CS6491 Project 3: Curve Average in 3D

Fall 2015

1 Deliverables

Teams of 1 or 2 people.

One project submission per team.

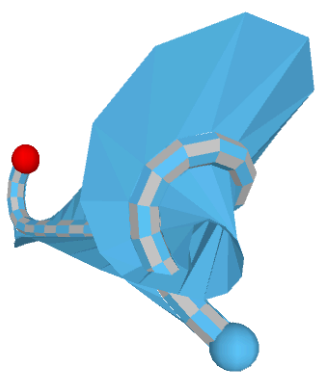
On T-square by Oct 27, before class

Submission should include:

- Concise write up as a short paper with header (title, date, class, names, pictures of authors), a clear problem statement, images showing the input and the various outputs of your implementation, some references to the most relevant prior art, details of the solution (math formulations, algorithm) with diagrams if needed, discussion of results (examples of results for various configurations, discussion of performance, of special cases, of numeric accuracy), and conclusions (what is the essence of your approach, how well does it work). In particular, the write up should define clearly, concisely, and completely the curve average (centerline that is the locus of the centers of touching balls) and its inflation. You should also provide a precise definition of compatibility between the two curves for this to work. Show a couple of examples where the curves are not valid (one with clear invalidity, and one which is almost valid). Your report should also contain concise, clear and complete explanations of:
    - How you resample a curve and compute a parallel transport along a curve to draw a nice tube of quads
    - How you establish correspondence between a point on the green curve and its match on the red curve.
    - How you draw the circular arcs between the corresponding points
    - How you animate the morph between the two curves (compute the intermediate curve at time t)
    - How you compute the inflation ("minimal" tube that is tangent to both curves)
- Zip of the sketch or source code for the project
- A short video showing the user editing the input curve control points, and then showing the resulting tube/track/animation.

2 Source code provided

You may start from the "swirl" source code provided for project 2 (Folder "09 Project 2"). You may also find useful the code "worm" provided in folder "14 Project 3" which displays a tube. Please check whether that code uses parallel transport so the quads of the tube are free from twist.



3 Project details

3.1 Input

The user manipulates a series of control points that define two curves, one red, one green.

They share the start and end points.

I suggest that each curve be defined by 5 control points (using Neville or Bezier formulations). So the user can edit 10 control points.

3.2 Display input curves as tubes
Use some version of the code provided in “worm” or your own that samples the two curves using equispaced samples in arclength and that displays them as checkerboard of quad rings as is done for the “worm”.

3.3 Compute and display the average curve
Compute the centers of minimal balls that are tangent to both curves such that the sum D of the distances on each curve between the contact points of the current ball and of the previous ball is constant. Use a small D to construct a polygonal approximation of the curve and draw it as a thin tube using quad rings. Then use a larger D and draw the balls, but with a small radius, to illustrate this symmetric parameterization. For each ball draw its projections (line segments to the closest projection on each curve). Show results when the 2 curves are in the plane and are compatible (the average curve should be the medial axis of their gap). Show results in 3d for a few of very different configurations. (For example, one curve can be straight and the other can could swirl around it.)

3.4 Display the transversal arcs
For each point on your crude sampling above, display the corresponding circular arc. Show the result for 2 configurations.

The arc is normal to each curve at the contact point and is inscribed in the isosceles triangle formed by the point on the average curve and by its projections.

3.5 Animate morph
When the user presses ‘a’, play a 2 seconds animation that morphs from green to red using a blue tube (drawn as quad rings).

3.6 Show net between the curves
Show the transversal arcs and a few frames of the animation (but displayed as simple curve, not tubes). This should look like a net between the two curves, but it does not sag in the middle. For example, if the curves are in the same plane, the net is in that plane.

3.7 Show the inflation
Display a variable radius tube where the radius matches locally the distance from the average curve to the red or green input curve.

Use twist-free rings of quads to display this inflation and use the symmetric sampling discussed above. Show it together with the small-radius tubes for the input curves, which should stich out from the inflation tube.

3.8 Show the bottom net of the inflation
Provide the option to show only half of each quad ring of the inflation, from green to red (for example going cw). Show the quads as wireframe together with the small tubes of the input curves. This net has a sag which is proportional to the local distance between the curves. Think of applications to architecture, amusement parks, playgrounds, water ducts… and create a few examples that illustrates this potential.

3.9 Video
Produce a video showing the input curves as green and red tubes and the net between them as the user rotates the view and edits the curve. Then switch to a mode where you show the inflation and then its bottom net. Finally show a mode where you display both the net and the bottom net in wireframe, as you are editing the mesh. Ensure (manually) that the editing preserves validity between the curves.