Goals and Outline

(1) SLEFEs: Enclosing Functions (and SLEVEs for curves and surfaces)
(2) *Mid-structures*: *Quantitatively* Coupling Curved and pw Linear Geometry
(3) *Constrained Design*: One-sided fitting

Midstructures

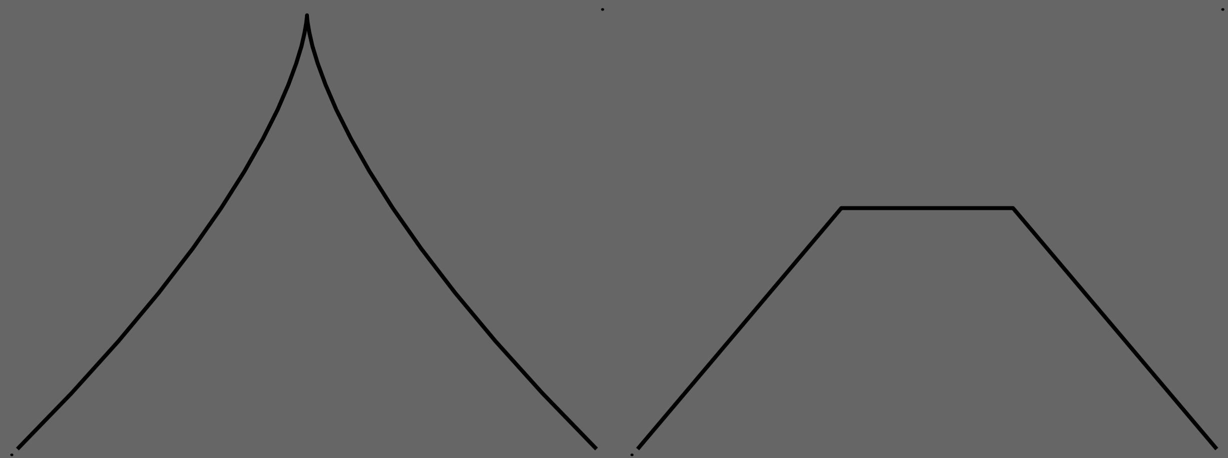
$$
\begin{aligned}
\overline{\underline{x}}(t) &:= (\overline{x}(t)+\underline{x}(t))/2 = \frac{1}{2}\Big(\sum_{\mu=0}^{m} \tilde{x}_\mu \mathbf{h}_\mu^m(t) + \sum_{\mu=0}^{m} \underset{\sim}{x}_\mu \mathbf{h}_\mu^m(t)\Big) \\
&= x_0(1-t) + x_d t + \sum_{\mu=1}^{m}\sum_{\nu=1}^{d-1} \mathcal{F}_\nu x \; \frac{a[d,m,+,\nu,\mu] + a[d,m,-,\nu,\mu]}{2} \mathbf{h}_\mu^m(t) \\
&= Lx(t) + \mathcal{F}x \;\cdot \underline{\overline{\mathbf{a}}} \;\cdot \mathbf{h}(t), \qquad \underline{\overline{\mathbf{a}}} \in \mathbb{R}^{d+1\times m+1}.
\end{aligned}
$$

— Modification for Bézier: endpoints are x_0 and x_d.
— Similar to center curve for interval splines.
— Well-defined for vector-valued curve or surface.

. Why is it useful?

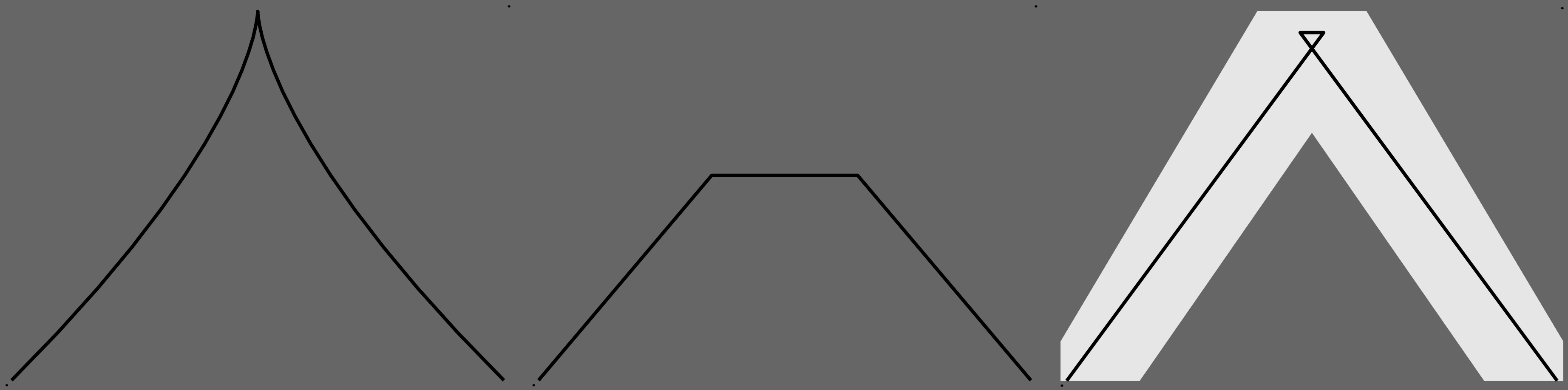
Application: Evaluation

compare to sampling:



Exact curve, sampled curve (4 points)

Application: Mid-Path



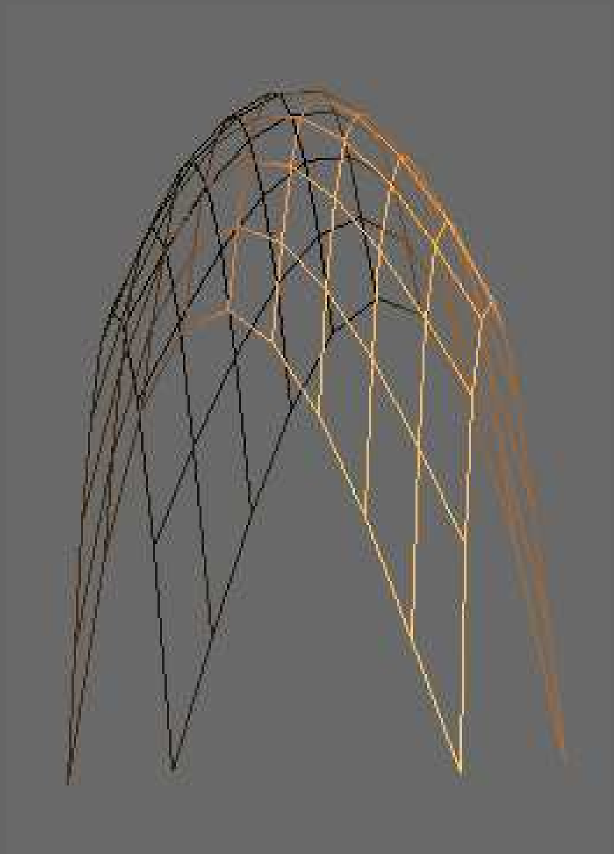
Exact curve, sampled curve, mid-path

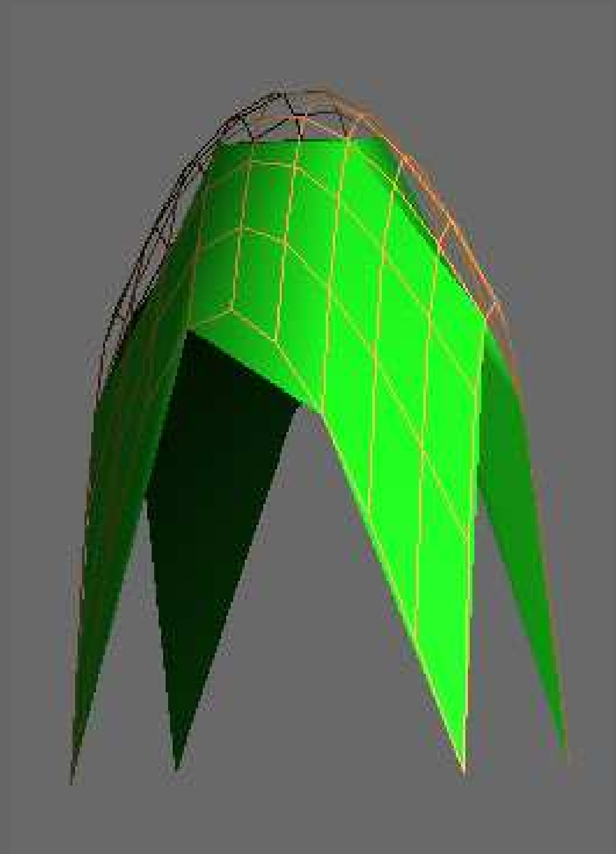
.
.
.

max-norm approximation,
lower cost* than evaluation,
error bounds available.

Mid-path = $(\underline{x} + \overline{x})/2$.

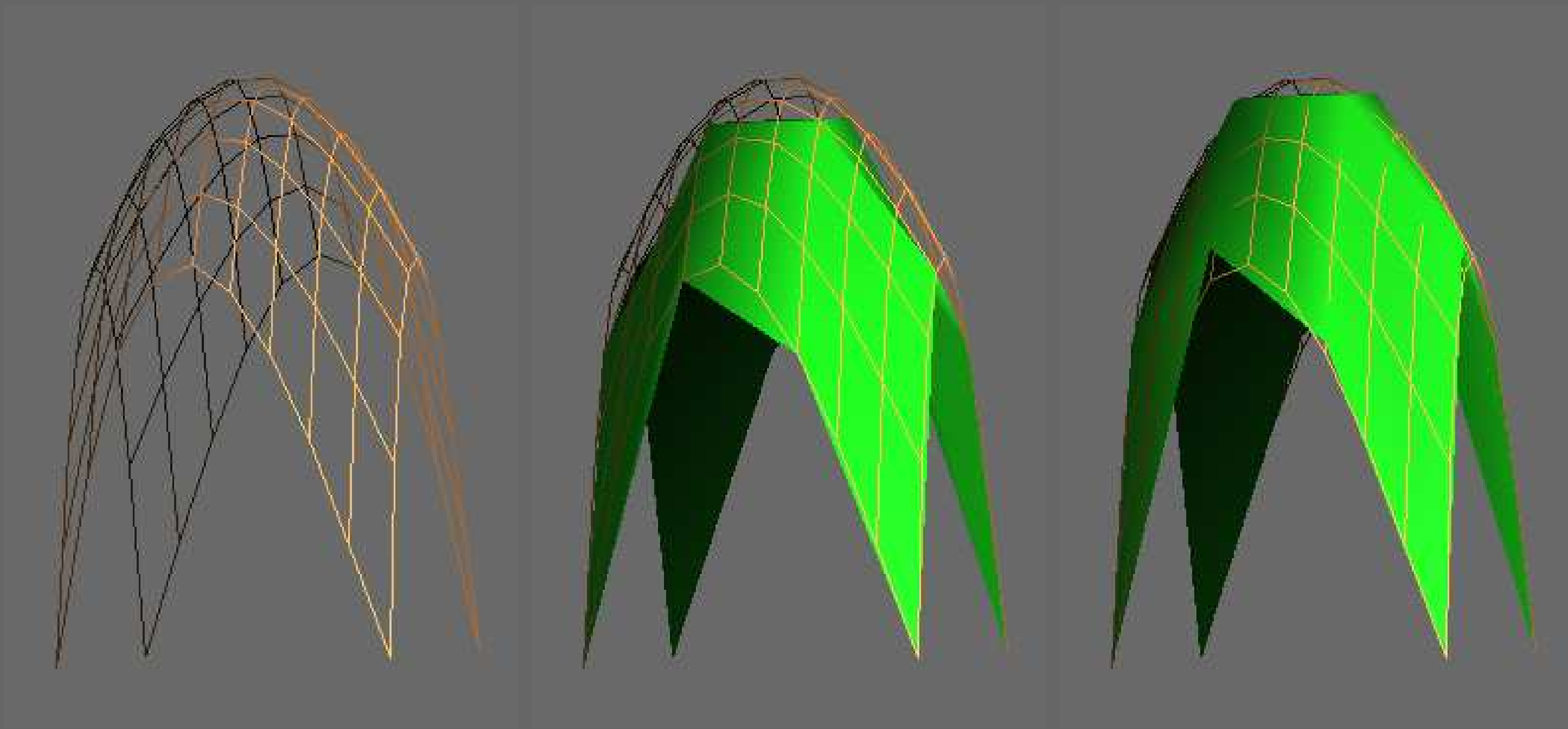
Mid-Patch: Approximate Rendering





finely sampled, coarsely sampled (green): budget=16 points

Mid-Patch: Approximate Rendering



. max-norm approximation,
. lower cost than evaluation,
. free error bounds.

(Bézier: boundary is defined by control points of the boundary only.)

Max-norm approximation

Mid-structures equi-oscillate.

Why not Chebyshev economization?

Why not Remez algorithm?

One-sided approximation!

Midpath Control Structures (Inversion)

If *number of segments = number of control points*, e.g. $m = d$
then $\underline{\overline{\mathbf{a}}}$ *is invertible**!

Reverse the midpath *coefficient* computation:

$$\underline{\overline{x}} - Lx = \mathcal{F}x \quad \underline{\overline{\mathbf{a}}}.$$

solve for $\mathcal{F}x$ and reconstruct the function

$$x(t) = Lx(t) + \mathbf{a}(t) \cdot \ \mathcal{F}x$$

mid-struct and control-polytope = spline coefficients in *different bases*!

Midpath Control Structures (Inversion)

Any broken line $\underline{\overline{x}}$ (plus boundary conditions Lx)
has an associated x
(Thm:) that lies in the same flat*.

Examples

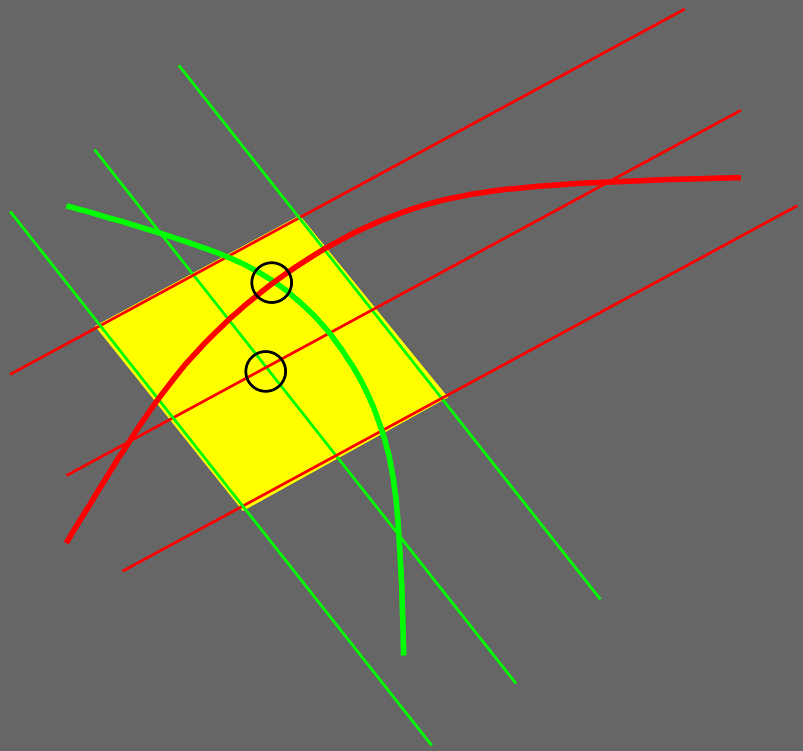
spline curve

box-spline

Application: Intersection

Robustness: *pw-linear*
Consistency: *intersect mid-structures*

2D: error at most $2\min\{\epsilon, \tilde{\epsilon}\}$ (false positive, false negative)



(P & Wu 2003): SLEFE construction to match a given ϵ.

Application Intersection

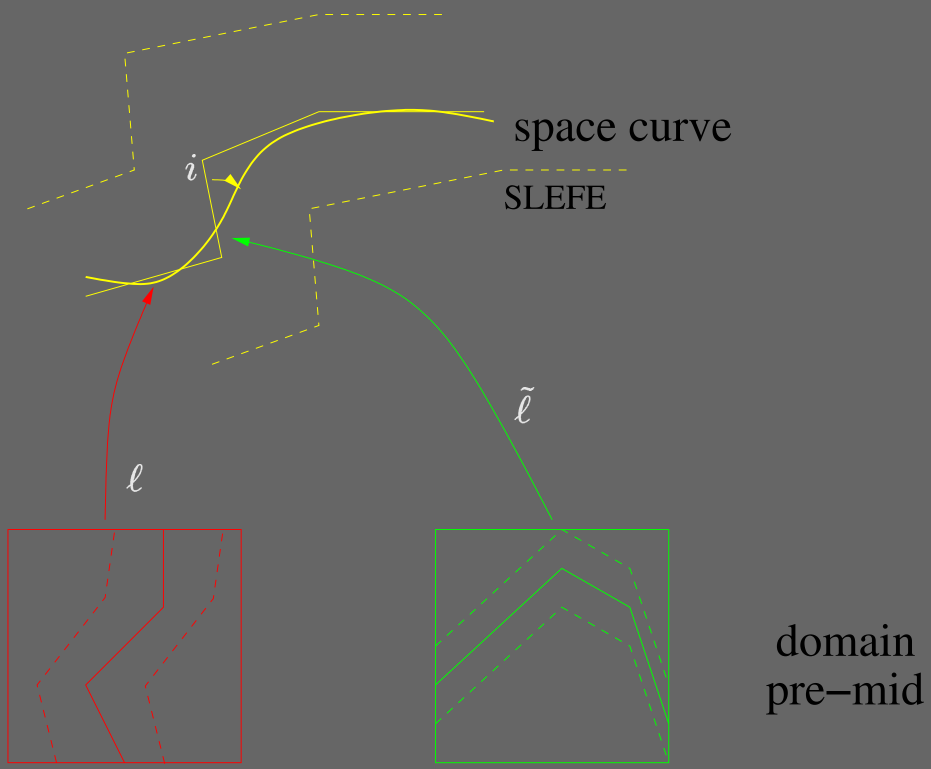
Robustness

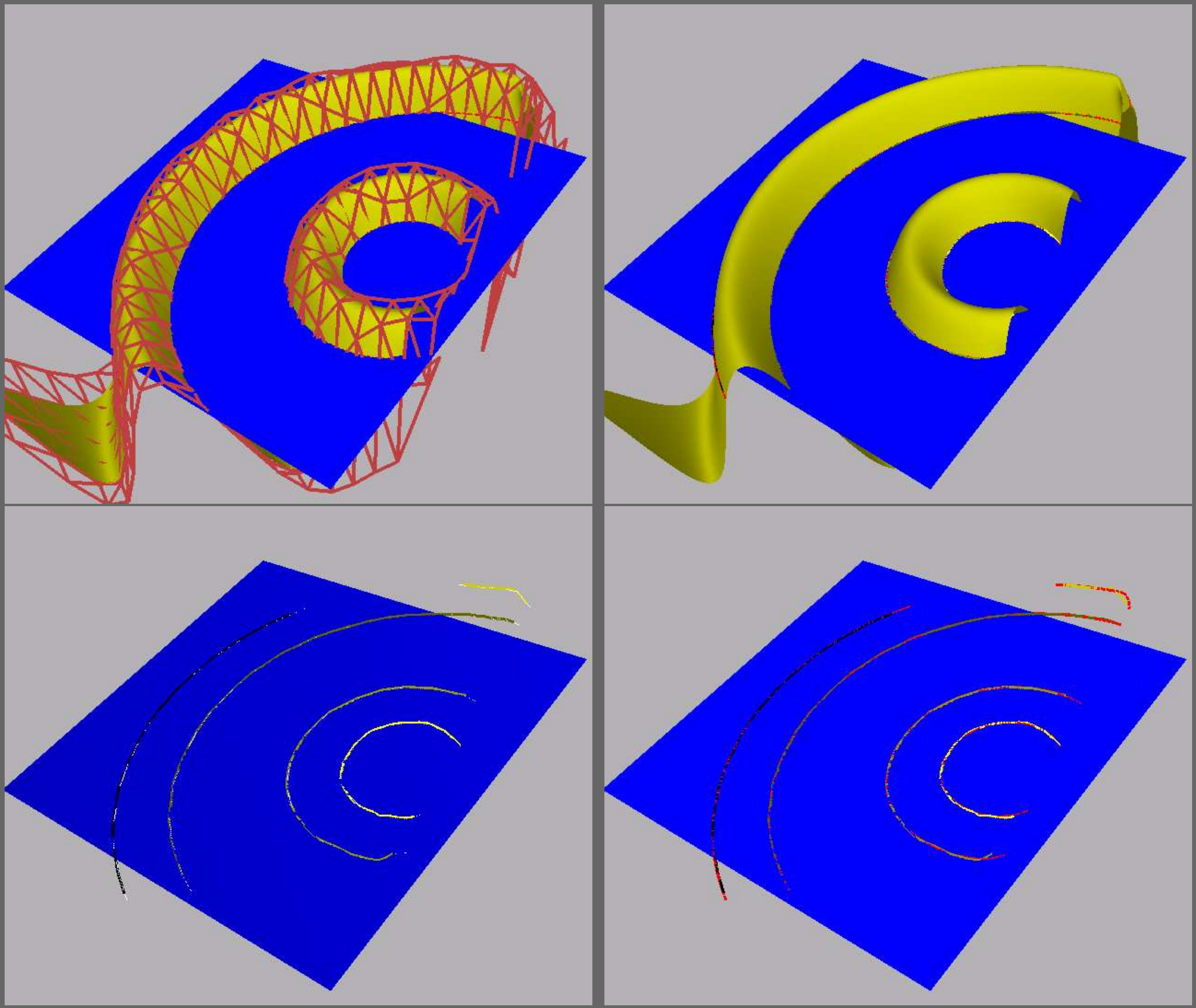
Consistency

intersect pw-linear mid-structures 3D, 2 variables:

interpret intersection of $\underline{\overline{\mathbf{x}}}$ and $\underline{\overline{\tilde{\mathbf{x}}}}$ as mid of a space curve.

Unique representer: $i(\ell(u)) = i(m) = i(\tilde{\ell}(\tilde{u}))$



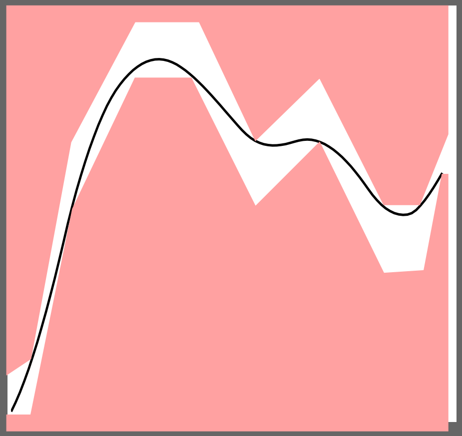


Goals and Outline

(1) SLEFEs: Enclosing Functions
(2) Mid-structures: Quantitatively Coupling Curved and pw Linear Geometry
(3) Constrained Design: *One-sided fitting*

Inverse problems: Channel

Given two locally non–intersecting input polygons, construct a spline that stays between and consists of a small number of pieces.

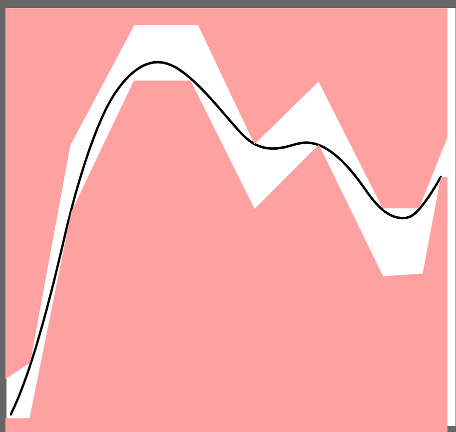


Channel

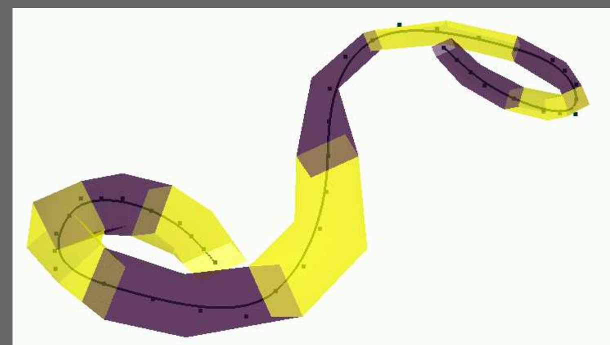
Continuous optimization problem!

Inverse problems: Channel

Given two locally non–intersecting input polygons,
construct a spline that stays between and consists of a small number of pieces.



Channel



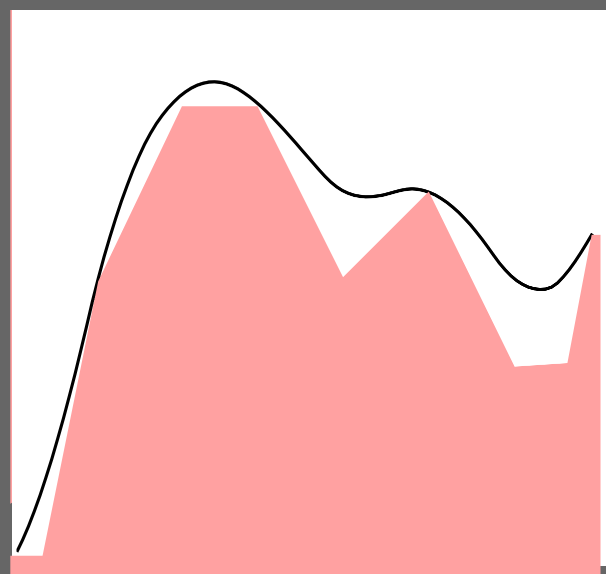
Channel 3D

Solve for spline coefficients such that SLEFE does not intersect.

Idea: fit a SLEFE into the channel = *linearize and discretize* – a linear program!

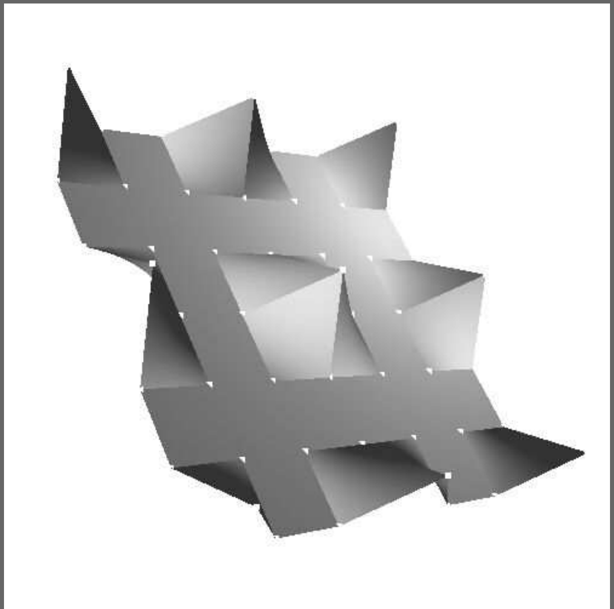
Inverse problems: Cover

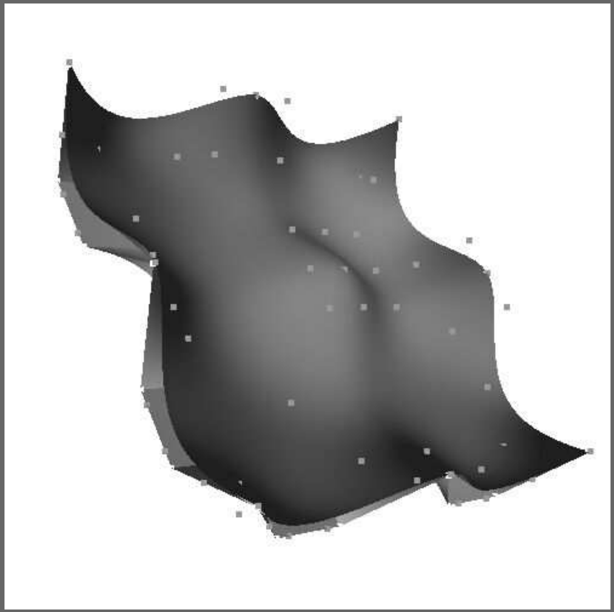
Given a barrier, construct x of a given degree and smoothness to stay close to and entirely above.



Motivation: bridge to computational geometry algorithms; *determinate assembly*, simplification for intersection testing.

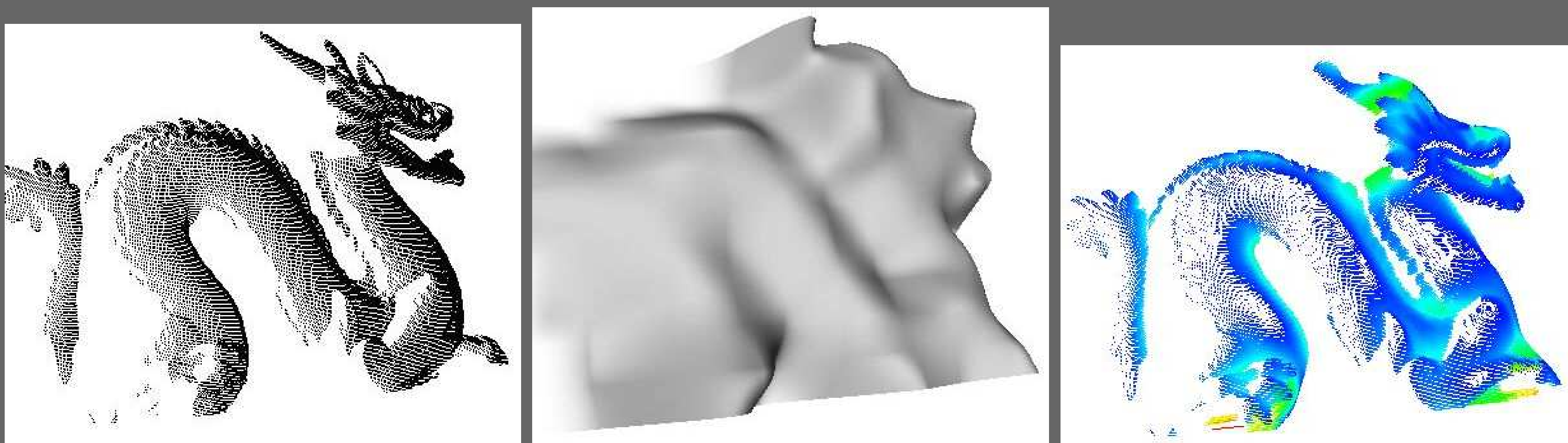
Bilinear cover





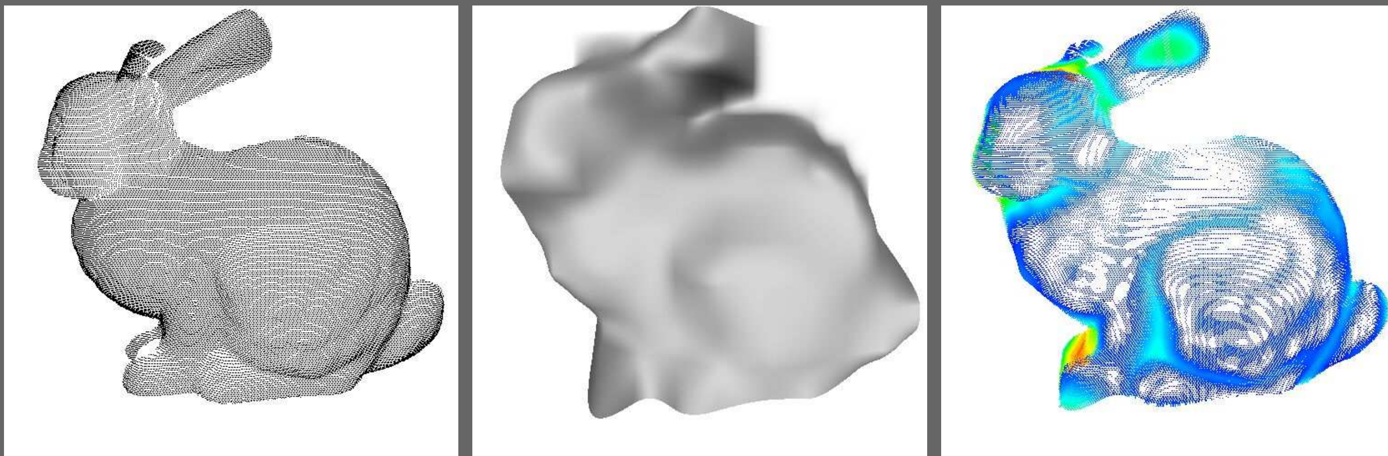
Range data fitting

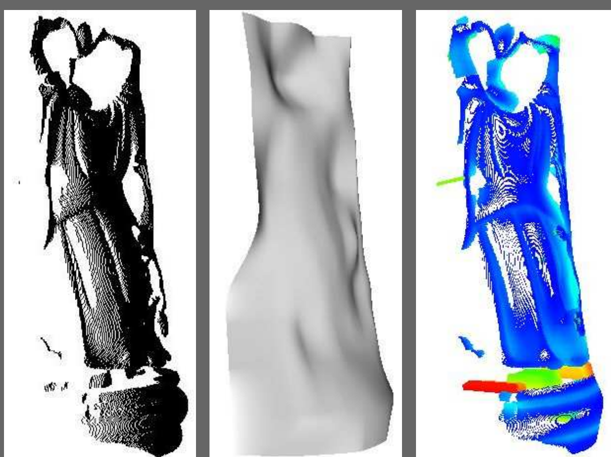
Left: Input range data (Dragon front view, 39526 points)
Middle: 10×10 bicubic spline
Right: offset distance from the range data.



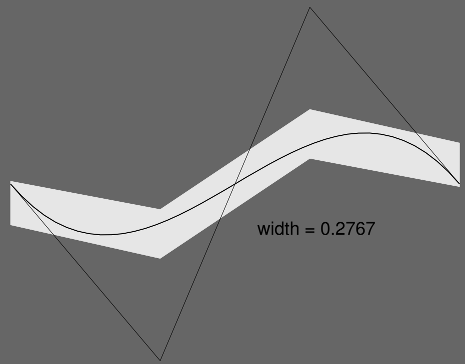
(orig) 4 min, (midpath fit + offset) 45 sec

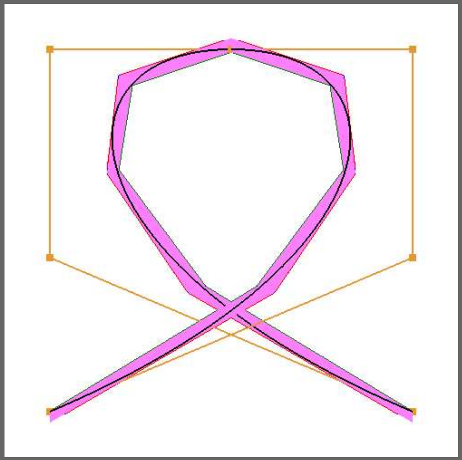
Range data fitting

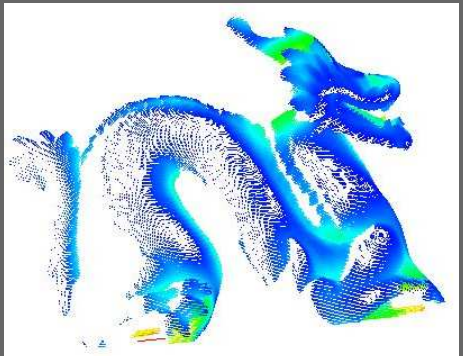




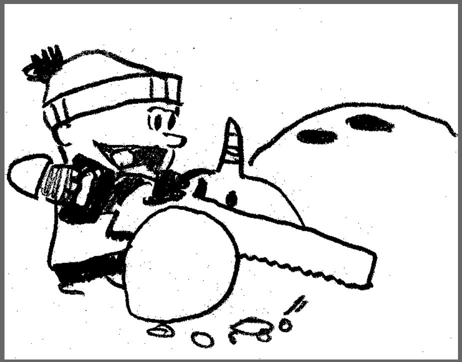
Questions?







(Nairn, P, Lutterkort 1998; Reif) Sharp, quantitative bounds on the distance ...
(Lutterkort & P 2000): Optimized Refinable Enclosures of Multivariate Polynomial Pieces
(Myles & P 2003): Threading splines through 3D channels
(P & Wu 2003): SLEVEs for planar spline curves



.