Oregon State University

Abstract

CONTENT-BASED IMAGE RETRIEVAL: PLANT SPECIES IDENTIFICATION

by Ashit Gandhi

Chairperson of the Supervisory Committee: Professor Thomas G. Dietterich
Department of Computer Science

With rapid increase in the number and size of image databases, there has been intensified interest in automated management and retrieval of digital images. In this work we develop a technique for automatic classification of digitized images of plant leaves. The inducement behind this work is to contribute to research in the field of computer vision and content-based image retrieval. The algorithms presented in this report can be applied in general for shape-based image retreival. They can be employed for querying and retrieving similar shapes that may be deformed, occluded and/or overlapped.

The performance of the technique built in this work was evaluated on six different plant species belonging to two separate genera. When applied to classify hand picked isolated leaves, the method yields 96% correct decisions. Classifying digitized plant images from the herbarium, with the training set consisting of only isolated leaves, yields an accuracy of 59%. An accuracy of 61% is obtained in classifying isolated leaves when the training set is constructed from the herbarium samples.

Table of Contents

Aknowledgements

It has been a great opportunity to work with Dr. Thomas Dietterich. I got a chance to think independently and received most appropriate guidance and monitions. Special thanks to Dr. Eric Mortensen for help provided in implementing the image-processing modules. I appreciate the support, comments and suggestions I received from all the students in the Machine Learning Group.

I acknowledge the help provided by Dr. Aaron Liston, Dr. Scott Sundberg, Ms. Thea Cook and others from the Plant and Botany Department, Oregon State University. Information on various plant species and the collection of digitized herbarium images were sourced from them.

1 Introduction

Computer vision and pattern recognition is an intrinsic part of recondite research in computer technologies. It has application in industrial inspection, biometrics, security, document analysis, robotics applications, image/video indexing and retrieval, underwater industrial applications, etc. In recent times, in areas like academia and entertainment, large collections of digital images have been created. Many of these collections are the product of digitizing existing collections of analogue photographs, drawings and paintings. With rapid increase in the number and size of image databases, there has been intensified interest in automated management and retrieval of digital images.

In this work we develop a technique for the automatic classification of digitized images of plant leaves. The method employs a dynamic programming algorithm [BEL57] for efficiently and accurately matching unblemished, deformed, occluded and overlapping leaf shapes. The inducement behind this work is to contribute to research in the field of computer vision and content-based image retrieval [GR97, RHC99]. The algorithms presented in this report can be applied in general for shape-based image retreival. They can be employed for querying and retrieving similar shapes that may be deformed, occluded and/or overlapped. Similar work has been done by [GMK88][PDM02]. They have considered distorted, occluded and partial matches but not overlapped shapes and they have experimented mainly with artificially created shapes.

1.1 Background

Digital image databases have seen an enormous growth over the past few years. There is a great need for automated, content-based methods that could help users retrieve, browse, classify or structure image databases. Users are exploiting the opportunity to access remotely stored images in all kinds of new and exciting ways. This has led to increased interest in research on content-based image retrieval (CBIR). CBIR essentially is retrieval of images based on features automatically extracted from the images themselves.

CBIR at its core is a computer vision and pattern recognition problem. Several approaches and their combinations can be used for image recognition. Image data can be compared using intensity (color and texture) and geometry (shape). Color features are relatively robust to background complication and independent of image size and orientation. Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity. It contains important information about the structural arrangement of surfaces. Shape representation can be divided into two categories, boundary-based and region-based. The former uses only the outer boundary of the shape while the latter uses the entire shape region. The Fourier Descriptor [PF94] is a representative of the boundary-based category, while the Moment Invariants [MKH77] is a representative of the region-based category. Shape, color and texture feature extraction depend on good image segmentation [RP98, SC95]. Accurate segmentation is highly desirable for shape features, while for other features, a coarse segmentation may suffice.

For our method, boundary shape is the main feature. Generally, after boundary detection, a scalar transformation [LON98] is performed. In this, the shape is described indirectly by means of a one-dimensional characteristic function of the boundary instead of the two-dimensional boundary itself. The one-dimensional function is easier to process. The scalar transformation can be information preserving, where accurate reconstruction of the shape is possible or information non-preserving, which is incapable or only partially capable of reconstructing the original shape.

1.2 Plant Species Identification: An Overview

We designed a method to identify species from digitized plant leaf images. Figure 1 shows the kinds of images the method has to deal with. These images are the result of digitizing samples from a large herbarium[1] collection. The way it is desired to work is as follows. We create a database consisting of isolated leaves or overlapped and occluded leaf images. These samples in the database library have their species labeled. For an incoming leaf (isolated or from the herbarium sample), whose species is to be determined, we retrieve from the database all samples that have a similar shape. The species is predicted based on the species of the annotated samples retrieved from the database.

Figure 1: Sample digitized images
(Hand collected isolated leaf in center
and samples from digitized herbarium on sides)

The method described here employs a discriminative approach as opposed to a model-based [CD86] approach. Discrimination is based on the external shape (boundary) of the leaves. The process first extracts the boundary from the leaf silhouette. A boundary scalar transformation transforms the boundary into a series of angles measured along the contour. Pattern matching is then performed using this scalar representation to retrieve similar leaves from the annotated database. A dynamic programming technique described in section 2 performs the matching of the whole, deformed, occluded and overlapped leaf

[1] The sample images are from the Herbarium at the Plant and Botany Department, Oregon State University

shapes. The species of the unknown leaf is then predicted based on the retrieved information. This whole proceeding is elucidated in section 3.

The pattern-matching algorithm is educed from the dynamic time warping [KL83] algorithm used for speech recognition and the sequence alignment algorithm [GOT82] used on biological sequences. Speech recognition using dynamic time warping can be defined as the problem of finding the minimum distance between a set of template streams and the input stream. The class chosen is the closest template. The algorithm for aligning biological sequences stems from the pioneering work of Needleman & Wunsch [NW70]. Sellers' [SEL74] work on a similar algorithm, later generalized by Waterman et al. [WSB76], allows multiple-sized insertions and deletions (gaps) of any length. The Dynamic time warping, the biological sequence alignment algorithm and a dynamic program for shape matching are described in the next section.

The performance of the algorithm developed in this project was evaluated using six different species belonging to the genera *Acer* (maples) and *Quercus* (oaks) [PK00]. For each species, we have both isolated and herbarium samples. The performance was evaluated for different configurations obtained by permuting isolated leaves and herbarium samples for construction of the template database and the evaluation test set. The experiment and its results are described in section 4.

2 Pattern Matching using Dynamic Programming

This section describes the application of dynamic programming to pattern matching. First, matching isolated spoken word signals by dynamic time warping is explained. A simple extension of this dynamic programming algorithm can perform shape matching. Algorithms for matching biological sequences exist that allow similar sequences containing multiple-sized gaps to be matched. One such algorithm is discussed here. Finally, a dynamic programming formulation for partial and overlapped cyclic patterns is developed. Expressing shapes as cyclic patterns offers translation and rotation invariance.

When comparing two patterns, it is necessary to examine the distances between every feature element of both sequences. This is conveniently done with a distance table. The columns represent the feature vector elements of one pattern and the rows represent the feature vector of the other. Each entry in the table is the distance between the two features corresponding to the given row and column. Distances between matching features of the two sequences will be smaller than the rest of the distances. Pattern matching can then be formulated as finding a minimum distance path, i.e. finding the path through the table with minimum sum of distances of the features compared in the path. This minimum sum or the cost of the minimum distance path gives a measure of how well the two sequences match.

Let M be the number of elements in the known pattern represented by rows in the distance table. Let the unknown pattern have N elements represented by the columns. The entry in the i^{th} row and j^{th} column of the distance table then corresponds to the distance between element i of the known contour and segment j of the unknown contour for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. The distance between the elements could be Euclidean distance or some other measure, which is inversely related to the similarity of the elements. The criterion for minimum distance path completeness is that the path must make use of all N elements of the unknown pattern. This corresponds to a path that begins in the first column of the distance table and ends in the last column. Due to the sequential nature of the patterns, the path may only proceed up and to the right (considering bottom-left corner to be the cell corresponding to first row and first column).

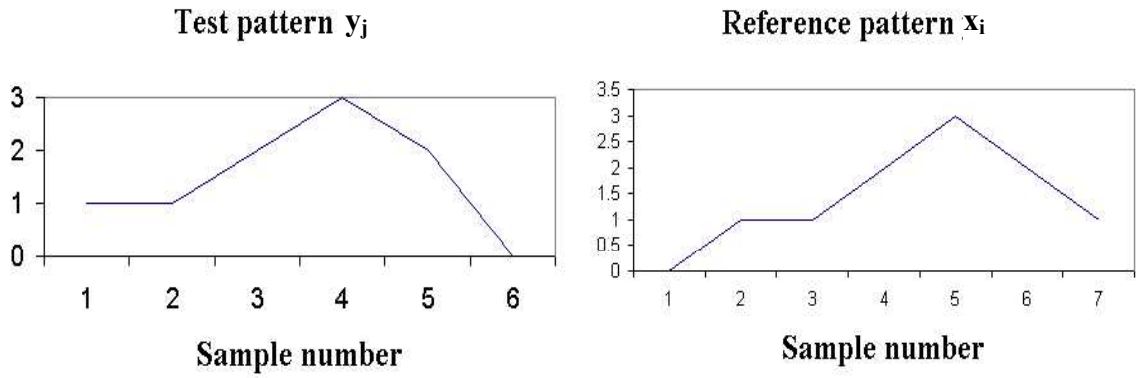
2.1 Dynamic Time Warping (DTW)

One of the earliest approaches to isolated-word speech recognition was to store a prototypical version of each word (called a template) in the vocabulary and to compare incoming speech with each word, taking the closest match. A dynamic programming technique called dynamic time warping can find the closet match. The cost of the minimum distance path in the distance table is calculated for each template and the incoming signal pair. The template that gives the lowest cost is then considered the best prediction for the incoming word stream.

Comparing the template with incoming speech might be achieved via a pair wise comparison of the feature vectors in each signal. The total distance between the sequences would be the sum or the mean of the individual distances between feature

vectors. The problem with this approach is that if constant window spacing is used, the lengths of the input and stored sequences are unlikely to be the same. Moreover, within a word, there will be variation in the length of individual phonemes. The matching process needs to compensate for length differences and take into account the non-linear nature of the length differences within the words. The Dynamic Time Warping algorithm achieves this goal; it finds an optimal match between two sequences of feature vectors, which allows for stretched and compressed sections of the sequence.

Suppose we wish to compare and evaluate the difference (minimum distance path) between the following two signals shown in figure 2. Both signals are similar in that they are single-peaked. However, the stored reference signal is longer than the test signal, and the peak is later.



a) (Input) test signal, y_j 1 1 2 3 2 0

b) (Stored) reference signal, x_i 0 1 1 2 3 2 1

Figure 2: Test and reference patterns for DTW

To calculate the minimum distance path, consider the distance table for sequences x_i and y_j with distance $d_{i,j}$ between i^{th} element of x and j^{th} element of y calculated as:

$$d_{i,j} = | x_i - y_j |$$

The table consists of seven rows and six columns. Figure 3 shows the feature vector difference table. Note that in this figure, the unknown (test) pattern is shown along the horizontal axis and templates (reference) along the vertical axis. There is a sequence of low numbers (shaded), close to the diagonal, indicating which samples of x_i are closest in value to those of y_j. This sequence actually corresponds to the minimum distance path. In this example, $D_{6,7}$ (as calculated using algorithm given in figure 4) gives the cost of best match between the two signals.

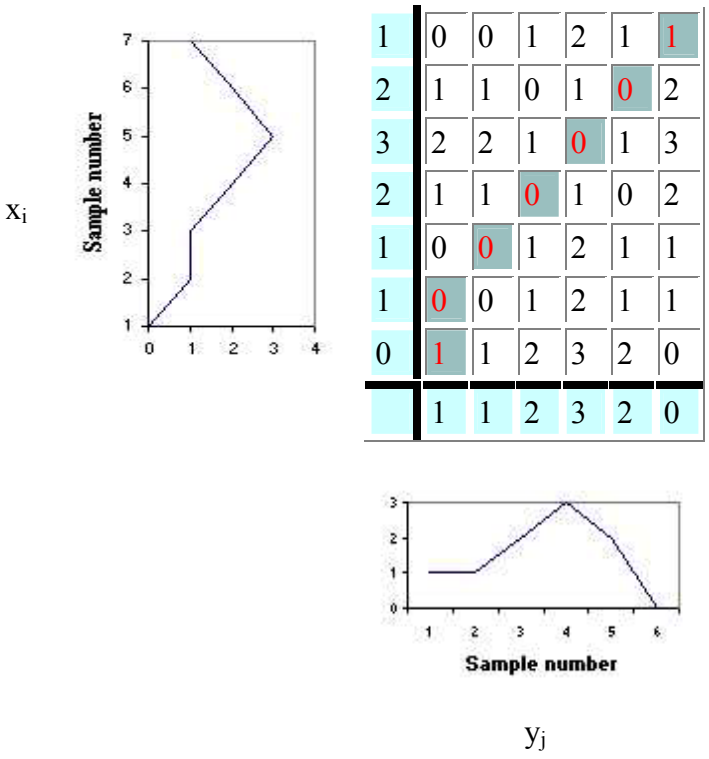


Figure 3: DTW Distance Table

For speech signals, following holds:

1) The endpoints of the two signals correspond to one another. That is, the path always begins at (1,1) and ends at (6,7), or whatever the coordinate of the upper-right cell happens to be.

2) Time moves forwards. This constraint can be satisfied by allowing just three kinds of move: up, right, or up-and-right. This means that there are only three ways of arriving at a particular cell: from below, from the left, or from below-left.

3) The best path lies near to the diagonal. In other words, all other things being equal, a diagonal move going upwards and to the right should be favored over simple upward or rightward moves. This constraint can be satisfied if we penalize the cost of a non-diagonal move, i.e. add extra cost for a pure upward or rightwards move. Alternatively, the cost of a diagonal move can be discounted, i.e. cost it at, say 50% of the cost of a move upwards or rightwards.

The Dynamic programming approach can efficiently calculate the cost of minimum distance path in *O(NM)* where *M* and *N* are the lengths of the two sequences being compared. For each cell *i,j* in the distance table, calculate the cost of arriving from below, left, and below-left, by adding the cell value to the minimum distance path up to the cell below, left, or below-left, respectively and then appropriately adjust for penalty or discount. The minimum of these three terms gives the minimum distance path ($D_{i,j}$) up to

the cell *i,j* from the start. Figure 4 shows the relation. Penalties can be appropriately adjusted depending on the distance function.

$D_{i,j} = \text{Min} [D_{i-1,j-1} , D_{i-1,j} + W_1, D_{i,j-1} + W_2] + d_{i,j}$ i>1 and j>1
$D_{1,1} = d_{1,1}$
$D_{i,1} = D_{i-1,1} + W1 + d_{i,1}$ i>1
$D_{1,j} = D_{1,j-1} + W2 + d_{1,j}$ j>1
where,
W_1 and W_2 are constant penalties

Figure 4: DTW Distance Table Construction

2.2 Dynamic Programming for DNA Sequence Alignment

Global alignment of two biological sequences can be calculated by Dynamic Programming. Algorithms have been designed to detect total correspondence, overlaps and containments amongst homologous sequences of DNA fragments, allowing for general substitution. Homologous sequences of similar lengths may reveal a total correspondence relationship if their protein molecules underwent only insignificant insertions and deletions. A containment relationship may exist between homologous sequences of significant different lengths if the molecule of the shorter sequence underwent major deletions or that of the longer sequence underwent major insertion. Homologous sequences may bear an overlapping relationship if significant insertions and deletions occurred at the termini of their molecules. Gotoh [GOT82] presented an algorithm to match two sequences that allows multiple-sized gaps (similar to Waterman et al.) and runs in MN steps where M and N are the lengths of the sequences being compared. Figure 5 shows the algorithm.

Given two sequences $A=a_1a_2...a_M$ and $B=b_1b_2...b_N$, the algorithm computes the smallest-scoring alignment of *A* and *B*. This alignment is called the optimal alignment. The score of the optimal alignment of *A* and *B* is $D_{M,N}$, which is referred to as the similarity score of *A* and *B*. This score can be computed in linear space. The algorithm penalizes each internal gap of length *k* by $w_k = uk + v$ $(u \geq 0, v \geq 0)$.

Algorithm

Let $A=a_1a_2\ldots a_M$ and $B=b_1b_2\ldots b_N$ be the two sequences to be compared, i.e. we want to find the best match score between the two sequences. A weight $d(a_i,b_j)$ is given to an aligned pair of residues a_i and b_j. $d(a_i,b_j) = 0$ if $a_i=b_j$, and $d(a_i,b_j)>0$. Waterman et al. generates a distance matrix $D_{i,j}$ by induction as follows:

$$D_{i,j} = \text{Min} [D_{i-1,j-1} + d(a_i,b_j) , P_{i,j} , Q_{i,j}],$$

Where,

$$P_{i,j} = \min_{1\le k\le i} [D_{i-k,j} + w_k]$$

$$Q_{i,j} = \min_{1\le k\le j} [D_{i,j-k} + w_k]$$

$w_k = uk + v$ is the linear gap penalty for a gap of length k
u (≥0) and v (≥0) are constants

$P_{i,j}$ and $Q_{i,j}$ can be obtained in a single step according to the following recursion relations:

$$P_{i,j} = \text{Min} [D_{i-1,j} + w_1 , \min_{2\le k\le i} (D_{i-k,j} + w_k)]$$

$$= \text{Min} [D_{i-1,j} + w_1 , \min_{1\le k\le i-1} (D_{i-1-k,j} + w_{k+1})]$$

$$= \text{Min} [D_{i-1,j} + w_1 , \min_{1\le k\le m-1} (D_{i-1-k,j} + w_k) + u]$$

$$= \text{Min} [D_{i-1,j} + w_1 , P_{i-1,j} + u]$$

and

$$Q_{i,j} = \min [D_{i,j-1} + w_1 , Q_{i,j-1} + u]$$

Here, $w_1 = u + v$

Figure 5: Algorithm for matching biological sequences

The algorithm iterates MN times, where each iteration consists of choosing the smallest of three terms for $D_{i,j}$ and the smaller of two terms for each of $P_{i,j}$ and $Q_{i,j}$. Calculation of $P_{i,j}$ and $Q_{i,j}$ is reduced to constant time by recursion as shown in the algorithm above. Thus, the algorithm essentially runs in $O(MN)$. At the beginning of the iterations, one may set $D_{i,0} = P_{i,0} = w_i$ $(1\le i\le M)$, and $D_{0,j} = Q_{0,j} = w_j$ $(1\le j\le N)$.

This algorithm differs from the DTW algorithm in one aspect. This algorithm allows an arbitrary number of gaps in the minimum distance path, with a penalty linearly proportional to the length of the gap. The $P_{i,j}$ term in the calculation $D_{i,j}$ determines best gaps in sequence along the horizontal axis and the term $Q_{i,j}$ determines gaps in the other sequence. In addition, the first element of one sequence need not match the first element of the other.

In a computer program, not all the elements of $D_{i,j}$, $P_{i,j}$ and $Q_{i,j}$ need be memorized; two one-dimensional arrays and one variable are sufficient to store temporary values of these quantities.

2.3 Dynamic Programming for Shape Recognition

Pattern matching for leaf-shape recognition should obey following two rules:

1) It should be invariant to translation, rotation and scaling of the shapes.
2) It should be able to handle deformities, occlusions and overlaps.

The first rule is required of any shape matching technique. The second condition is a special rule necessary for the digital images obtained from the herbarium. No two leaves of same species have exactly the same shape. Some leaves might be deformed, folded or even overlapping. These distortions have to be handled.

The Dynamic Time Warping (DTW) algorithm is a good general approach for matching similar signals. It can be used to compare objects with stored templates. However, a good one-dimensional representation for two-dimensional shapes is required. Scalar transformation of Cartesian representation of the boundary into a one-dimensional signal generally filters out the location information. Rotation of the original shape results in a cyclic shift in the one-dimensional representation.

Minor modifications, as shown in figure 6, to DTW algorithm gives a dynamic program that obeys the first rule. This change allows the minimum distance path to wrap-around one side of the distance table to the other, thus accommodating for any cyclic shifts in the signals being matched. For calculation of the $D_{1,j}$ term, $D_{M,j}$ is ignored for efficient implementation. Omission of this term does not reduce the effectiveness of the algorithm, as wrap-around in the final path will not occur more than once for each leaf.

$$D_{i,j} = \text{Min}\left[D_{i-1,j-1} , D_{i-1,j} + \text{Penalty}, D_{i,j-1} + \text{Penalty} \right] + d_{i,j}$$
$$D_{1,1} = d_{1,1}$$
$$D_{i,1} = D_{i-1,1} + \text{Penalty} + d_{i,1} \qquad i>1$$
$$D_{1,j} = \text{Min}\left[D_{M,j-1} , D_{1,j-1} + \text{Penalty} \right] + d_{1,j} \qquad j>1$$

Figure 6: Modification to DTW

If the two shapes are of different scale, the terms $D_{i-1,j}$ and $D_{i,j-1}$ in calculation of $D_{i,j}$ will allow a single point of one stream to match with multiple points on the other. This strategy also works on local dissimilarities.

Ideas from DNA sequence matching algorithm, which was described earlier, can be borrowed to allow multiple gaps and occlusions. A dynamic programming algorithm for matching whole, partial and overlapped shapes is shown in figure 7. The term $Q_{i,k}$ determines gaps in one of the sequences. For each gap, a penalty w_k proportional to the

length of the gap is added to the minimum distance path. The term $S_{i,j}$, which is split into terms $P_{i,j}$ and $R_{i,j}$, allows overlaps, i.e. allows the same part of the second pattern to match multiple parts of the first pattern.

Algorithm

Let $A=a_1a_2\ldots a_M$ and $B=b_1b_2\ldots b_N$ be the two sequences to be compared

$$D_{i,j} = \text{Min}\,[\, D_{i-1,j-1} , D_{i-1,j} + W_1 , D_{i,j-1} + W_2 , S_{i,j} , Q_{i,j} \,] + d(a_i,b_j)$$

Where,
W_1 and W_2 are constant penalties,

$$Q_{i,j} = \min_{1\le k\le j} [\, D_{i,k} + w_k \,] \quad \text{and}$$

$$S_{i,j} = \min_{1\le k\le M} [\, D_{k,j-1} \,] + W_3$$

$w_k = W_4k$ $(u\ge 0, v\ge 0)$ is the linear gap penalty for a gap of length k

W_3 and W_4 are a constants

$S_{i,j}$ can be split into two terms $P_{i,j}$ and $R_{i,j}$

Where,

$$P_{i,j} = \min_{1\le k\le i} [\, D_{k,j-1} \,] + W_3$$

$$R_{i,j} = \min_{i+1\le k\le M} [\, D_{k,j-1} \,] + W_3 \quad \text{and}$$

$$S_{i,j} = \text{Min}\,[\, P_{i,j} , R_{i,j} \,]$$

Thus,
$$D_{i,j} = \text{Min}\,[\, D_{i-1,j-1} , D_{i-1,j} + \text{Penalty1} , D_{i,j-1} + \text{Penalty1} , P_{i,j} , Q_{i,j} , R_{i,j}] + d(a_i,b_j)$$

The cost of the minimum distance path is the minimum of all $D_{i,N}$

Figure 7: Dynamic Program for shape recognition

This algorithm just like the earlier algorithms iterates MN times. In each iteration, a minimum of five terms, viz. $D_{i-1,\,j-1}$, $D_{i-1,j} + W_1$, $D_{i,j-1} + W_2$, $S_{i,j}$ and $Q_{i,j}$, must be calculated. By an inductive process, $S_{i,\,j}$ (i.e. $P_{i,\,j}$ and $R_{i,\,j}$) and $Q_{i,j}$ can be calculated in single step as shown in figure 8. Thus, each iteration executes in constant time, and the algorithm runs in *O(MN)*.

$$Q_{i,j} = \text{Min}\left[D_{i-1,j} + w_1 , \underset{2\le k\le i}{\text{Min}} (D_{i-k,j} + w_k) \right]$$

$$= \text{Min}\left[D_{i-1,j} + w_1 , \underset{1\le k\le i-1}{\text{Min}} (D_{i-1-k,j} + w_{k+1}) \right]$$

$$= \text{Min}\left[D_{i-1,j} + w_1 , \underset{1\le k\le m-1}{\text{Min}} (D_{i-1-k,j} + w_k) + W_4 \right]$$

$$= \text{Min}\left[D_{i-1,j} + w_1 , Q_{i-1,j} + W_4 \right]$$

$$S_{i,j} = \text{Min}\left[P_{i,j} , R_{i,j} \right]$$

$$P_{i,j} = \underset{1\le k\le i}{\text{Min}} \left[D_{k,j-1} \right] + W_3$$

$$= \text{Min}\left[D_{i,j-1} , \underset{1\le k\le i-1}{\text{Min}} (D_{k,j-1}) \right]$$

$$= \text{Min}\left[D_{i,j-1} , P_{i-1,j} \right]$$

Similarly,

$$R_{i,j} = \text{Min}\left[D_{i,j-1} , R_{i-1,j} \right]$$

Figure 8: Inductive calculation of Si,j and Qi,j

Figure 9 illustrates the significance of such an algorithm. It shows matching of an isolated whole leaf template with two unknown overlapping leaves. Note the unknown pattern is along the horizontal and the known pattern is along the vertical. Initially, the minimum distance path consists of cells corresponding to matching elements of the isolated leaf pattern with the left side leaf of the two overlapped leaves. The first three terms in calculation of $D_{i,j}$ determine this path. Then there is a jump (made possible by terms $P_{i,j}$ and $R_{i,j}$) in the path, and the isolated leaf pattern starts matching the boundary of right side leaf. The figure also illustrates the wrap around. Finally, the overlapped area of the two leaves does not match anything on the isolated leaf and hence results in a gap. The gaps can be attributed to term $Q_{i,j}$ in calculation of $D_{i,j}$. The minimum distance path starts from the first column and terminates at the last column. The cost of the minimum distance path is the minimum of all $D_{i,N}$ values in the last column.

W_1 and W_2 should be set to small values. A small value larger than zero makes diagonal moves preferable. Constant W_3 determines the threshold for gap inclusion. W_4 assures that there are no arbitrary jumps in an attempt to match the same part of known sequence with different parts of the unknown sequence.

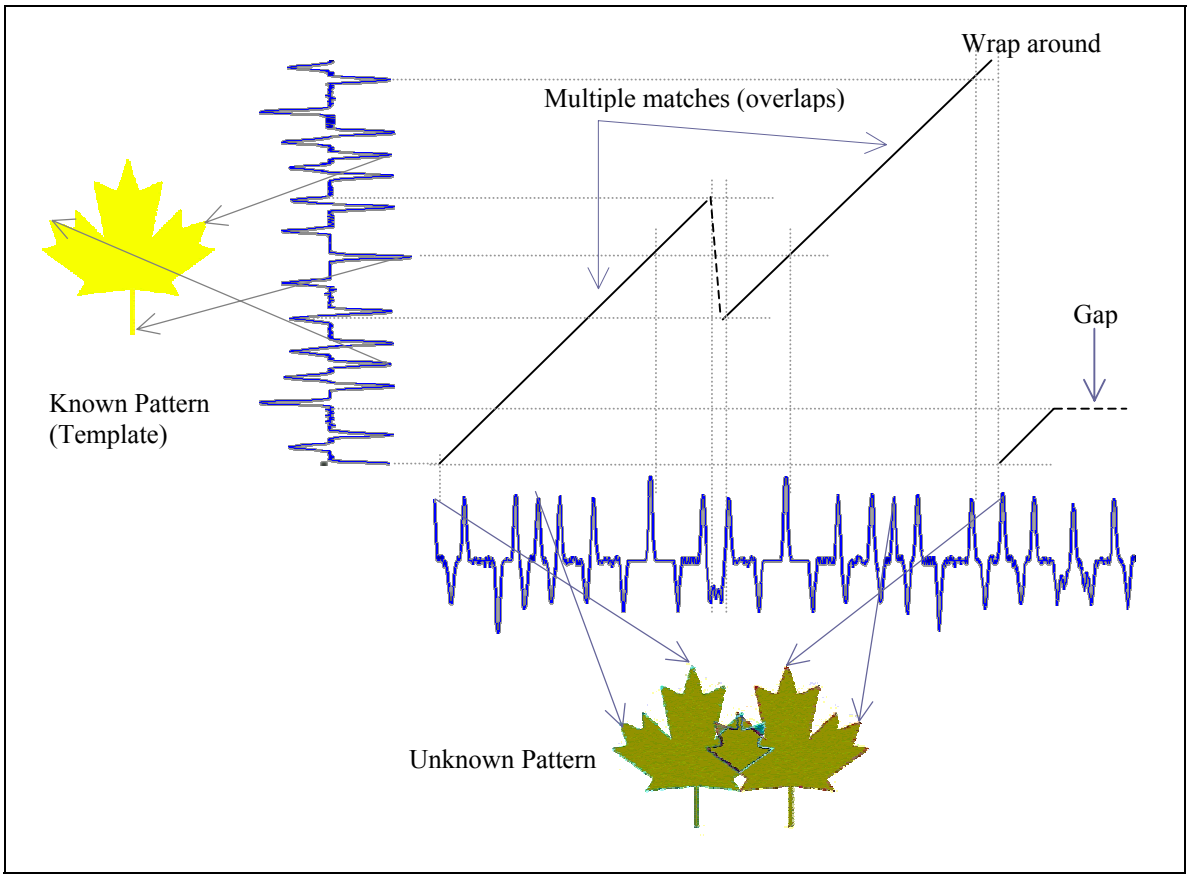


Figure 9: Minimum distance path for matching overlapped leaves to an isolated leaf

3 Plant Species Identification

The last section described the dynamic programming algorithms for pattern matching. This section describes an implementation for plant species identification.

3.1 The Whole Process

There are two modules in the implementation. These modules run in batch mode. The first one performs basal image processing. It extracts features that describe the shape of the leaves. The second module embeds the dynamic program and performs the stint of template matching and species prediction. The image processing module and the pattern-matching algorithm were implemented to work on digitized image samples as shown in figures 10 and 11. Figure 10 shows complete and isolated leaves. Figure 11 shows digitized plant samples from an herbarium.

The following configurations have been considered:

1) Training and testing the system with only isolated leaves
2) Training with isolated leaves and predicting species of herbarium samples
3) Training with herbarium samples and predicting species of isolated leaves

Figure 10: Digitized isolated leaf pictures

Figure 11: Digitized herbarium images

The features extracted by the first module are the angles measured at each pixel point along the boundary of the leaves. The actual procedure for extracting these angles is described later. The angles recorded along a boundary form a sequence, which is input to the dynamic program. The output of the dynamic program is the cost of the best path match between two streams corresponding to the two leaf-shapes being compared. To predict the species of an unknown leaf, the leaf shape is compared to all template leaves using the dynamic programming algorithm. The *K* best matches are selected. Each of the *K* nearest neighbors votes in favor of its (annotated) species, and the species that gets the most votes is predicted to be the species of the input (unknown) leaf shape.

3.2 Image Processing

Color segmentation is performed to separate out the leaf pixels from the background pixels. The scanned pictures of the leaves are not void of dust, seeds, and other particles. Filtering is required to remove this unwanted noise. A morphological operator called erosion [BAR95] was applied to remove pixels corresponding to such noise. The images are converted to a matrix of binary values 0 and 1 or *ON* and *OFF*. *ON* corresponds to pixels belonging to the leaf area, and the rest of the cells in the matrix are *OFF* (see figure 12). For each island of *ON* pixels, its boundary is extracted.

Figure 12: Image processing for isolated leaf

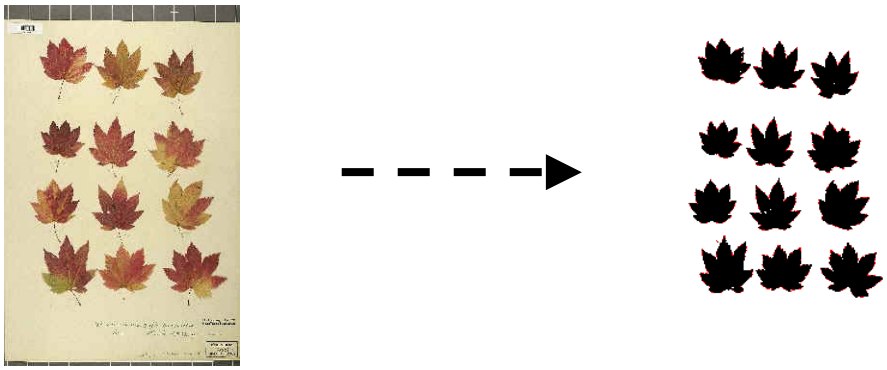


Figure 13: Image processing for Herbarium sample image

The procedure for navigating the boundary starts by scanning from pixel (0,0) – left to right, bottom to top – until it finds the first leaf-pixel, i.e. a pixel that is *ON*. It extracts all

boundaries in the anti-clockwise direction. To do this, at each pixel it keeps track of the direction followed to reach that pixel. The direction at the start pixel is *RIGHT* (At each pixel, one of four directions is possible – *RIGHT, LEFT, UP* and *DOWN*). Depending on the current direction, a search is done for the next boundary pixel in the counter-clockwise direction. The trace ends when we return to the start pixel. Each closed boundary that is extracted from the image is called an island.

The island boundaries that are significantly smaller compared to the size of the whole image are rejected. Specifically, a boundary of size less than one-tenth the height of the entire images may be considered insignificant.

```
Search in raster order for first pixel that is ON
Let it be P[i][j]

Direction=RIGHT

Switch (direction)

Case RIGHT:   if (P[i+1][j-1]=ON) {direction=DOWN; P[i+1][j-1] is the next pixel}
              else if (P[i+1][j]=OFF) {direction=UP; still at the same pixel}
              else P[i+1][j] is the next pixel

Case LEFT:    if (P[i-1][j+1]=ON) {direction=UP; P[i-1][j+1] is the next pixel }
              else if (P[I-1][j]=OFF) {direction=DOWN; still at the same pixel }
              else P[i-1][j] is the next pixel

Case UP:      if (P[i+1][j+1]=ON) {direction=RIGHT; P[i+1][j+1] is the next pixel }
              else if (P[i][j+1]=OFF) {direction=LEFT; still at the same pixel }
              else P[i][j+1] is the next pixel

Case DOWN:    if (P[i-1][j-1]=ON) {direction=LEFT; P[i-1][j-1] is the next pixel }
              else if (P[i][j-1]=OFF) {direction=RIGHT; still at the same pixel }
              else P[i][j-1] is the next pixel

Record the next pixel coordinates

Repeat the above case until back to the start pixel
```

Figure 14: Boundary extraction algorithm

The pseudo code of figure 14 outputs the coordinates of the pixels along the boundary of the leaves. The coordinates are converted into a stream of angles before applying the dynamic programming algorithm. Figure 15 shows how this conversion is done. At each pixel, we construct a polygon approximation of the local curvature and measure the exterior angle *A* as shown in figure 15. To obtain the exterior angle at pixel *i* along the boundary, we compute the angle between the line segment joining pixel *i* to pixel *i-10* and the line segment joining pixel *i* to pixel *i+10*. The sequence of angles measured along

the boundary in counter clockwise direction gives a one-dimensional stream representation of the boundary.

Figure 15: *Angle* measurements along the boundary

3.3 Image Retrieval

The dynamic program discussed in section 2.3 compares the stream of angles obtained for given input leaf against all of the templates in the library i.e. the training set. The cost of the minimum distance path for each match is calculated. Finally, the *K* best matches, i.e. the templates that give minimum cost are picked out. A majority of the retrieved images is expected to belong to the same species as the unknown leaf. The retrieved images vote in favor of their (annotated) species, and the species that gets the most votes is predicted to be the species of the unknown leaf shape.

For an input image of an isolated leaf, predicting its class is easy. After retrieving the *K* best matches, the species with the largest number of images retrieved from database is predicted as the species of the unknown leaf. An herbarium sample image after image processing may produce more than one island. In this case, each boundary is used to retrieve *K* matches, and the species whose highest number of images is retrieved from the database is predicted as the class of the given input image.

If the only requirement is to predict the species of an unknown leaf image, then the template images need not be stored in the database. Only the boundary description and the annotations are required to be stored in the database. If the aim is to build a content-based image retrieval system, then the images can be preprocessed and the shape description can be stored along with the images for faster matching and retrieval.

The next section summarizes the performance of the implementation. The percentage accuracy in predicting the species of the unknown leaves is one way to evaluate performance. The precision-recall curves (section 4.2) show how relevant the retrieved images are to the querying image. The performance has been evaluated using six different species of leaves belonging to two genera, Acer (Maples) and Quercus (Oaks).

4 Results

Figure 16 shows the genus Acer (maples). The botanical names and the common names are shown in the figure. Figure 17 shows two leaves belonging to genus Quercus (oaks).

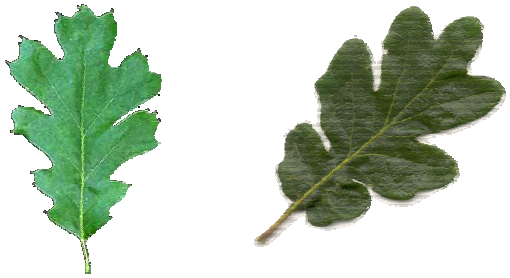
The performance of the algorithms has been evaluated for following cases:

1. Training and testing the system with only isolated leaves
2. Training with isolated leaves and predicting species of herbarium samples
3. Training with herbarium sample and predicting species of isolated leaves

The performance can be evaluated in terms of percentage accuracy in predicting the species of the unknown leaf shapes. The percentage of input images correctly labeled gives a good estimate of how well the algorithms perform. Precision-recall curves show how relevant the retrieved images are to the querying image.

Acer Circinatum (Vine Maple) | Acer Glabrum (Douglas Maple) | Acer Macrophyllum (Big-leaf Maple) | Acer Negundo (Boxelder)

Figure 16: Genus *Acer* (maple)



Quercus Kelloggii (California Black Oak) | Quercu Garryan (Oregon White Oak)

Figure 17: Genus *Quercus* (oak)

4.1 Species Prediction Accuracy

Table 1 shows the percentage accuracy obtained in predicting the species of isolated leaves. The template database for this result consisted only of isolated leaves of the six species mentioned before. The training set (i.e. the template database) was constructed with twenty isolated leaves of each species. The overall accuracy obtained for isolated leaves is 96.8%.

Species	Total leaves	Classified as						Correctly classified	Accuracy
		AC	AG	AM	AN	QG	QK		
Acer Circinatum (AC)	40	40	0	0	0	0	0	40	100%
Acer Glabrum (AG)	40	0	38	2	0	0	0	38	95%
Acer Macrophyllum (AM)	40	2	0	35	0	3	0	35	88%
Acer Negundo (AN)	18	1	0	0	17	0	0	17	94%
Quercus Garryana (QG)	40	1	0	0	0	39	0	39	98%
Quercus Kelloggii (QK)	40	0	0	0	0	0	40	40	100%

Table 1: Predicting species of isolated leaves

Table 2 shows the percentage accuracy for predicting plant species of samples from the herbarium. The training set for this case is same as the one used above. The test set or the unknown image set was constructed with processed digitized samples from the herbarium. The over all accuracy in this case is 59%.

Species	Total leaves	Classified as						Correctly classified	Accuracy
		AC	AG	AM	AN	QG	QK		
Acer Circinatum (AC)	30	25	3	2	0	0	0	25	83%
Acer Glabrum (AG)	30	1	20	4	1	4	0	20	67%
Acer Macrophyllum (AM)	30	0	16	14	0	0	0	14	47%
Acer Negundo (AN)	8	1	4	1	2	0	0	2	25%
Quercus Garryana (QG)	30	4	2	4	0	19	1	19	63%
Quercus Kelloggii (QK)	20	2	3	4	0	3	8	8	40%

Table 2: Predicting species of digitized herbarium sample images

Table 3 shows the performance accuracy for predicting plant species of isolated leaves. However, in this case, the training set consists of leaf shapes extracted from the digitized herbarium samples. The overall accuracy in this case is 61%.

Species	Total leaves	Classified as						Correctly classified	Accuracy
		AC	AG	AM	AN	QG	QK		
Acer Circinatum (AC)	40	34	1	1	0	4	0	34	85%
Acer Glabrum (AG)	40	7	33	0	0	0	0	33	83%
Acer Macrophyllum (AM)	40	18	7	9	0	5	1	9	23%
Acer Negundo (AN)	18	15	1	2	0	0	0	0	0%
Quercus Garryana (QG)	40	3	2	0	0	32	3	32	80%
Quercus Kelloggii (QK)	40	3	0	11	0	0	26	26	65%

Table 3: Predicting species of isolated leaves with training templates consisting of only herbarium sample images

(a) (b) (c)

Figure 18: (a) Isolated leaf, (b) Incorrectly classified sample, (c) Correctly classified sample

The predictions made in tables 1 and 2 are results of retrieving the nine best matches and then having these nine isolated leaves to vote for the species. The results in table 3 are obtained by retrieving 20 best matches. In this case, a larger number (20) was used because the training set consists of many more boundaries (as compared to 120 isolated leaves in case of tables 1 and 2) and includes non-leaf shaped boundaries. Therefore, it is preferable to retrieve more matching shapes and then having them to vote. The accuracy in predicting species *Acer Negundo* is low because of lack of good quality herbarium samples for this species. Figure 18 shows examples of correct and incorrect classification decisions made by the method. It also gives a sense of how difficult the classification task is.

4.2 Precision-Recall Plots

The standard measure of performance for information retrieval systems is the precision-recall plot. Consider a query to an information retrieval system (in this case, an image of an isolated leaf). We can view the information retrieval system as computing a ranking of all the documents (i.e. herbarium samples) in the database and returning the top K most relevant documents. In our application, the user wants the most relevant documents to be the ones from the same species as the query. The "precision" of the retrieval is the percentage of the K documents that belong to the correct species. The "recall" of the retrieval is the percentage of all documents for the correct species that are included in the top K retrieved documents. There is always a precision-recall tradeoff: If the information retrieval system returns the entire set of documents, then recall will be 100%, but precision will be very low. If the system returns just one document from the correct class, then precision will be 100%, but recall will be very low. The tradeoff can be visualized by plotting precision and recall as K is increased from one to some maximum value.

Figure 18 shows the precision-recall plot for the isolated leaf classification of the six plant species individually. Figure 19 shows the curves for all species combined and the curves for all species belonging to genus *Acer* and *Quercus*. The template database consists of 20 isolated leaves of each species. For small values of recalls (i.e. small setting of K), precision is over 90%. Precision gradually decreases as K is increased.

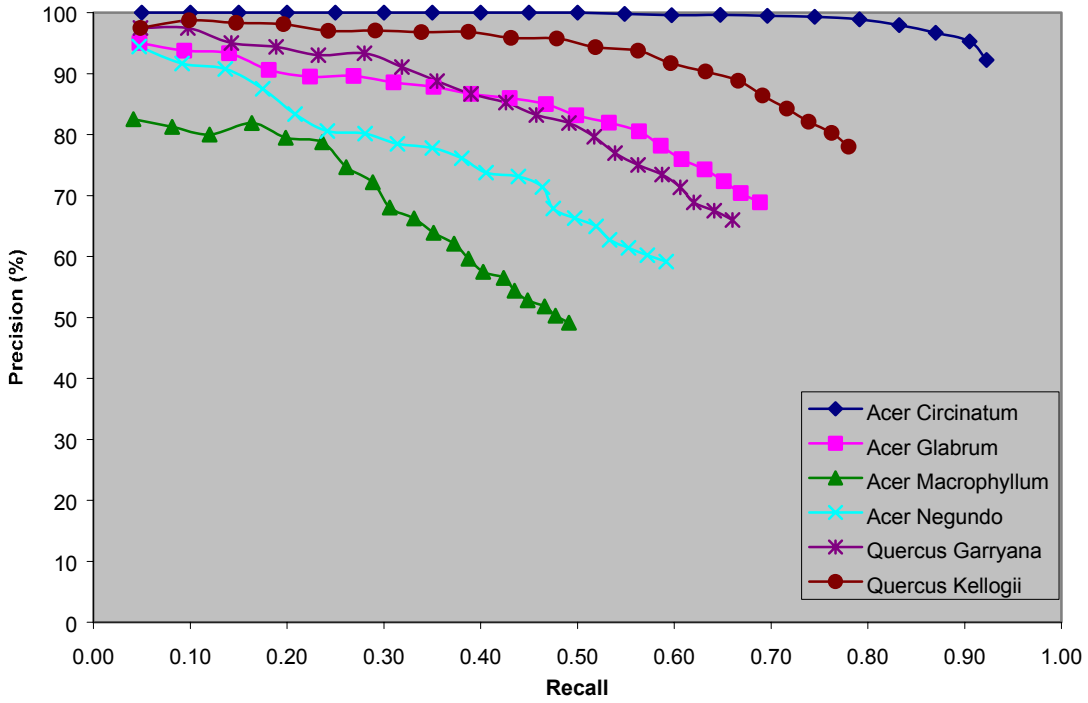


Figure 18: Precision-recall curves for isolated leaves

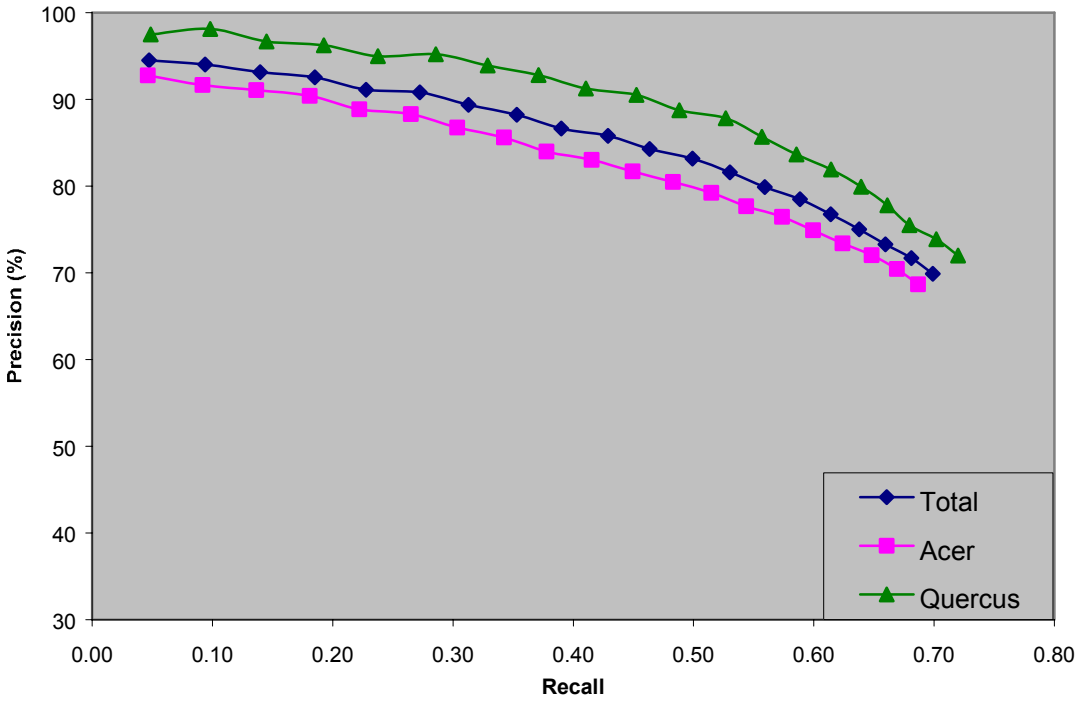


Figure 19: Precision-recall plot for the two genera and the overall performance

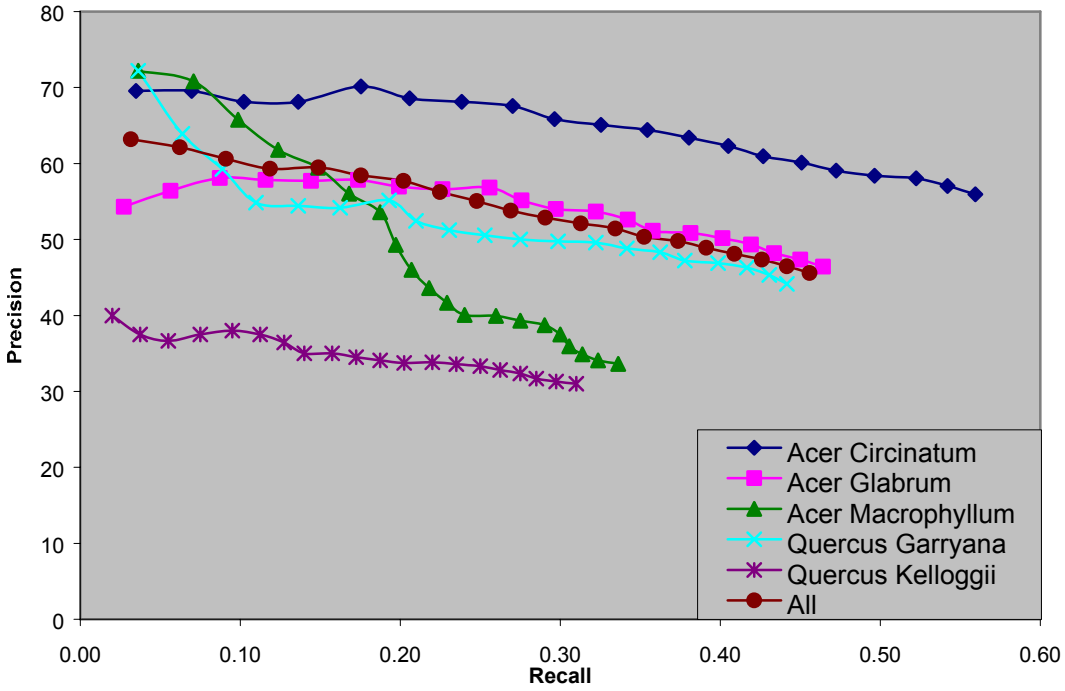


Figure 20: Precision-recall for retrieval of isolated leaves with herbarium samples as query

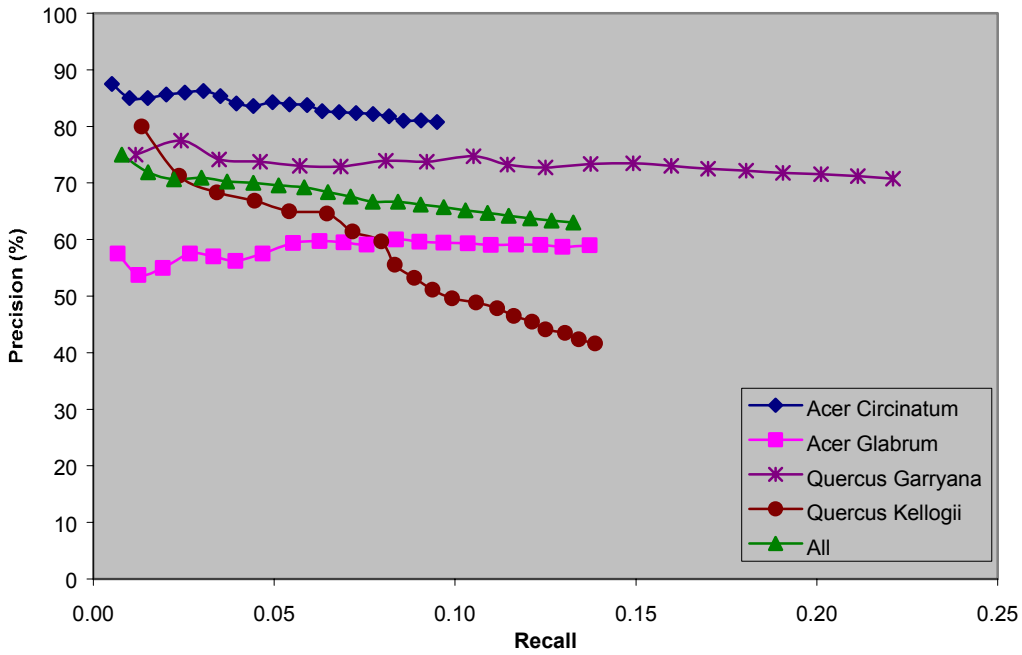


Figure 21: Precision-recall for retrieval of herbarium samples with isolated leaves as queries

Figure 20 shows precision in retrieving isolated leaves when herbarium samples are used for querying. *Acer Negundo* has been skipped in the plots because of inadequate good quality herbarium samples. Figure 21 shows the precision in retrieving herbarium samples from the database with isolated leaves used as query images. Note that the precision has been calculated only for small values of recall since the database consists of a huge collection of unfiltered boundaries from the herbarium samples. That is, the boundaries also include stems, flowers and other unwanted boundaries.

4.3 Performance and Penalties

Four penalties or parameters need to be set in the dynamic programming algorithm given in figure 7 of section 2.3. These penalties are W_1, W_2, W_3 and W_4. W_1 and W_2 are the penalties for non-diagonal moves in the minimum distance path. These penalties should be small and yet exist just to control non-diagonal moves. Generally, these penalties should be set to less than $1/10^{th}$ the maximum of the distances (i.e. maximum of $d_{i,j}$) in the distance table. In the experiments presented in this section $max(d_{i,j})$ is 360 (the difference between maximum angle value that can be measured along the boundary; which is 180 and –180 degrees), and W_1 and W_2 are set to 10.

The penalty W_4 determines how easily a gap can be inserted in the boundary being matched. A part of the boundary can be skipped instead of being matched to an arbitrary part of the other leaf. For example, some of the leaf boundaries might include a stalk while the others could be missing it. In this case, instead of matching the stalk to an arbitrary part of the other leaf, a gap can be inserted with a total penalty proportional to the length of the gap inserted. In the algorithm of figure 4, the gap penalty is W_4 times the gap length. This penalty should be larger than the maximum value of the distance that is considered a match between two feature vectors. Again, for feature vectors that consist of angles, a value of about 90 is suitable. The penalty W_3 allows multiple sections of one shape to match same part of the other shape. A sufficiently large value for this penalty is required to avoid situations where three lobes of *Acer Glabrum* are matched twice with six lobes of *Acer Circinatum*. On the other hand it should not be too large as to prohibit any reasonable attempts to match a single leaf to multiple overlapped leaves. For the implementation shown here, a value in the range of 250 to 500 performs well.

The dynamic program used in the implementation is robust to small changes in the values of the penalties. However, prohibiting gaps and multiple sections matching by setting W_3 and W_4 to very high values degrades the performance on the herbarium samples. With the penalties set to the values given above, the accuracy in predicting the species of herbarium sample is 59%. Setting W_3 and W_4 to very high values reduces this accuracy to 55%.

5 Conclusion and Future Work

The algorithms perform well on the isolated leaves as shown in the result section. Precision in retrieving isolated leaves from a library of isolated leaves is over 90%. Working with the digitized herbarium is a little tricky. On an average, the best overall accuracy obtained in predicting species of leaves on herbarium samples is 59%. In the current implementation, stems, flowers, seeds, etc. are not filtered out. These objects are assumed to be leaf shapes and often vote against the correct prediction. Predicting species of isolated leaves with a training set constructed from herbarium samples yields 61% accuracy.

The accuracy when working with herbarium samples can be improved if we can separate the boundaries corresponding to non-leaf (stem, flowers, seeds, etc.) shapes and the boundaries that are indistinct due to heavy mutation or excessive snarling of leaves. One way to perform such filtering is to hand label all or some non-leaf objects. If only some stems and flowers are hand labeled, then these hand labeled data can be used to identify stems, flowers, etc. in the remaining samples, and they can be removed from the training set or excluded from the test set.

There is no concrete evidence to show that there is a definite advantage of using penalties W_3 and W_4. However, some experiments do suggest that there is a performance improvement on using these. A controlled experiment is required to tune these parameters and find out how much benefit is obtained by using these penalties. Of course, using these does not increase run time of the algorithm beyond a constant fraction.

References

[BAR95] *Image Analysis for the Biological Sciences*, C. A. Glasbey and G. H. Horgan, John Wiley & Sons, 1995

[BEL57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[CC01] *Shape Analysis and Classification, Theory and Practice*, by Luciano da Fontoura Costa and Roberto Marcondes Cesar Jr., 2001

[CD86] R. Chin and C.R. Dyer, *Model-Based Recognition in Robot Vision*. Computing Surveys; Vol 18, No 1, 1986.

[GOT82] O. Gotoh. *An improved algorithm for matching biological sequences*. Journal of Molecular Biology, 162, 705-708, 1982.

[GMK88] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, *Partial Shape Recognition using Dynamic Programming*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 10, No., 2, March 1988.

[GR97] V. N. Gudivala and V. V. Raghavan. *Content-based image retrieval systems*. IEEE Computers on Computer Vision and Pattern Recognition, pages 678-683, 1997.

[KL83] J. B. Kruskall & M. Liberman. *The symmetric time warping algorithm: From continuous to discrete*. In Time Warps, String Edits and Macromolecules: The Theory and Practice of String Comparison. Addison-Wesley, 1983.

[LON98] Sven Loncaric. *A Survey of Shape Analysis Techniques*. Pattern Recognition, 31(8):983--1001, 1998.

[MKH77] M. K. Hu. *Visual pattern recognition by moment invariants*. In Computer Methods in Image Analysis. Los Angeles: IEEE Computer Society, 1977.

[NW70] S. B. Needleman and C. D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequences of two proteins*. Journal of Molecular Biology, 48, 443-453, 1970.

[PDM02] E. G. M. Petrakis, A. Diplaros, E. Milios, *Matching and Retrieval of Distorted and Occluded Shapes using Dynamic Programming*. IEEE Transactions on Patterm Analysis and Machine Intelligence, accepted, March 2002.

[PF94] E. Persoon and K. S. Fu. *Shape discrimination using fourier descriptors*. IEEE Transaction on Systems, Man and Cybernetics, 7(3): 170-179, March 1977.

[PK00] *Plants of the Pacific Northwest Coast: Washington, Oregon, British Columbia, and Alaska* by Jim Pojar, Andy MacKinnon (Editor), Pojar Mackinnon 2000.

[RHC99] Yong Rui, Thomas S Huang, and Shih-Fu Chang, *Image retrieval: Current techniques, promising directions and open issues*. Journal of Visual Communications and Image Representation, 10(1):1-23, March 1999.

[RP98] V. Rehrmann and L. Priese. *Fast and robust segmentation of natural color scenes*. In Proceedings of the 3 rd Asian Conference on Computer Vision, volume 1, pages 598--606, HongKong, January 1998.

[SC95] J. R. Smith and S. F. Chang. *Single Color Extraction and Image Query*. In Proceedings, IEEE International Conference on Image Processing (ICIP-95), Washington, DC, October 1995.

[SEL74] P. H. Sellers, *On the theory and computation of evolutionary distances*, SIAM. Journal of Applied Mathematics, 26, 787-793, 1974

[WSB76] M. S. Waterman, T. F. Smith, and W. A. Beyer *Some bioloical sequence metrics*. Advanced Mathematics, 20, 367-387, 1976.