**CS6491-2017—Project 2: PCC Cage for FFD**

****

**Joonho Kim**

**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.*

# 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.

# 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 because:

It allows to model all possible strokes

Two different sets of control points represent different strokes

To justify the above claim, consider…

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.

Point A is the center of the circle that supports the green arc of bS. B is ….

Point D is the center of the circle that supports the red arc of bS….

# 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. This representation stores:

* Points: ??? that represent ???
* Scalars: ??? that represent ???
* Bits: ??? that represent ???

The figure below illustrates the role of these entities for a typical stroke shown 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.

# Medial Axis

The medial axis M of a stroke S is ….

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.

# Computation of the Medial Axis

The medial axis M of a valid stroke S is a smooth curve composed of smoothly joined conic sections. To justify this claim, consider a decomposition of S into ***sectors*** such that each sector is bounded by 4 circular arcs, as shown in the diagram below….

We compute M by first computing the above decomposition of S into sectors and by then computing for each sector a representation of the corresponding segment of M.

The details are as follows…

We sample M to produce samples at roughly uniform arc-length steps along M.

The sampling is performed as followed.

Results are shown below (with the different segments of M color coded differently) for a set of 6 very different strokes and samples marked as small disks.

- OR -

We represent M by an approximating polygon with n+1 vertices Mi vertices that lie on M and are sampled at roughly uniform arc-length steps along M. The details are as follows.

We compute the first vertex M0 as follows:

We compute the spacing s between vertices as follows:

We compute each consecutive vertex Mi as follows:

We terminate this process by detecting…. and by computing the last vertex Mn

A polygonal approximation of M with vertices Mi marked as small disks is shown for a set of 6 very different strokes.

# Transversals of a stroke

Given a sample Mi on M, the corresponding transversal Ti 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.

We represent Ti by …

We compute this representation as follows…

We use this representation to draw Ti by using the procedure … which (at a high level) does the following…

We show below the set of transversals for several very different stokes.

# Extended transversals

We extend the stroke transversal construction past its medial axis as follows…

We compute these transversals as follows….

We show below the set of natural transversals (discussed in the previous section) in black and the additional transversals in grey several very different stokes.

# 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….

We also parameterize each transversal with a parameter y in [-1,1].

Given the local ***stroke-coordinates*** (x,y) of a point P with respect to a stroke S, we define the global location of P as follows.

In practice, to compute that global location, we ….

Inversely, given the global location of a point P, its local stroke-coordinates (x,y) are defined as follows.

In practice, to compute (x,y), we ….

To illustrate the local-to-global conversion, we show below the ***natural quad mesh*** of several strokes S (which is defined by iso-parametric y-lines (x=0.05, x=0.10…. x=0.95) and iso-parametric x-lines (y=-0.95, y=-0.90, .. y=0.90, y=0.95).

To illustrate the global-to-local conversion, we show below ???

# 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 S1 aligned to roughly lasso the face, and a final stroke S2 over an empty canvas.

We use the local-to-global conversion of the vertices (crossings) of the natural quad mesh of S1 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 S2 to display the quad moved and deformed natural quad mesh of S2 using texture mapping of I.

Below, we show several examples of original images I, with overlaid lasso S1 and moved and deformed cut-out defined by S2.

# Animation

To show the smoothness of our stroke construction when the control points are moved, we prescribe a different cyclic motion (with a different duration) to each control point of by S2 and show (in the accompanying video) the animated FFD of the image cut-out.