

# **Fourier Descriptors**

Properties and Utility in Leaf Classification

ECE 533 Fall 2006

Tyler Karrels

## **Introduction**

Now that large-scale data storage is feasible due to the large capacity and low cost of hard drives, huge image databases are becoming more prevalent. We need to be able to search these databases with a textual description of the image we desire in order for a large database to have any use. Manually entering this searchable information (meta-data) is tedious and impractical when the number of images is large. One way to quickly extract and assign information contained in images is using Fourier descriptors to recognize shapes.

Fourier descriptors are a classical method to shape recognition and they have grown into a general method to encode various shape signatures. Previous experiments have used Fourier descriptors to recognize different types of marine life, product deformations, and tree leaves. I chose to implement a tree leaf identification program using Matlab because of a personal interest in nature and a database of leaf images is easy to create. There are many unique leaf shapes such as the oak tree leaf and the maple tree leaf. These leaves are easily recognized by any one who has grown up around these trees. But there are also more subtle differences between leaf shapes. A red maple leaf has notched lobes; a sugar maple leaf has smooth lobes. Because a good method for shape recognition needs to detect these subtleties, leaf shapes are a great example to test the limits of Fourier descriptor methods.

The general procedure begins with a color image containing a shape that we want to recognize. The color image is converted to a grayscale image. Then a threshold is applied to the grayscale image converting it into a black and white (ie. binary image). The threshold is applied so that the shape is enhanced and can easily be found. The shape

is located and its boundary is extracted. A shape signature is then used to describe the boundary. Several shape signatures exist such as the canonical complex boundary sequence or the centroid contour distance curve, which I apply here.

## Approach

The database images are pictures of five different species of tree leaves and one type of shrub. Each leaf was placed on a white sheet of paper and labeled with the leaf type and a number for identification purposes. Over 200 pictures were taken of 59 different leaf samples in different orientations and places in the image. Once a database of leaf images was acquired, work began to extract the leaf boundary.

Before any processing takes place the image size is reduced from 3 Mega-pixels to 0.5 Mega-pixel. This speeds up all calculations by eliminating unnecessary precision in the image. The color image is then converted to a grayscale image. An appropriate grayscale threshold is obtained by locating the largest valley in the grayscale histogram. Due to the controlled environment during image acquisition, a single threshold clearly distinguishes the leaf and stem from the background in most cases.

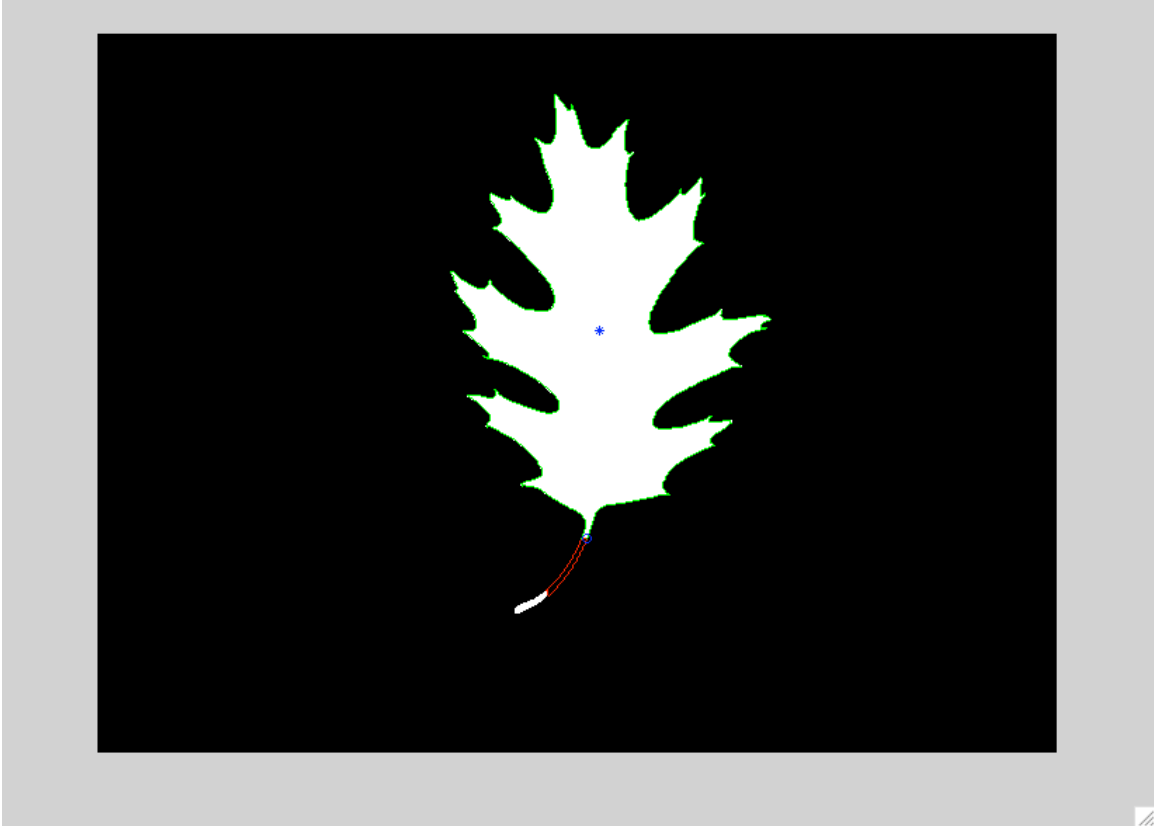
The resulting binary image has a black background and the leaf and stem appear white. The image is then dilated by a small (3 pixel square) structuring element to remove any minor tears in the leaf boundary. Then the six longest boundaries are



displayed, and a human must select which one is the leaf. This human interaction makes working with a database of just over 200 images tedious and time consuming. A final solution must address this problem. After the leaf is identified, it is isolated by placing it in the center of a new zero-padded (black) image. Any internal holes in the leaf are then filled.

To achieve a rotation invariant description, the stem is always used as the starting point. To find the stem, the leaf is eroded (10 pixel square structuring element) to remove the stem. The resulting leaf body is dilated twice (15 pixel square structuring element) so the body grows larger than it was before erosion. This “window” is subtracted from the original leaf yielding only the stem. The stem boundary is the longest boundary remaining. The stem is next subtracted from the original leaf image to yield just the leaf. The leaf boundary is the longest boundary remaining. Precision is preserved since the result has not been eroded or dilated (except initially to remove tears).

To find the starting point on the leaf boundary we find the closest point on the stem boundary to the leaf. To do this, the centroid of the leaf is calculated and then the closest point to it on the stem boundary is found. This point is the closest to the body of the leaf. Then the point on the leaf boundary that is closest to the stem point is found. This point is the starting point of the centroid contour distance curve. This is a quick and computationally efficient method to find a reliable starting point.



The leaf outline is green, stem outline red, and the centroid and starting point are marked in blue.

Next the centroid contour distance curve (CCDC) and Fourier descriptors are calculated. The boundary sequence is rotated so that the first index is the derived starting point. Then using the Euclidean distance metric the distance from boundary point to centroid is calculated. Using the centroid as a frame of reference for our calculation gives us a translation invariant description. The first 64 coefficients of the Fourier transform of the CCDC are the resulting Fourier descriptors. These remaining coefficients are scaled to the interval  $[0, 1]$  by the first coefficient. This makes the Fourier descriptors scale invariant.

To recognize a leaf the above procedure is followed to obtain the Fourier descriptors (FDs) of the leaf in question. A set of the FDs for the known leaf types are

created by averaging the FDs of leaf images in which the leaf and stem were correctly located. To save processing time a database of the Fourier Descriptors for each type of leaf was constructed using file I/O and comma-delimited files. The type of leaf (class) is chosen by minimizing the Time Warp Distance.

The Time Warp Distance (TWD) is not your average metric (ie. it violates the triangle inequality). The two FD's being compared are converted back to the spatial domain. The TWD takes into account phase shifts in the spatial domain. Prominent peaks and valleys are lined up if they are within a few samples of each other. This corrects for inaccuracies due to the starting point or slight variations in shape. The type of leaf that minimizes this distance is suggested as the leaf in question's type. Recognition is complete.

### **Work Performed**

Before any coding or data analysis could take place. I walked around campus, during my free time and in between classes, collecting leaf samples from trees I could identify. After pressing these leaves in books for a day or two, I used Wisconsin Forestry departments webpage tool to identify the tree and leaf types. I proceeded to take pictures of each leaf (59 in total) on a white background with identifying information (common name, technical name, leaf number). I took a few pictures of each leaf, each time in a different orientation to produce more database images and test the rotation and translation invariance of the Fourier descriptors.

The initial coding took a substantial amount of time as I experimented with various methods of dilation, erosion, and skeletons to derive a consistent starting point. I

ended up with a good method that only requires three to seven millimeters of stem to work. Once I had a method to find the starting point, I began coding the following functions:

1. **find\_good\_data** This script loops through all the database images and with human help remembers all of the leaf images that correctly threshold with a stem.
2. **extract\_good\_data** This script reopens the images reported from find\_good\_data and writes the FDs to a comma delimited file creating a database.
3. **classify\_leaf** This script is at the top of the function hierarchy for recognizing a leaf. It has been modified to automatically generate a random set of test data from the good image data. It then calls the following functions to identify the leaf.
4. **find\_ccdc\_fd** This function takes the leaf image and returns the Fourier transform of the CCDC. It calls the following 4 functions.
5. **process\_leaf** This function accepts a color image of a leaf and returns the leaf boundary, stem boundary, and leaf image (binary). The bulk of the processing takes place here.
6. **find\_centroid** This function takes a binary image and returns the centroid of the largest object.
7. **find\_closest** This function takes a point and a vector and returns the value and index of a point from the vector that is closest to the given point.
8. **calc\_ccdc** This function takes the starting index, the centroid, and the leaf boundary and returns the CCDC.

9. **generate\_test\_key** This function reads the FDs database files and returns a matrix where each row corresponds to the average FD of each leaf type.
10. **dtw** The function (obtained from Mathworks Matlab file exchange) calculates the Time Warp Distance between two vectors.

After all the coding was done, I ran several tests using the `classify_leaf` script and kept an accurate record of the results for every run. A random set of 32 images was chosen each time. A few tests were run with all six leaf types, and a few were run with only five. The unknown shrub was removed to see how the method performs with visually unique shapes. The unknown shrub and the black cherry tree both are elongated ovals with a single point.

### Results

<b>Number Correct</b>	<b>Test Size</b>	<b>Restrictions</b>	<b>% Correct</b>	<b>Random Guess %</b>	<b>x Better Than Guess</b>	<b>Test Vector</b>
17	32	none	0.5312	0.1667	3.1872	random
18	32	none	0.5312	0.1667	3.1872	random
18	32	none	0.5625	0.1667	3.3743	random
16	32	None	0.5000	0.1667	2.999	random
21	32	no shrub	0.6562	0.2000	3.281	random
23	32	no shrub	0.7188	0.2000	3.594	random
20	32	no shrub	0.6250	0.2000	3.125	random
21	32	no shrub	0.6562	0.2000	3.281	random

### Discussion

Applying a single threshold to the leaf images properly enhanced the leaf shape and stem 80% of the time. When it did fail, it was due to a few factors. Some leaves didn't have a stem and therefore their image could not be used. The threshold

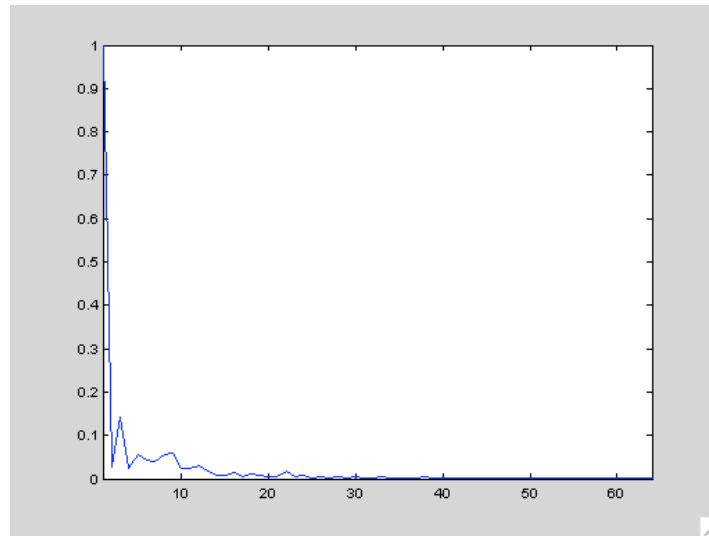
fragmented others because the leaf color intensity was close to some of the background intensity. The leaves were collected in the late Fall/early Winter so they each had a different color. If they were collected while they were still green, the threshold would have been more effective. Finally, sometimes the stem was fragmented.

Finding the leaf itself is currently a manual effort. Now that a database of FDs has been assembled, it would be possible to automatically analyze the six longest boundaries and find the one that best matches the features of a leaf. The eccentricity could also have been used to narrow down the selection by eliminating abnormally long or wide boundaries.

Originally the starting point was to be derived with the use of thinning and nearest neighbor local curvature maximum. This method involves thinning the leaf to its skeleton and then at each endpoint of the skeleton, calculating the curvature of the leaf boundary closest to it. After several experiments with this method, it proved to be computationally inefficient. For example, the skeleton of a sugar maple leaf had upwards of 20 endpoints and the structure of the leaf was not captured. This method would have provided very inconsistent starting points destroying the ability to recognize leaves. After experimenting, I discovered a quick, easy, and consistent method of deriving the starting point (described above is approach).

The CCDC was used as the shape signature because it can capture the periodic nature of a leaf's shape and provides a translation invariant description without any extra work. Classification tests were conducted with 64 Fourier coefficients, an unnecessary amount. The majority of the information was contained in the first 16 coefficients and I

think 32 could be used without loss in recognition precision. 64 were used just in case, to eliminate the chance of errors due to inaccuracy.



(Fourier coefficients for the red oak leaf pictured previously.)

The set of test FDs that each leaf was compared too could have been calculated in a different way. Instead of averaging all the database FDs together, I could have compared the leaf to each individual FD in each class and then minimize the accumulated distance. This may or may not improve results. The actual method of comparison using TWD worked surprisingly well. The distance between a leaf and its class was always very small compared to the other classes, except in a few cases. The black cherry tree leaf and the unknown shrub leaf had the same shape. Their FDs where almost identical, therefore classification of these leaves was almost a 50/50 guess. The unknown shrub was removed from future experiments to see if recognition would improve. It did, but not substantially. The new problem was the red oak leaf being identified as a birch leaf. Both have pointy jagged edges. The additional use of an angle code histogram could

have helped differentiate the two. An angle code histogram is a histogram of the angles of the leaf border. This would characterize the jaggedness of the leaf boundary.

The results above show that this method as implemented has an average recognition rate of 60% and it performs on average 3.2 times better than randomly assigning the leaf to a group. For an initial method, I consider it a success. The FDs clearly capture the leaf shape, and with the use of TWD reasonably accurate results were obtained. With the addition of an angle code histogram, I believe recognition would be improved by another 8 to 20%, putting the method at 68 to 80% accuracy.

### **Research References**

1. Costa, Luciano da Fontoura, and Roberto Marcondes Cesar Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2001.
2. Zhiyong Wang, Zheru Chi, Dagan Feng, and Qing Wang. "Leaf Image Retrieval with Shape Feature." *VISUAL 2000, LNCS 1929* (2000): 477-487.
3. Zhiyong Wang, Zheru Chi, and Dagan Feng. "Shape Based Leaf Image Retrieval." *IEE Proc. – Vis. Image Signal Process., Vol. 150* (2003).
4. Zhang, Dengsheng, and Guojun Lu. "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape signatures." Gippsland School of Computing and Information Technology.
5. Ilaria Bartolini, Paola Ciaccia, and Marco Patella. "WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27 (2005).

## Tasks

<b>Job</b>	<b>Tyler Karrels</b>
Project Proposal	100%
Software Implementation	100%
Research	100%
Presentation	100%
Report	100%
TOTAL	100%

**Tyler Karrels**

---