows average rates of success for retrievals where the querying images are found in the top 1%, 2%, 3%, 4% and 5% of retrieved images. From Table 3, we can observe that our categorization scheme enhances the retrieval effectiveness significantly.

The average time taken to extract the feature points by using MATLAB was 0.55 s, and the average time to categorize the leaf venation using this result was 0.08 s.

By incorporating this scheme, we can reduce the time for retrieving similar images from this database. Ideally, if there are N types of venations and M images in the

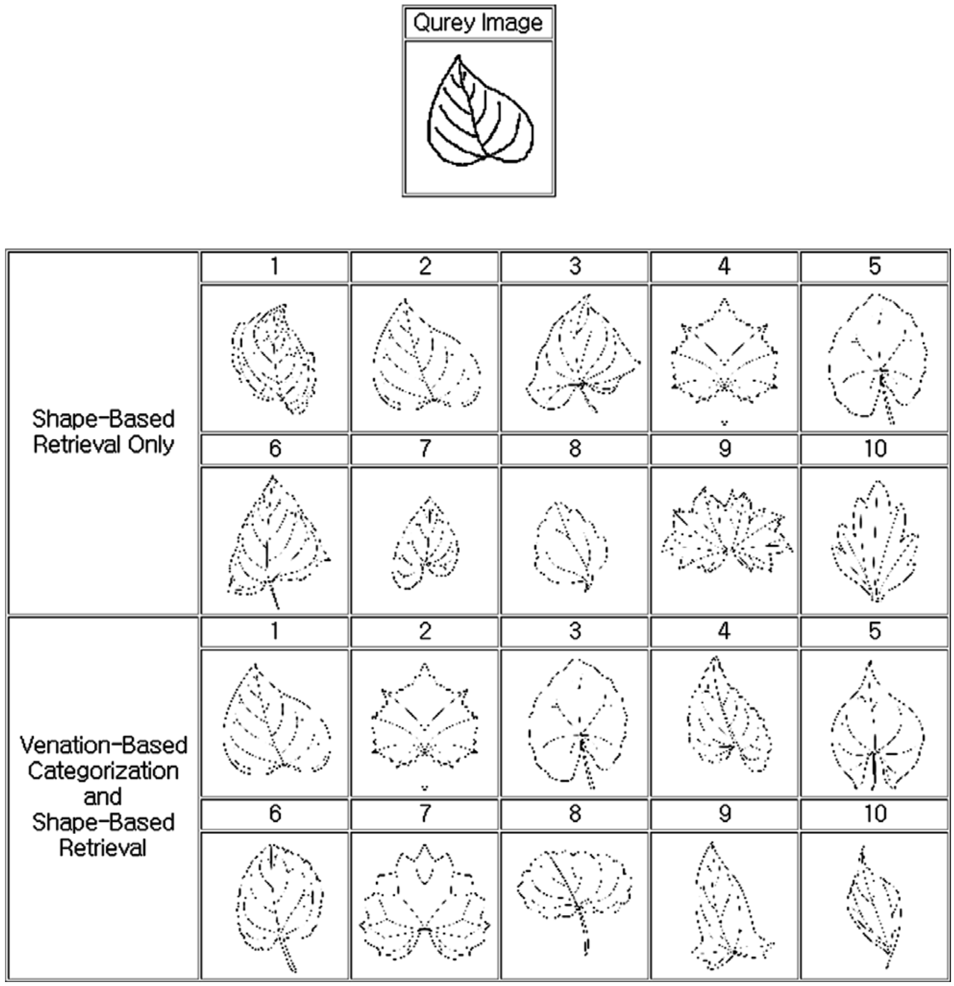

Fig. 20. Image retrieval results.

Table 3
Average retrieval success rates

Method	Success rate (%)				
	1	2	3	4	5
Without categorization	25	58.33	58.33	83.33	83.33
With categorization	50	60.67	75	100	100

database, then by this scheme, we can reduce the search space by M/N using the proposed categorization scheme.

5. Discussion and conclusions

In this paper, we have presented a new and unique leaf image categorization scheme by using the its venation feature. Typically, there are several common venation types among leaves. By extracting and representing these venation types, we can improve the retrieval performance by reducing the number of images to be searched for. Through many experiments, we have shown that recognizing and categorizing venation features has taken very little time (less than 1 s); whereas it has enhanced the retrieval time and accuracy significantly.

Even though we just considered four representative venation types in the paper, our proposed scheme can be extended easily to handle new venation types. To include a new venation type, all we need to do is to identify a distinct criterion from the venation type for the categorization purpose and plug into the categorization engine. Another issue that we would like to mention is the broken vein problems which are observed from real world leaf images. Real world images such as digital camera images may contain imperfect leaf, folded leaf or overlapped leaves. If parts of a leaf image are missing, it is very difficult to predict such missing parts and to recover them is beyond the scope of this paper. In a research on the shape-based leaf image retrieval (Mokhtarian and Abbasi, 2004), they proposed how to retrieve leaf images even when leaves in the image are self-intersected. This result may help to solve the broken vein problem of the latter two cases. However, to identify intersections in the experiment, they put a light source behind leaves when they took picture of them, which is not a reasonable assumption for general leaf images.

Our scheme can be extended to handle other kinds of objects. For example, animals have their own distinct bone structure. In the case of fishes, bones consist of one main backbone and many ribs from it. If we have X-ray images of fishes, then our algorithm can categorize them based on the bone structure.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC support program supervised by the IITA and a grant (No. BDM0100211) to JRL from the Strategic National R& D Program through the Generic Resources and Information Network Center funded by the Korean Ministry of Science and Technology.

References

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6), 679–698.

Chan, Y.K., Chen, C.Y., 2004. Image retrieval system based on color-complexity and color-spatial features. Journal of Systems and Software 71, 65–70.

Fu, H., Chi, Z., 2003. A two-stage approach for leaf vein extraction. In: Proceedings of International Conference on Neural Networks and Signal Processing (ICNNSP2003), vol. I, Nanjing, Jiangsu, China, December 12–15, 2003, pp. 208–211.

Gonzalez, R., Woods, R., 2002. Digital Image Processing, second ed. Prentice Hall.

Gouveia, F., Filipe, V., Reis, M., Couto, C., Bulas-Cruz, J., 1997. Biometry: the characterisation of chestnut-tree leaves using computer vision. In: Proceedings of the IEEE International Symposium on 7–11 July, vol. 3, pp. 757–760.

Hamrapur, A., et al., 1997. Virage video engine. In: SPIE Proceedings on Storage and Retrieval for Image and Video Databases V, San Jose, February, pp. 188–197.

Im, C., Nishida, H., Kunii, T.L., 1998. Recognizing plant species by leaf shapes – a case study of the Acer family. Fourteenth International Conference on Pattern Recognition, 1171–1173.

Im, C., Nishida, H., Kunii, T.L., 1999. Recognizing Plant Species by Normalized Leaf Shapes. In: Vision Interface '99, Trois-Rivi?res, Canada, 19–21 May, pp. 397–404.

Kim, S., Tak, Y., Nam, Y., Hwang, E., 2005. mClover: mobile content-based leaf image retrieval system. LNCS (ICADL 2005), vol. 3815, pp. 139–148.

Kirchgessner, N., Scharr, H., Schurr, U., 2002. Robust vein extraction on plant leaf images. In: 2nd IASTED International Conference Visualization, Imaging and Image Processing (VIIP 2002), Malaga, Spain, 9–12 September 2002.

Lee, C.B., 1999. Illustrated flora of Korea. ISBN-8971871954, Hangmoonsa.

Lee, D.H., Kim, H.J., 2001. A fast content-based indexing and retrieval technique by the shape information in large image database. The Journal of Systems and Software 56, 165–182.

Lee, D.J., Antani, S., Long, L.R., 2003. Similarity measurement using polygon curve representation and Fourier descriptors for shape-based vertebral image retrieval. In: Proceedings of IS&T/SPIE Medical Imaging 2003: Image Processing, vol. SPIE 5032, San Diego, CA, February, pp. 1283–1291.

Li, Y.F., Zhu, Q.S., Cao, Y.K., Wang, C.L, 2005. A Leaf Vein Extraction Method Based On Snakes Technique. International Conference on Neural Networks and Brain 2005. ICNN& B'05. vol. 2, 13–15 Oct. 2005, pp. 885–888.

Lin, H.J., Kao, Y.T., Yen, S.H., Wang C.J., 2004, A study of shape-based image retrieval. In: Proceedings of the 6th International Workshop on Multimedia Network Systems and Application, pp. 118–123.

Manjunath, B.S., Ma, W.Y., 1996. Texture feature for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8), 837–842.

Mokhtarian, F., Abbasi, S., Kittler, J., 1996. Robust and efficient shape indexing through curvature scalespace. Proceedings on British Machine Vision Conference, 53–62.

Mokhtarian, F., Suomela, R., 1998. Curvature scale scope based image corner detection. In: Proceedings of the European Signal Processing Conference, Greece, pp. 2549–2552.

Mokhtarian, F., Abbasi, S., 2004. Matching shapes with self-intersections: application to leaf classification. IEEE Transactions on Image Processing 13 (5), 653–661.

Nam, Y., Hwang, E., 2005. A Shape-Based Retrieval Scheme for Leaf Image. In: Lecture Notes in Computer Science, vol. 3767. Springer-Verlog, pp. 876–887.

Nam, Y., Hwang, E., Kim, D., 2005. CLOVER: a mobile content-based leaf image retrieval system. In: International Conference on Asian Digital Libraries No. 8, Bangkok , Thailand, vol. 3815, pp. 139–148.

Niblcack, W., Barber, R., et al., 1993. The QBIC project: querying images by content using color, texture, and shape. SPIE 1908, 173–187.

Parzen, E., 1962. On estimation of a probability density function and mode. Annals of Mathematical Statistics 33, 1065–1076.

Pentland, A., Picard, R.W., Sclaroff, S., 1994. PhotoBook: tools for content-based manipulation of image database. SPIE 2185, 34–47.

Sklansky, J., Chazin, R.L., Hansen, B.J., 1972. Minimum perimeter polygons of digitized silhouettes. IEEE Transactions on Computers 2, 34–47.

Smith, J.R., Chang, S.F., 1996. VisualSEEk: fully automated content-based image query system. In: Proceedings of the ACM International Multimedia Conference & Exhibition, November, pp. 87–98.

Sonobe, H., Takagi, S., Yoshimoto, F., 2004, Mobile computing system for fish image retrieval. In: Proceedings of International Workshop on Advanced Image Technology (IWAIT), Singapore, January, pp. 33–37.

Wang, Z., Chi, Z., Feng, D., Wang, Q., 2000. Leaf image retrieval with shape features. Lecture Notes in Computer Science 1929, 477–487.