Tube Reconstruction & Gate Stabbing

Ulrich Bauer

Mathematical Geometry Processing
FU Berlin

DFG Research Center Matheon

Methods for Discrete Structures Monday Colloquium, 21.5.2007

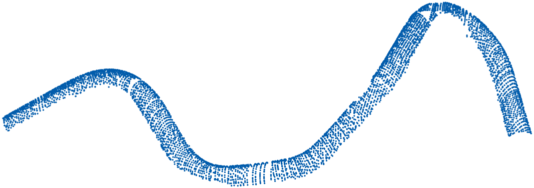
Problem description

A problem from industry...

- Given: a bent metal tube

Problem description

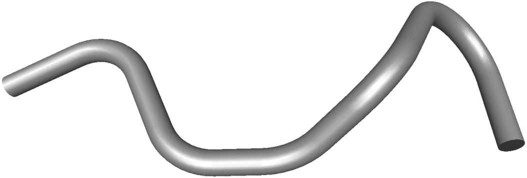
A problem from industry...



- Given: a bent metal tube
- Input: a point cloud of the tube surface (from laser scanner)

Problem description

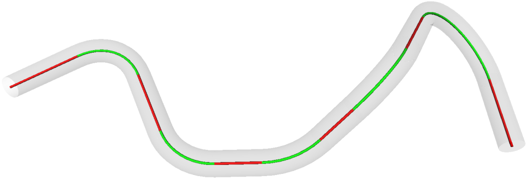
A problem from industry...



- Given: a bent metal tube
- Input: a point cloud of the tube surface (from laser scanner)
- Wanted: a parametric description of the tube surface

Problem description

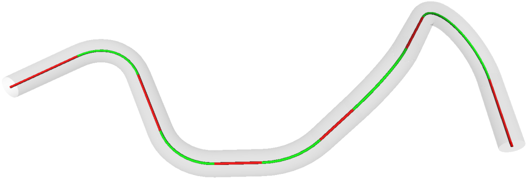
A problem from industry...



- Given: a bent metal tube
- Input: a point cloud of the tube surface (from laser scanner)
- Wanted: a parametric description of the tube surface
- Surface consists of G^1 continuous cylinder and torus segments

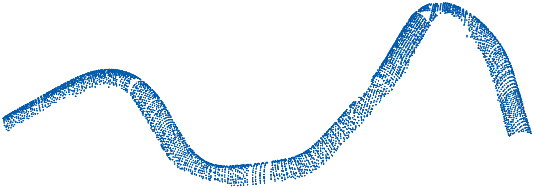
Problem description

A problem from industry...



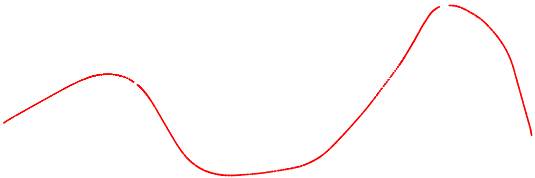
- Given: a bent metal tube
- Input: a point cloud of the tube surface (from laser scanner)
- Wanted: a parametric description of the tube surface
- Surface consists of G^1 continuous cylinder and torus segments
- *Pipe surface* (envelope of a ball moving along the *spine curve*)

Overview of the algorithm



Decompose into subproblems:

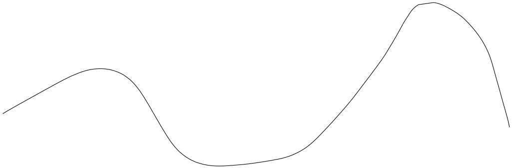
Overview of the algorithm



Decompose into subproblems:

- Project the surface points onto the spine curve

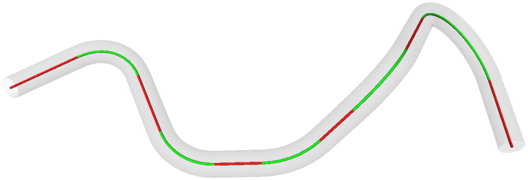
Overview of the algorithm



Decompose into subproblems:

- Project the surface points onto the spine curve
- Join and simplify the spine curve points

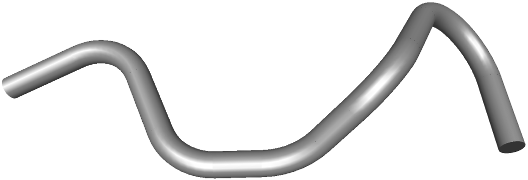
Overview of the algorithm



Decompose into subproblems:

- Project the surface points onto the spine curve
- Join and simplify the spine curve points
- Approximate the curve by G^1 continuous arcs and line segments

Overview of the algorithm



Decompose into subproblems:

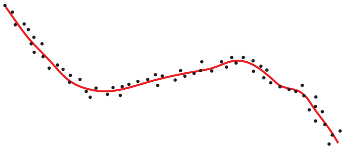
- Project the surface points onto the spine curve
- Join and simplify the spine curve points
- Approximate the curve by G^1 continuous arcs and line segments

Moving least squares projection

To find the spine curve, we adapt the moving least squares method.

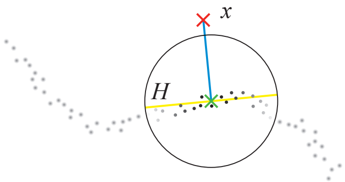
MLS curve (D. Levin, 1998)

Define a curve from a set of points



Moving least squares projection

To find the spine curve, we adapt the moving least squares method.



MLS curve (D. Levin, 1998)

Define a curve from a set of points

Let x be any point in the plane.

- Locally biased least squares fitting of a hyperplane H
- Center of bias moves with the projection of point x onto H
- x lies on $H \Leftrightarrow x$ is a point on the curve

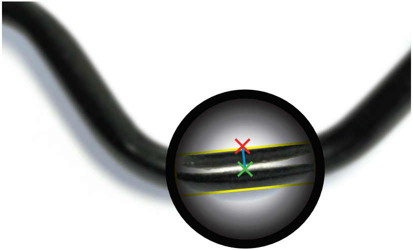
Cylinder MLS projection

Idea: extend MLS projection to other primitives than hyperplanes

Cylinder MLS projection

Idea: extend MLS projection to other primitives than hyperplanes

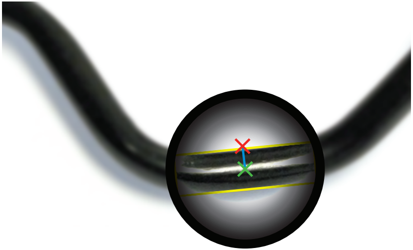
- Locally fit cylinders to surface samples



Cylinder MLS projection

Idea: extend MLS projection to other primitives than hyperplanes

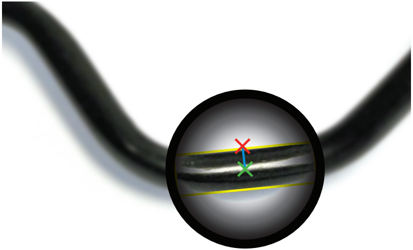
- Locally fit cylinders to surface samples
- Project samples onto axis of cylinders



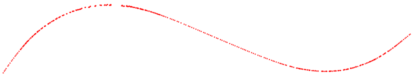
Cylinder MLS projection

Idea: extend MLS projection to other primitives than hyperplanes

- Locally fit cylinders to surface samples
- Project samples onto axis of cylinders
- Goal: approximate spine curve of pipe surface

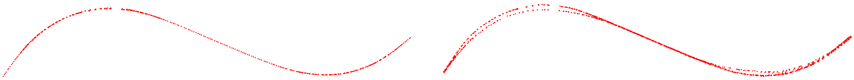


Comparison to previous work



Our method

Comparison to previous work



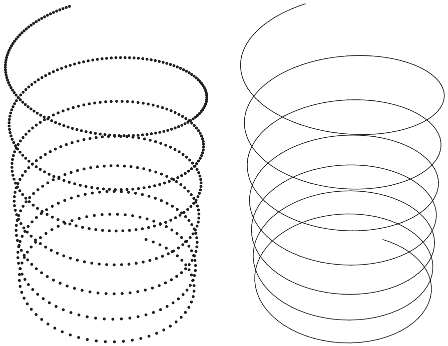
Our method

In-Kwon Lee, 2000

- Previous work: only use estimated normals and radius to shift samples
- Additional smoothing required

Connect the dots

Curve reconstruction: widely considered problem

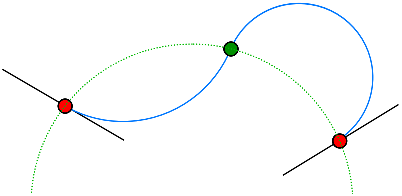


We used the NN-Crust algorithm (Tamal Dey, 1999)

Local control of arc spline

Consider a biarc in the plane with fixed end points and tangents.

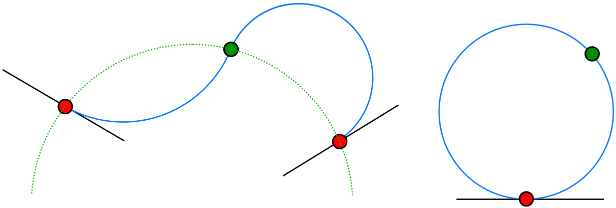
- One degree of freedom left



Local control of arc spline

Consider a biarc in the plane with fixed end points and tangents.

- One degree of freedom left
- ...except when start and end points and tangents are the same: two degrees of freedom



Arc splines are difficult to handle algorithmically!

Problem definition

Find a G^1 continuous curve of arc and line segments *(arc-line spline)*

- with distance $< \epsilon$ to the vertices of the input polygon
- with minimum number of segments

This problem is similar to a simpler problem...

Polygon simplification

Problem: Given a polygonal curve P.
Find another polygonal curve P' with distance $d(P, P') < \epsilon$ with minimum number of segments.

Framework (H. Imai, M. Iri, 1986) used by many algorithms:

- Build a shortcut graph over vertices of polygon
- An edge e_{ij} is in the graph if the line segment $\overline{p_i p_j}$ approximates $[p_i, p_{i+1}, \dots, p_j]$
- Find a shortest path through the shortcut graph
- $\mathcal{O}(n^3)$ time, $\mathcal{O}(n)$ space (don't construct graph explicitly)

Restriction to vertices of input polygon
For unrestricted vertex positions in $\mathbb{R}^3$, no algorithm known

Arc-line spline simplification

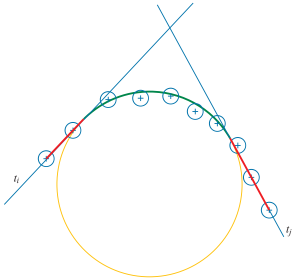
Try something similar for arc-line splines

- Restrict solution set to something we can handle by a graph
- Find optimal solution for the restricted set using BFS

Estimated tangent lines as vertices of graph

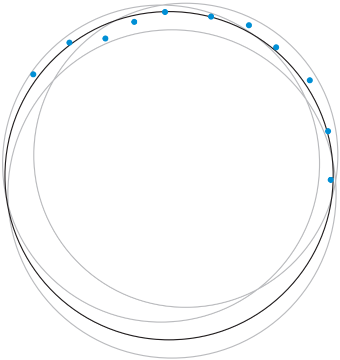
Problem: tangent lines are in general not coplanar

- To compute edge e_{ij}, adjust (tilt) tangent line t_j to make it coplanar with t_i
- Shortest path to j determines tangent line t_j for further computations



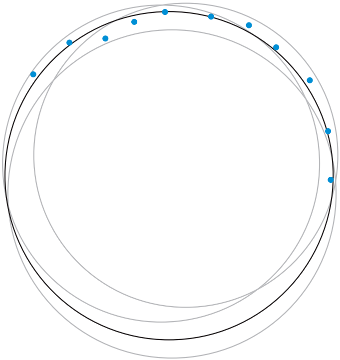
Demonstration

Computing the set of approximating circles



Compute the set of circles approximating a set of points in the plane with distance less than ϵ

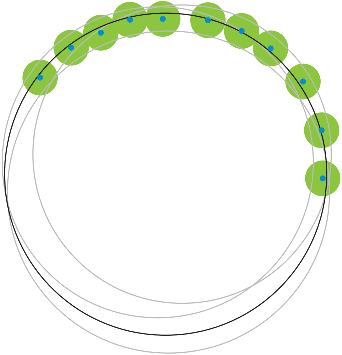
Computing the set of approximating circles



Compute the set of circles approximating a set of points in the plane with distance less than ϵ

- Intersection of sets of approximating circles for each point

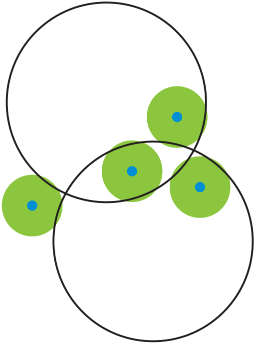
Computing the set of approximating circles



Compute the set of circles approximating a set of points in the plane with distance less than ϵ

- Intersection of sets of approximating circles for each point
- In general, approximating circles (stabbing an ϵ-ball aroud a point) are not a convex set (intersections are not connected)

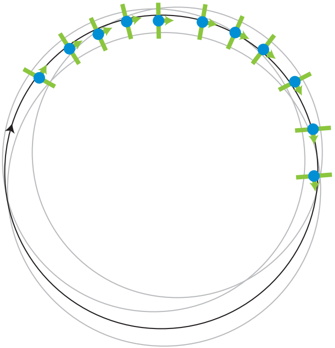
Computing the set of approximating circles



Compute the set of circles approximating a set of points in the plane with distance less than ϵ

- Intersection of sets of approximating circles for each point
- In general, approximating circles (stabbing an ϵ-ball aroud a point) are not a convex set (intersections are not connected)

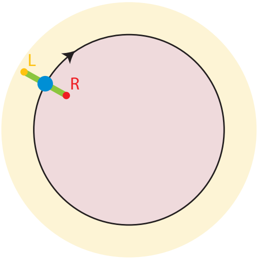
Computing the set of approximating circles



Compute the set of circles approximating a set of points in the plane with distance less than ϵ

- Intersection of sets of approximating circles for each point
- In general, approximating circles (stabbing an ϵ-ball aroud a point) are not a convex set (intersections are not connected)
- But for ϵ-gates (line segments, with stabbing direction), they are always connected

The set of circles stabbing a gate



When does a circle stab a gate?

- Left (right) endpoint of the gate is to the left (right) of the circle.
- Set of circles stabbing a gate: intersection of two sets of circles with one point on their left (right) side
- Reduce gate stabbing to describing all circles with a point on its left (right) side.

A fundamental observation

Observation

Consider the set of circles with a point (x, y) on their left (right) side.

The intersection of any number of these sets is still connected.

We will use these sets as halfspaces to construct convex polyhedra.

Oriented circles

The (algebraic) set

$$C(a,b,c,d) = \{(x,y) \mid a(x^2+y^2) + bx + cy + d = 0\}$$

describes (for $a \neq 0$) a circle with center $\left(-\frac{b}{2a}, -\frac{c}{2a}\right)$ and radius $\sqrt{\left(\frac{b}{2a}\right)^2 + \left(\frac{c}{2a}\right)^2 - \frac{d}{a}}$

- Homogeneous coordinates: $C(a,b,c,d) = C(\lambda\, a, \lambda\, b, \lambda\, c, \lambda\, d)$
- The sign of a lets us define an orientation: clockwise for $a < 0$, counterclockwise for $a > 0$.
- For $a = 0$, we get oriented lines pointing in direction (b, c).
- Different to Laguerre geometry (interior/exterior of a circle with radius 0 not defined!)

Oriented disks

The (semialgebraic) sets

$$C_+(a,b,c,d) = \{(x,y) \mid a(x^2+y^2) + bx + cy + d > 0\}$$

$$C_-(a,b,c,d) = \{(x,y) \mid a(x^2+y^2) + bx + cy + d < 0\}$$

describe the points to the right (left) of an oriented circle $C(a,b,c,d)$. We will call these sets *(open) oriented disks*.

- Note that $C_-(a,b,c,d) = C_+(-a,-b,-c,-d)$.
- For $a < 0$, $C_+(a,b,c,d)$ is the interior of the circle $C(a,b,c,d)$; for $a > 0$, it is the exterior.

Halfspaces of oriented disks

Conversely, each point (x, y) defines the set of all oriented disks containing (x, y):

$$D_+(x, y) = \{C_+(a, b, c, d) \mid (x, y) \in C_+(a, b, c, d)\}$$

$$D_-(x, y) = \{C_+(a, b, c, d) \mid (x, y) \in C_-(a, b, c, d)\}$$

We observed that these sets have characteristic properties of open convex halfspaces:

Halfspaces of oriented disks

Lemma

For two oriented disks $C_1 = C_+(a_1, b_1, c_1, d_1)$ and $C_2 = C_+(a_2, b_2, c_2, d_2)$ containing (x, y), each oriented disk $C_\lambda = (1-\lambda)C_1 + \lambda C_2$ with $0 \leq \lambda \leq 1$ also contains (x, y).

Proof.

We have

$$
\begin{aligned}
a_\lambda(x^2 + y^2) + b_\lambda x + c_\lambda y + d_\lambda \\
&= (1-\lambda)(a_1(x^2 + y^2) + b_1 x + c_1 y + d_1) \\
&\qquad + \lambda(a_2(x^2 + y^2) + b_2 x + c_2 y + d_2) \\
&> 0
\end{aligned}
$$

□

Halfspaces of oriented disks

Lemma

For two oriented disks $C_1 = C_+(a_1, b_1, c_1, d_1)$ *and* $C_2 = C_+(a_2, b_2, c_2, d_2)$ *containing* (x, y)*, each oriented disk* $C_\lambda = (1-\lambda)C_1 + \lambda C_2$ *with* $0 \leq \lambda \leq 1$ *also contains* (x, y)*.*

Proof.

We have

$$\begin{aligned} & a_\lambda(x^2+y^2) + b_\lambda x + c_\lambda y + d_\lambda \\ &\qquad = (1-\lambda)(a_1(x^2+y^2) + b_1 x + c_1 y + d_1) \\ &\qquad\qquad + \lambda(a_2(x^2+y^2) + b_2 x + c_2 y + d_2) \\ &\qquad > 0 \end{aligned}$$

Halfspaces of oriented disks

Lemma

For two oriented disks $C_1 = C_+(a_1, b_1, c_1, d_1)$ *and* $C_2 = C_+(a_2, b_2, c_2, d_2)$ *containing* (x, y)*, each oriented disk* $C_\lambda = (1-\lambda)C_1 + \lambda C_2$ *with* $0 \leq \lambda \leq 1$ *also contains* (x, y)*.*

Proof.

We have

$$\begin{aligned} a_\lambda(x^2 + y^2) + b_\lambda x + c_\lambda y + d_\lambda \\ &= (1-\lambda)(a_1(x^2+y^2) + b_1 x + c_1 y + d_1) \\ &\quad + \lambda(a_2(x^2+y^2) + b_2 x + c_2 y + d_2) \\ &> 0 \end{aligned}$$

Halfspaces of oriented disks

Lemma

For two oriented disks $C_1 = C_+(a_1, b_1, c_1, d_1)$ *and* $C_2 = C_+(a_2, b_2, c_2, d_2)$ *containing* (x, y)*, each oriented disk* $C_\lambda = (1-\lambda)C_1 + \lambda C_2$ *with* $0 \leq \lambda \leq 1$ *also contains* (x, y)*.*

Proof.

We have

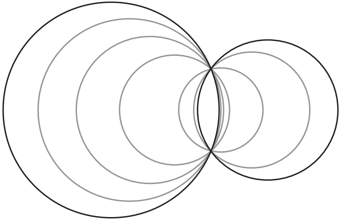
$$\begin{aligned} a_\lambda(x^2+y^2) + b_\lambda x + c_\lambda y + d_\lambda \\ &= (1-\lambda)\left(a_1(x^2+y^2) + b_1 x + c_1 y + d_1\right) \\ &\quad + \lambda\left(a_2(x^2+y^2) + b_2 x + c_2 y + d_2\right) \\ &> 0 \end{aligned}$$

From halfspaces to polyhedra

Idea: Use well-developed theory of convex polyhedra!

- A convex polyhedron is the intersection of a finite number of halfspaces
- Compute vertices, facets, redundant halfspaces
- Some questions to consider...

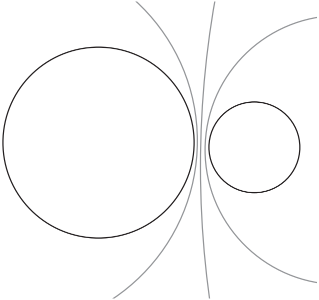
Coaxal circles



The interpolating circles C_λ are *coaxal circles*

- All C_λ share two common points: the intersection of C_1 and C_2.

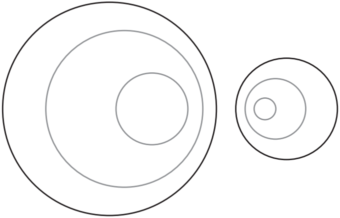
Coaxal circles



The interpolating circles C_λ are *coaxal circles*

- All C_λ share two common points: the intersection of C_1 and C_2.
- The intersection points may have imaginary coordinates.

Coaxal circles



The interpolating circles C_λ are *coaxal circles*

- All C_λ share two common points: the intersection of C_1 and C_2.
- The intersection points may have imaginary coordinates.
- The interpolating circle may also have imaginary radius.

Circles with imaginary radius

A halfspace may contain improper circles (imaginary radius).
But improper circles cannot stab a gate:

Lemma

The interior of a circle with imaginary radius is empty.

Proof.

For $\left(\frac{b}{2a}\right)^2 + \left(\frac{c}{2a}\right)^2 - \frac{d}{a} < 0$ (imaginary radius) and $a < 0$ (interior of circle), we always have $a(x^2 + y^2) + bx + cy + d < 0$:

- $a(x^2 + y^2) + bx + cy + d$ has maximum at circle center $(x, y) = \left(-\frac{b}{2a}, -\frac{c}{2a}\right)$.
- Function value at center is exactly $\left(\frac{b}{2a}\right)^2 + \left(\frac{c}{2a}\right)^2 - \frac{d}{a}$.

Therefore, it is negative for all points (x, y). □

Antipodal oriented disks

Consider two oriented disks $C_1 = C_+(a, b, c, d)$ and $C_2 = C_+(-a, -b, -c, -d)$.

- In this case, linear interpolation of the coordinates does not work.

However, our open halfspaces do not contain antipodal disks:

- No circle has a point both in its interior and its exterior.

Stabbing lines

All this also works for lines: set $a = 0$.

- A point defines a set of open halfspaces containing the point:

$$bx + cy + d > 0$$

- Interpolation of halfspaces containing a point again yields halfspaces containing that point.
- No problems with complex coordinates or radii.

Embedding into Euclidean space

We can embed the sets of oriented circles into $\mathbb{R}^4$ canonically:

- A circle $C(a, b, c, d)$ corresponds to a point $p = (a, b, c, d)$
- A point (x, y) defines a halfspace of all disks containing that point by

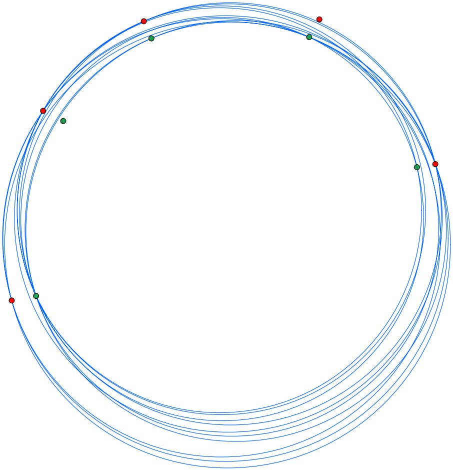
  $$\left\{p \mid \left(\left(x^2 + y^2\right), x, y, 1\right) \cdot p > 0\right\}$$

  and all disks not containing that point by

  $$\left\{p \mid \left(-\left(x^2 + y^2\right), -x, -y, -1\right) \cdot p > 0\right\}$$

These sets can therefore be handled using standard algorithms for convex polyhedra.

A small example



Conclusion

The set of circles (lines) stabbing a set of gates can be represented by convex polyhedra in $\mathbb{R}^4$ ($\mathbb{R}^3$).

Open questions

- Can we describe this observation in terms of convexity on Riemannian manifolds?
- Can we use this machinery for curve approximation by arc splines?