CS6491-2017—Project 2: PCC Cage for FFD

First Last

First Last

ABSTRACT

The first goal of this project is to define and implement a planar cage, R, that is controlled by 6 control points that has, for boundary, a smooth Piecewise-Circular Jordan Curve and that has a branch-free Medial Axis. The second goal is to parameterize R, so as to define a homeomorphism (continuous bijective mapping) between a rectangle and R. The third goal is to create an animation R(t) of R by prescribing a cyclic motion for each one of its control points and then to use the resulting time parameterized map to animate a portion of an image, I, that is defined as the intersection (cut-out) of I with a user-controlled initial version of R.

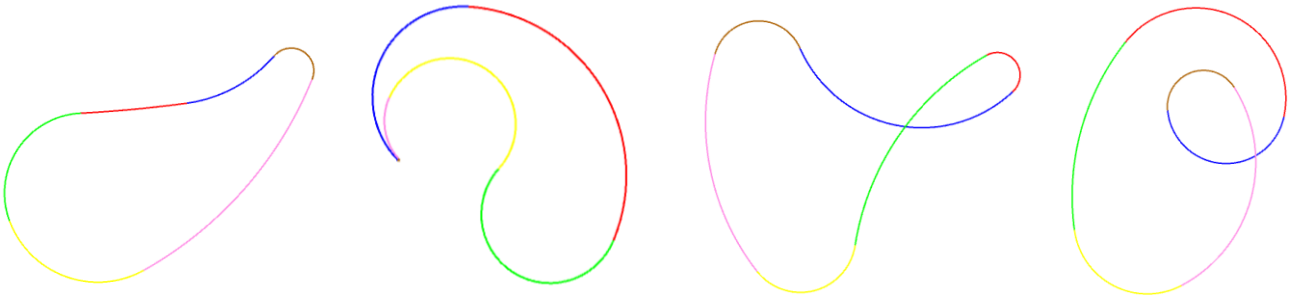
1 Smooth Piecewise-Circular Boundary (SPCB)

We define a ***Smooth Piecewise-Circular Boundary*** (***SPCB***), B, to be a planar curve that has the following properties:

- B is planar
- B is a Jordan curve (closed loop curve without self-crossing or self-overlap)
- B is composed of smoothly joined circular arcs.

Note that we consider a straight-line segment to be a degenerate (special case) circular arc.

We show below (left) an example of an SPCB with arcs drawn using different colors and (right) 3 examples of Piecewise-Circular Curves that are not SPCBs because they are not smooth, self-cross, or self-overlap.



2 Stroke and its control points

We define a ***stroke***, S, to be a planar region bounded by an SPCB made of 6 arcs.

We propose to use a set of 6 ***control points*** (A, B, C, D, E, F) to control the shape of the SPCB that bounds stroke S.

This number of control points is “optimal”. It allows to model a lot of possible strokes. Two different sets of control points represent different strokes. Although more control points can model more possible strokes, it is harder to compute the boundary and to predict what we will get given only the control points.

The relation between a stroke and its control points is best described by the following construction referencing arc by their color and the accompanying diagram showing the arcs of the boundary bS of B in different colors.

We also use the following picture to illustrate the role of each control point as follows:

Point A is the end point of the blue arc and the brown arc of bS;

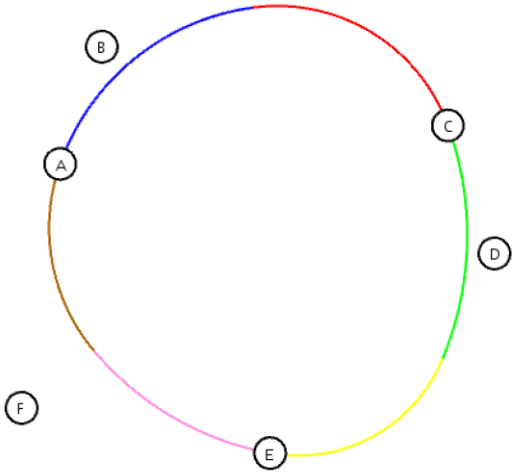
Point B is the end point of vector V(A, B) which is tangent to the blue arc and the brown arc of bS;

Point C is the end point of the red arc and the green arc of bS;

Point D is the end point of vector V(C, D) which is tangent to the red arc and the green arc of bS;

Point E is the end point of the yellow arc and the pink arc of bS;

Point F is the end point of vector V(E, F) which is tangent to the yellow arc and the pink arc of bS;



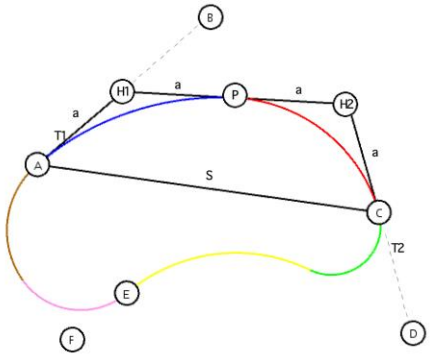
3 Representation and computation of the SPCB of a stroke

To draw and process a stroke S, we first compute its ***explicit representation*** from its control points. The stroke is bounded by 6 arcs. Thus all we need to do is computing exact representation of these arcs. We need representation of each arc as follows.

The representation of each arc stores:

- Points: point C being the center of the supporting circle, point A being the starting point and point B being the ending point
- Bits: 1 bit to indicate the direction of the arc (clockwise or counter-clockwise, thus our arc is uniquely defined) and 1 bit to indicate the convexity of the arc (convex or concave, which helps compute the medial axis in the following sections). In the figure below, the blue arc, with starting point at A and ending point at B, is clockwise and convex. The yellow arc is counter-clockwise and concave.

The figure below shows a typical stroke with its multi-colored boundary.



Given control points (A,B,C,D,E,F) of a stroke S, we compute the explicit representation of S by the following process. We provide the high-level English explanation of each step, the geometric expressions involved in the construction, a diagram, and a justification of their validity.

We use bi-arc method to construct the boundary. We can compute a bi-arc given 2 points and 2 directions. We compute a bi-arc for (A, B, C, D), a bi-arc for (C, D, E, F) and a bi-arc for (E, F, A, B). We take the first bi-arc as an example to illustrate our construction.

Step 1. Compute unit vector $T_1 = \frac{AB}{|AB|}$ and unit vector $T_2 = \frac{CD}{|CD|}$.

Step 2. Find the common end point P of the potential arc starting at A (the blue arc in the above picture) and the potential arc ending at C (the red arc in the above picture). Assume that H_1 is the tip of the hat corresponding to the blue arc and H_2 is the tip of the hat corresponding to the red arc. We connect H_1 and H_2, and P is supposed to be on the line going through H_1 and H_2. Assume that $H_1 = A + aT_1$ and $H_2 = C - bT_2$, we can get $d(H_1, H_2) = a + b$. There are many possible choices of pair (a, b). For simplicity, we add one more constraint $a = b$ such that we can solve for a directly. To sum up, we get the following equations:

$$H_1 = A + aT_1$$
$$H_2 = C - aT_2$$
$$P = H_1 + \frac{1}{2}H_1H_2$$
$$d(H_1, H_2) = 2a$$

Step 3. Get the location of H_1, H_2 and P by solving the equations in Step 2. Then we get triangle (A, H_1, P) and triangle (P, H_2, C). Thus, we can draw the arcs controlled by these triangles (or what we call “hats” in project 1).

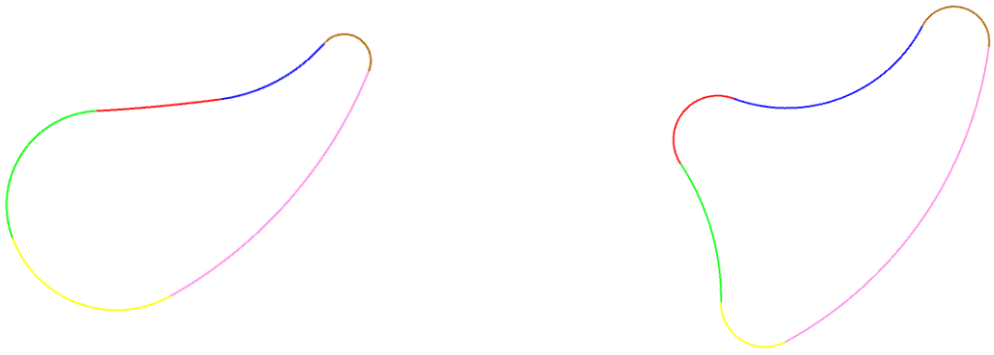
Step 4. Choose points C, D, E, F and repeat from step 1; and then Choose points E, F, A, B and repeat again.

After we computer the exact representation of these arcs, we should also decide the convexity of each arc, which will help in the computation of medial axis. We use shooting ray method to decide if an arc is convex or not. For an arc, we stand at its middle point and shoot a ray which is parallel to the tangent at this point. We compute the number of times this ray intersects with the boundary. If this number if even , then this arc is convex, else it is concave.

4 Medial Axis

The medial axis M of a stroke S is the locus of the centers of circles that are tangent to the boundary in two or more points, where all such circles are contained in S.

We restrict our solution to strokes for which the medial is a smooth curve segment without bifurcation. We show below (left) a valid stroke and (right) an invalid stroke.



5 Computation of the Medial Axis

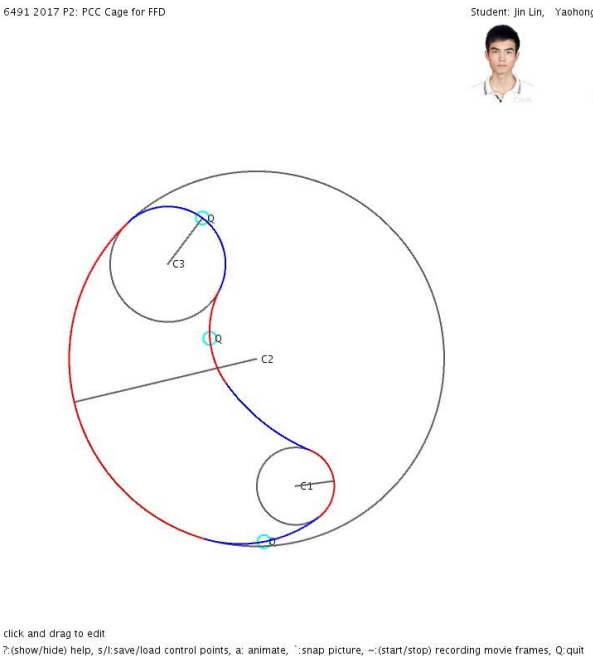
We represent M by an approximating polyline with n vertices M_i that lie on M and are sampled at roughly uniform arc-length steps along M. The details are as follows.

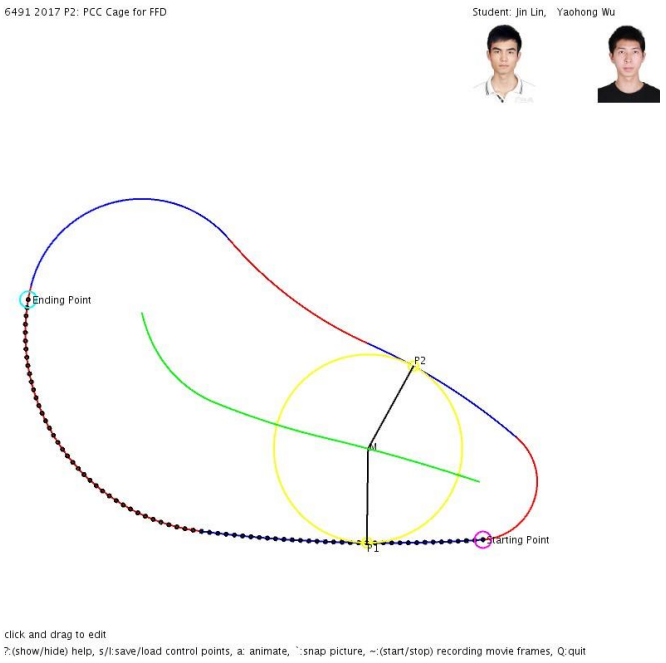
Step 1. Find the "starting" arc and the "ending" arc.

We first find the arcs whose supporting circles are inside the stroke. Ideally, there would be 2 such arcs. If we have more than 2, the medial axis will have branches. Consider the top-left blue arc A_i (we use an italic character to denote an arc) in the following figure (left). We construct its complement A_i' (note that A_i and A_i' form a full circle). We check whether A_i' intersects with other arcs (excluding its neighboring arcs since they always intersected) on the boundary **and** whether the middle point (the cyan point Q in the following picture) of a neighboring arc of A_i is outside the supporting circle of A_i'. The later test can help us discard those huge circles which are likely not to intersect with other arcs on the boundary. It is very possible that this circle may not be inside the stroke. See the huge circle with center C_2 in the following figure (left) for example. In the picture below (left), we find two arcs with supporting circles inside the stroke and we mark them as "starting" arc and "ending" arc respectively.

Step 2. Walking along the arcs between the "starting" arc and the "ending" arc we compute in Step 1.

Use A_i and A_j to denote the "starting" arc and the "ending" arc respectively. We start from the arc right after the "starting" arc, i.e. arc A_{i+1}. The connecting point of A_i and $A_{i+1}$is our starting point. Then we walk along $A_{i+1}, A_{i+2}, \ldots, A_{j-1}$ and sample points with equal arc-length spacing. It is easy to compute arc-length since we are dealing with circular arcs. The arc-length for a circular arc is wr, where w is the angle of it and r is the radius. We also need to determine the spacing. After doing some experiments, we observe that we can get good results when we set the arc-length spacing to be 10 unit length or smaller. Starting point (shown as a small magenta disk) and ending point (shown as a small cyan disk) are shown in the following picture (right). We can also see that the points we sample are shown as small black circles.





Step 3. Compute a maximum disk w.r.t. the boundary at every sampled point.

Assume that point P is on arc A_k, we compute the corresponding maximum disk w.r.t. arc A_k and arc A_l for every $l \neq k$. Because we know their convexities, it is easy to find the center of the maximum disk. However, some maximum disks will intersect with the boundary, which we will discard. We only need to store the smallest disk inside the stroke. The yellow circle in the above right picture shows the valid maximum disk at a point P_1.

Step 4. Connect all the centers of maximum disks to approximate the medial axis.

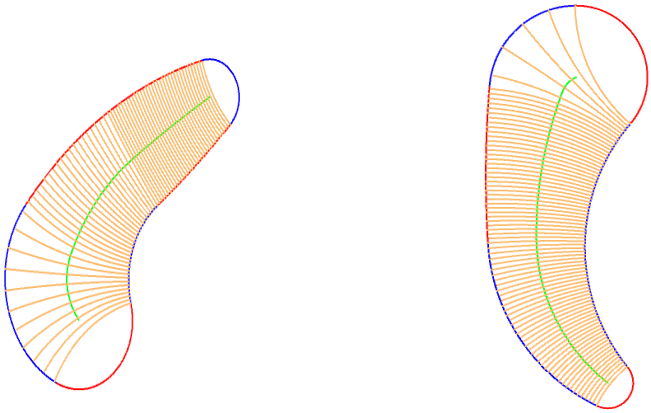
When we arrive at the "ending" arc, we stop sampling. Simply connecting all the centers of the maximum disks we have found so far will give us an approximation of the medial axis. See the green polyline in the above picture (right).

6 Transversals of a stroke

Given a sample M_i on M, the corresponding transversal T_i is a circular arc that starts and ends at points of bS and is tangent at these points to the normal vector to bS at these points. Use P_{i1} ad P_{i2} to denote these two tangency points.

We represent T_i by the circular arc defined by "hat" $P_{i1}M_iP_{i2}$. The computation is the same as that in project 1. Please refer to the corresponding write-up for more details.

We show below the set of transversals for two different stokes.

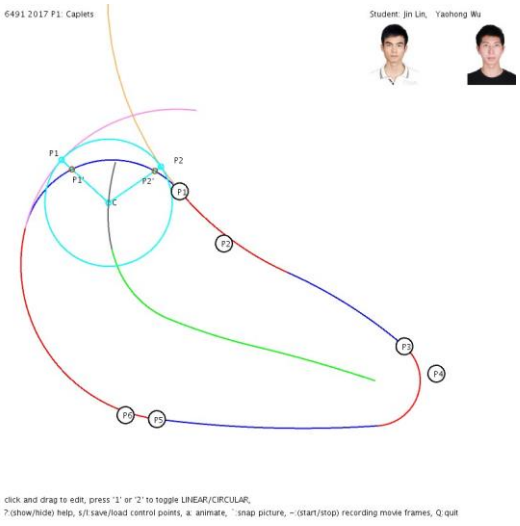


7 Extended transversals

We extend the stroke transversal construction past its medial axis as follows. Briefly speaking, we need to extend some arcs and then compute extended parts of the medial axis. Connecting the extended parts and the one we compute before gives us a new medial axis which touches the boundary of the stroke. In our construction of the extended medial axis, we can store information (the center and two tangency points) of each maximum disk. While this time, the two tangency points are not on the boundary of our stroke, we need to pick some other points, together with the center of the disk, to define a hat. For more details, see below.

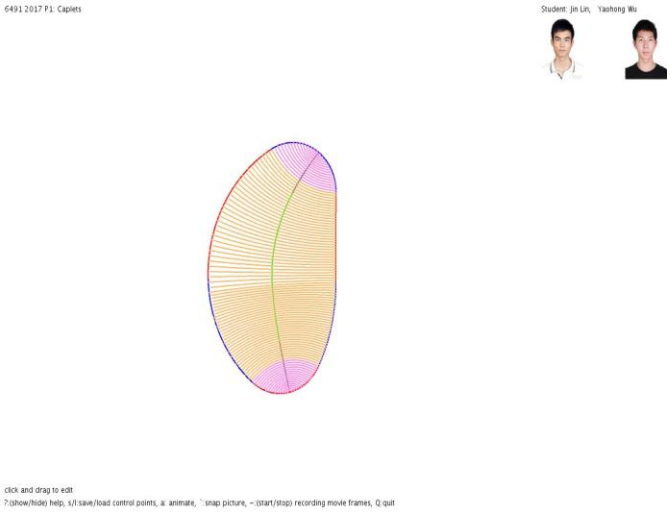
Step 1. Extend arcs.

We extend the arcs near the "starting" arc and the "ending" arc because the natural medial axis ends at exactly the centers of the supporting circles of these arcs. Let's focus on the "starting" arc A_i. We extend A_{i-1} and A_{i+1} to some degree. We observe that it is enough when they are extended to semi-circles. For example, in the following left picture, the pink arc and the red arc on its left form a semi-circle.



Step 2. Extend medial axis and extend transversals.

We compute the medial axis between the extended part of A_{i-1} (pink arc in the following left picture) and the extended part of A_{i+1} (sand arc in the following left picture). Consider a point P_1 at the pink arc. We find its maximum disk D with center C and a tangency point P_2 on the sand arc. CP_1 will intersect with the blue arc between the pink and sand arcs. We compute this intersection point P_1'. Similarly, we can compute P_2'. We use the "hat" $P_1'CP_2'$ to define a transversal at point P.



We show above the set of natural transversals (discussed in the previous section) in sand and the additional transversals in pink. We can also see the natural medial axis is in green while the extended part is in grey (though not obvious).

8 Parameterization and quad mesh of a stroke

We associate the extended transversals with a parameter x in [0,1]. The precise definition of this association is as follows. Assume that the number of transversals is N. We can index our transversals along some direction if the medial axis doesn't have two or more branches. If the index of a transversal is i, then its value of x is $\frac{i}{N}$.

We also parameterize each transversal with a parameter y in [0,1]. Each transversal is a circular arc of a certain circle. Assume that the arc is from point A to point B and the center of the supporting circle of this arc is C. The value of y for a point P is defined as $\frac{CA\hat{}CP}{CA\hat{}CB}$. Simply speaking, y is proportional to the angle between CA and CP.

Given the local ***stroke-coordinates*** (x, y) of a point P with respect to a stroke S, we compute the global location of P as follows. We first find the quad which P is in. We can know the global locations of the four vertices of this quad. Using bi-linear interpolation will give us an approximation of P's global location.

Inversely, given the global location of a point P, its local stroke-coordinates (x, y) are computed as follows. We first find the quad that P is in. We use bi-linear interpolation again to get P's (x, y) since we know the stroke-coordinates of the vertices.

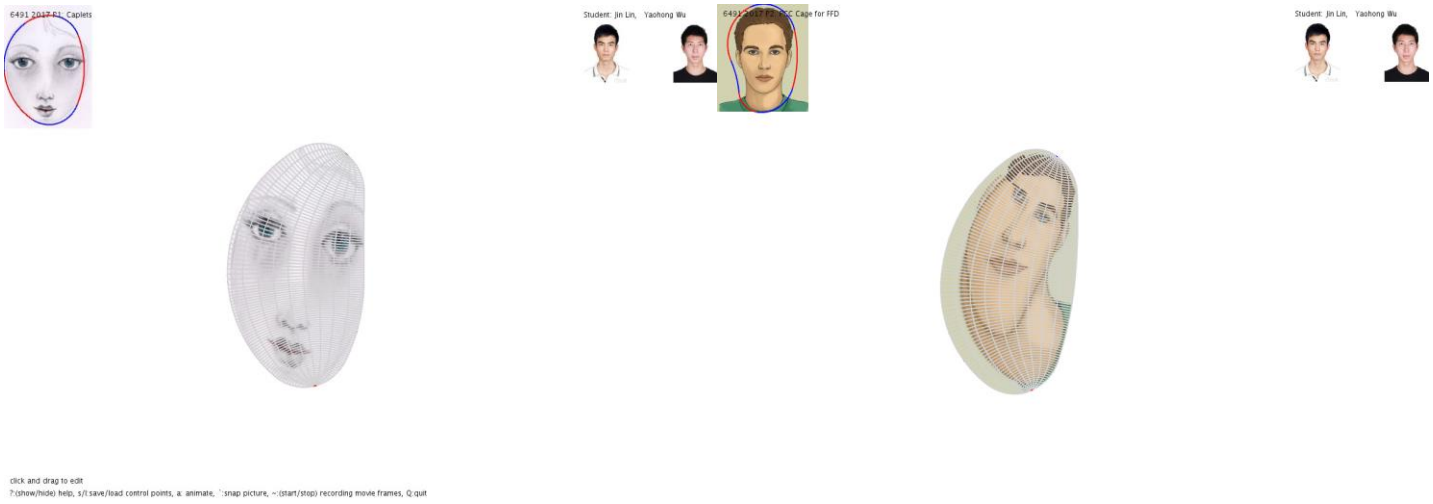
To illustrate the local-to-global conversion, we show below the ***natural quad mesh*** of 2 strokes.

9 Texture transfer

To illustrate the power of strokes and of the mapping above, we show their application to Free-Form Deformation (FFD). We are given an image I that contains a human face, and initial stroke S_1 aligned to roughly lasso the face, and a final stroke S_2 over an empty canvas.

We use the local-to-global conversion of the vertices (crossings) of the natural quad mesh of S_1 to define their texture values (relative locations in I). We then use the local-to-global conversion of the vertices of the natural quad mesh of S_2 to display the quad moved and deformed natural quad mesh of S_2 using texture mapping of I.

Below, we show 2 examples of original images I, with overlaid lasso S_1 and moved and deformed cut-out defined by S_2.



10 Animation

To show the smoothness of our stroke construction when the control points are moved, we prescribe a different motion to each control point of S_2 and show (in the accompanying video) the animated FFD of the image cut-out.