Assignment #4: Two-dimensional Shapes and Leaf Recognition

For this assignment, the data are the outlines of 24 leaves which belong to 4 sets of 6, each set coming from a single species of plant. Each outline is given as a 2×500 matrix containing the x- and y-coordinates of a polygonal approximation to the outline of the leaf. One of each of these leaves is classified for you, the others are not. Moreover, these leaves have been randomly rotated and scaled so the challenge is to recognize the species solely on the basis of their distinctive 'shapes'. The data can be found in the file `DataAss4.mat` which unpacks to 4 2×500 matrices for the prototypes `leafA, leafB, leafC and leafD` and one $2 \times 500 \times 20$ matrix of the remaining unknown leaves. Moreover, we have held back 24 more leaves, 6 more of each type, for testing! In the first two parts, you are asked to explore the shapes of the prototypes in detail, implementing the geometric smoothing and the medial axis algorithms. In the third, you are asked to look at the whole database and, by a combination of numerical features and by human intuition (botanists having been doing for centuries), guess the classification of the other 20 leaves. Then devise a MatLab routine for automatically classifying any other leaves of these types. We will test your code on the test set and award the winner(s) a chocolate Easter bunny (or equivalent prize for non-chocolate lovers).

(1.) Multi-scale analysis of the prototypes leaves

For each of the contours C, compute the orientation $(\theta_C)_k$ and curvature $(\kappa_C)_k$ at each marker point. Calculate θ for each edge and approximate curvature κ as the first difference of θ divided by the appropriate arclength. Don't forget to allow for 'wrap around', as the contour is closed. Plot the leaves and mark their max, min and zeros of curvature from your data. There may be an awful lot of these. Next implement the geometric heat equation:

$$\frac{\partial C_t}{\partial t} = \kappa \cdot \vec{n}_{C_t}.$$

Do this using the given marker points $(x_k(t), y_k(t)), 1 \leq k \leq N$ which evolve by ODE's, using normals and curvature approximated at each time step. To do this, it is essential not to let the marker points get too close or else instabilities arise. So we propose you write a loop in which you alternately (i) throw away odd marker points which are closer than some limit Δs to one of their even neighbors and (ii) update the marker points using a difference form of the equation and an appropriate Δt. The rule of thumb is to keep $\Delta t < (\Delta s)^2/2$ (it is instructive to see what happens when this is not obeyed). As you go, plot (separately or on top of each other) representative C_{t_i}'s. You should see how it works quite beautifully. Note that the number of max, min and zeros of curvature gradually decrease, giving a coarse portrait of the contour, until the contour gets almost identical to a circle and finally dies. Graph how this number decreases and describe in words how this correlates with the progressive simplification of the contour.

(2.) The medial axis of leaves
For this one, MatLab has graciously given us excellent code, though not in exactly the form we could have wished for. We have made up a small piece of commented MatLab code, in the file `simplemedial.m`, for you to use. Use this to plot the axes for the 4 prototype leaves. Relate the branches of the axis to the maxima of curvature. Now check out the medial axis on a sample of the melted contours C_t from part 1. Note how the medial axis progressively simplifies and show this in plots explained in words. Actually, it doesn't always do so, though I don't think examples will be found in these leaves. Extra credit if you can guess what causes the medial axis to grow a new branch during the heat equation evolution.

(3.) The competition
First plot all the remaining 20 leaves and spend some time guessing which species they belong to. Numerical features often sharpen our intuition, so consider calculating some easy quantities:

1. **integrals:** area, second moments and its eigenvalues, perimeter, total curvature $\int_C |\kappa| ds$ and total square curvature $\int_c \kappa^2 ds$. Note that integrals over the interior of the contour equal line integrals around C by virtue of Green's theorem, e.g. $\iint_{\mathrm{Int}(C)} x^i y^j dxdy = \int_C x^{i+1} y^j/(i+1) dy$, so we get these instantly with one line of MatLab code. But also note that these have dimensions of $(\text{distance})^i$, for $i = -1, 0, 1, 2$ as the case may be and you need an invariant which is dimensionless to distinguish leaves. Note that you can always normalize the leaf under scaling by changing the contour to (cx_k, cy_k) for that c which makes one of these dimensioned quantities to 1.

2. **special points:** numbers of max, min and zeros of curvature or singularities of the medial axis after some amount of smoothing.

3. **features derived from histograms:** The histogram both of θ and κ contain much information, such as modes, quartiles. The graph of θ and κ have even more instructive information but some base point s_0 is need to normalize these.

After some experimenting, formulate some rules for deciding which species a leaf belongs to and carefully explain the purpose of each rule or step. What rules did you expect to work that didn't? What failed? What surprising rules worked well? What obvious visual aspects of the shape did you try hard to quantify? I am not averse to ideas which may be hard to implement, but turn in some piece of code, even if it is a compromise that we can run on our test leaves. Also, turn in guesses on the species of the 20 leaves you are given.