Smooth Shape-Aware Functions with Controlled Extrema

Alec Jacobson[1]
Tino Weinkauf[2]
Olga Sorkine[1]

[1]ETH Zurich
[2]MPI Saarbrücken

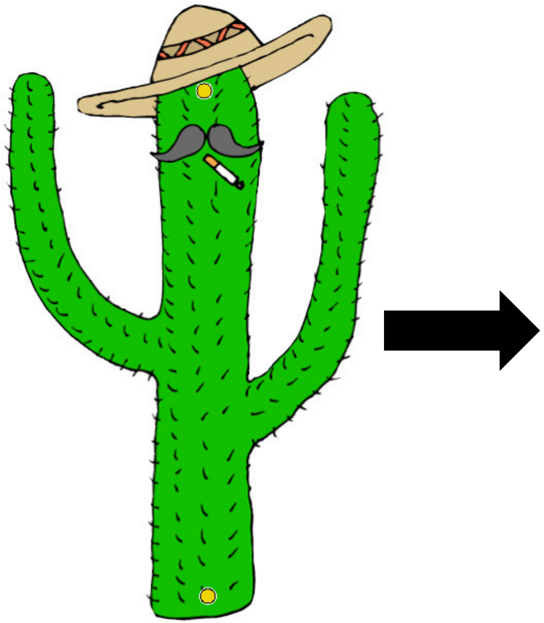
INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

August 9, 2012

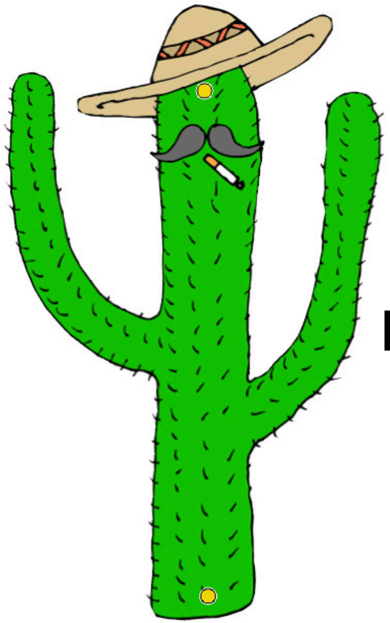
Real-time deformation relies on smooth, shape-aware functions

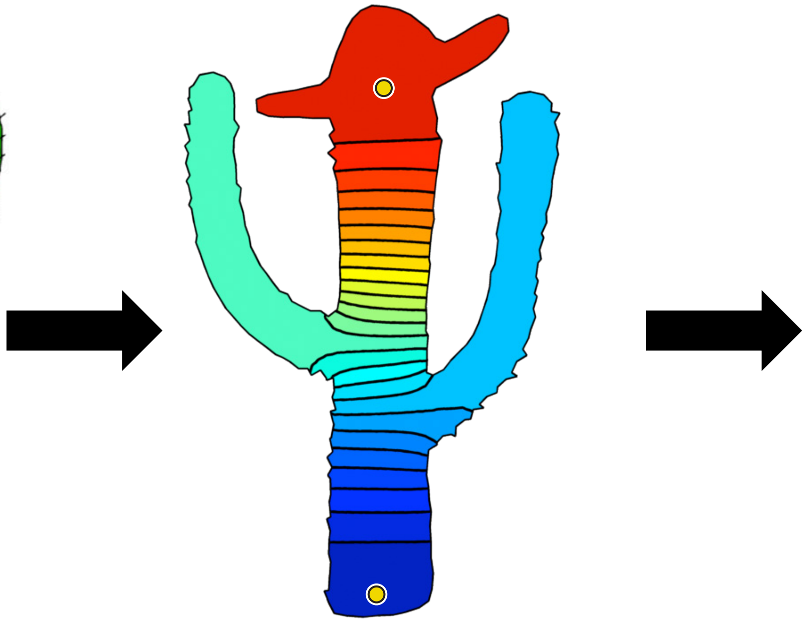
input shape + handles



Real-time deformation relies on smooth, shape-aware functions

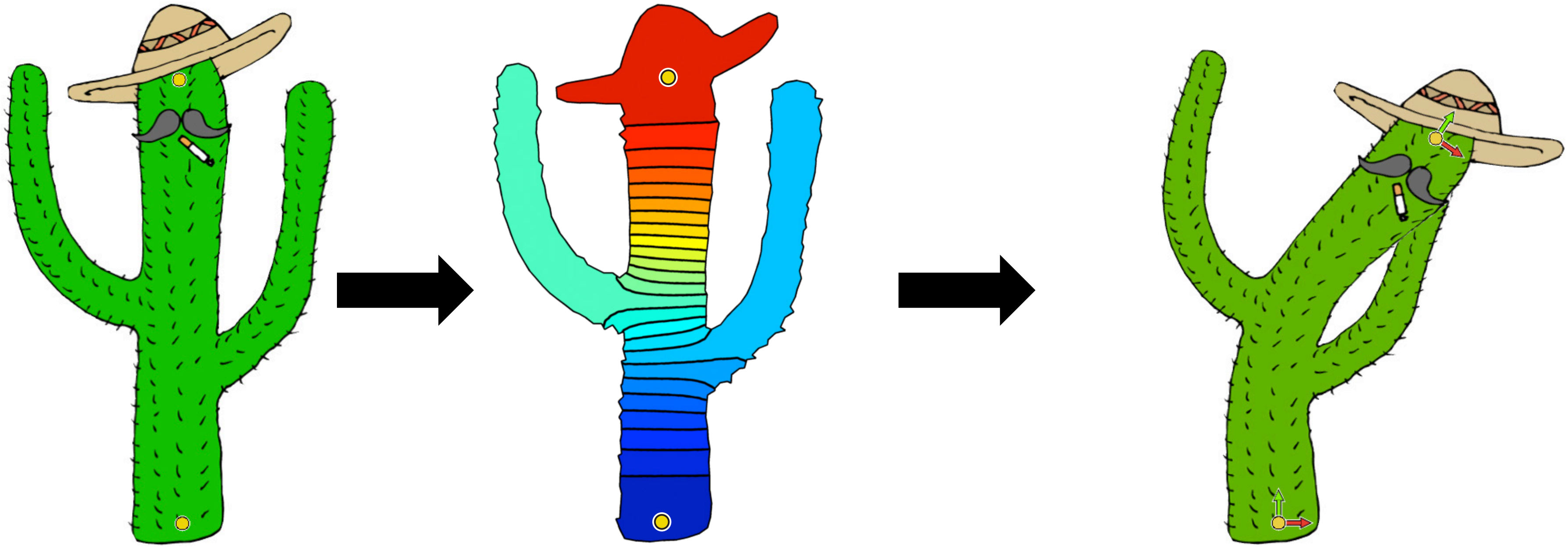
precompute weight functions





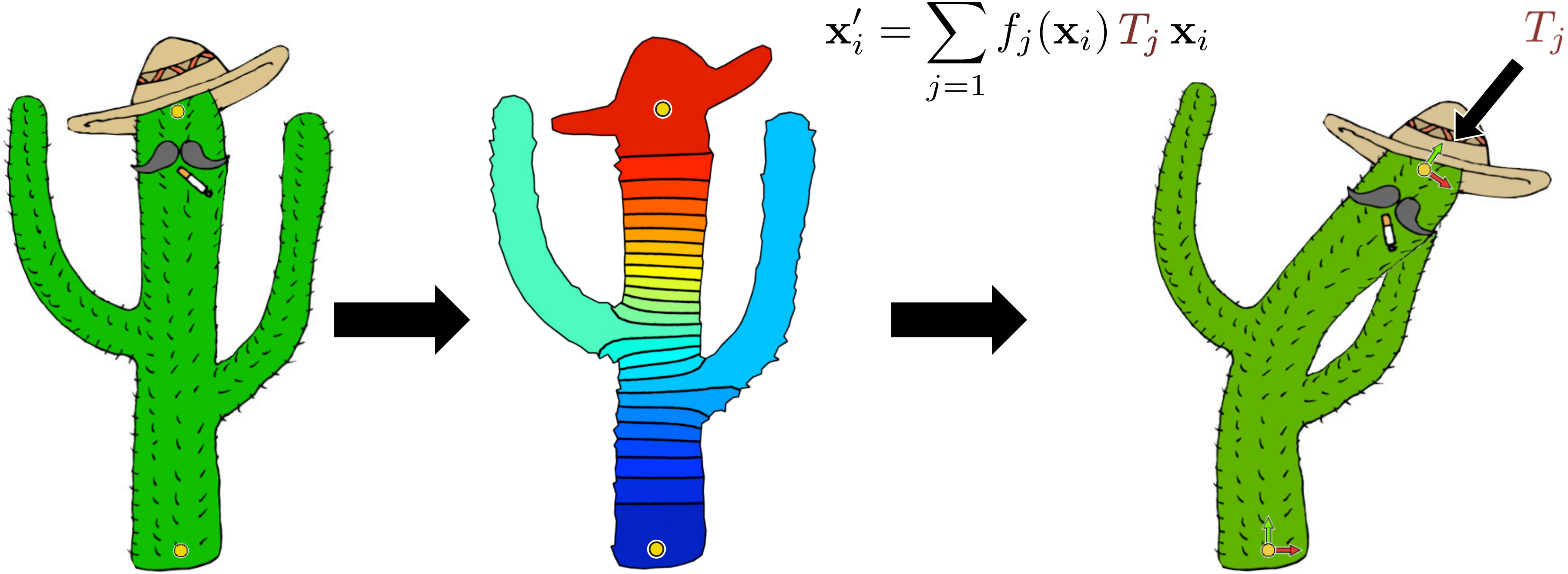
Real-time deformation relies on smooth, shape-aware functions

deform handles → deform shape

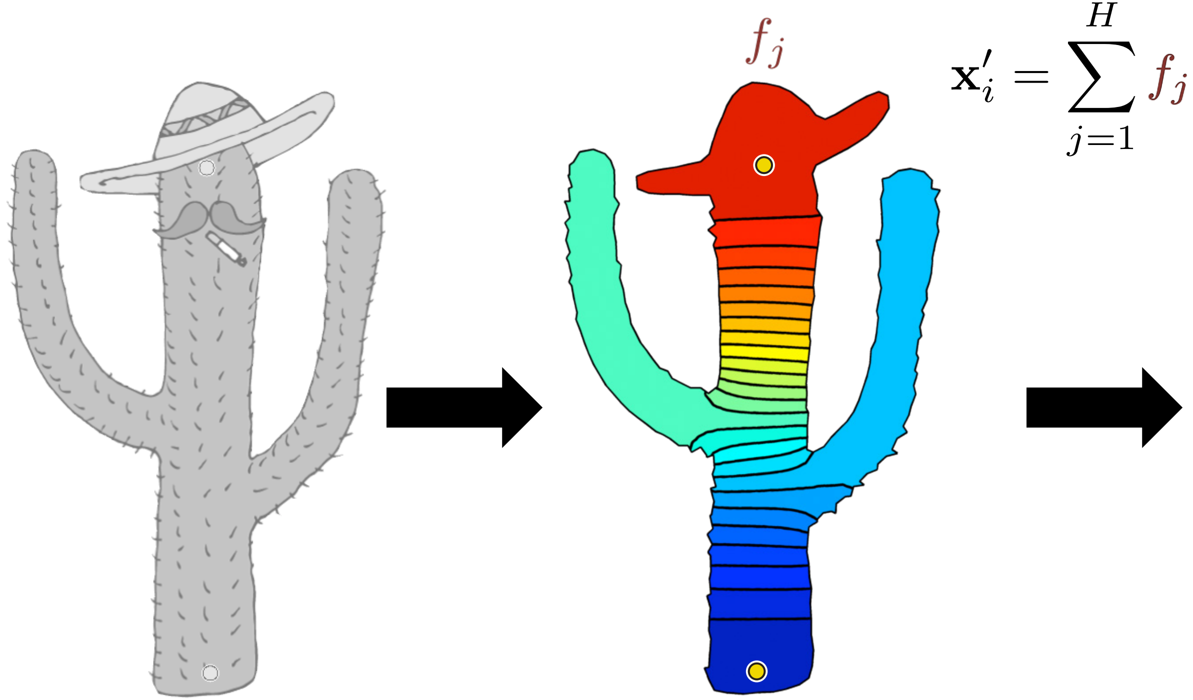


Real-time deformation relies on smooth, shape-aware functions

$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$



Real-time deformation relies on smooth, shape-aware functions

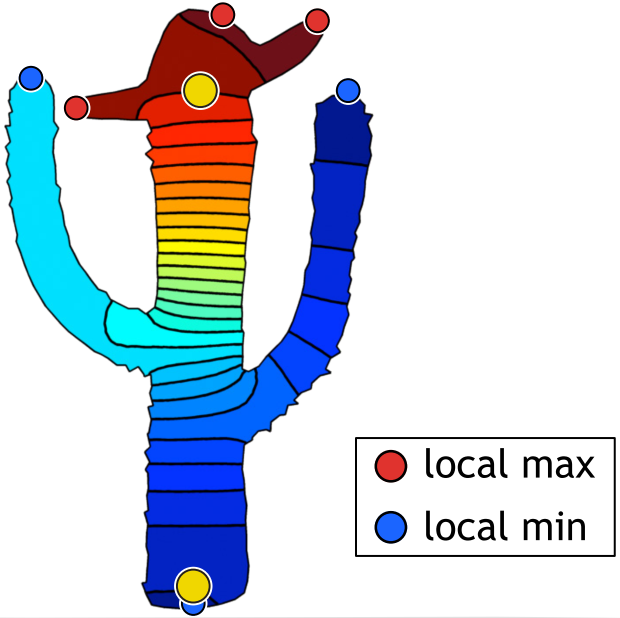


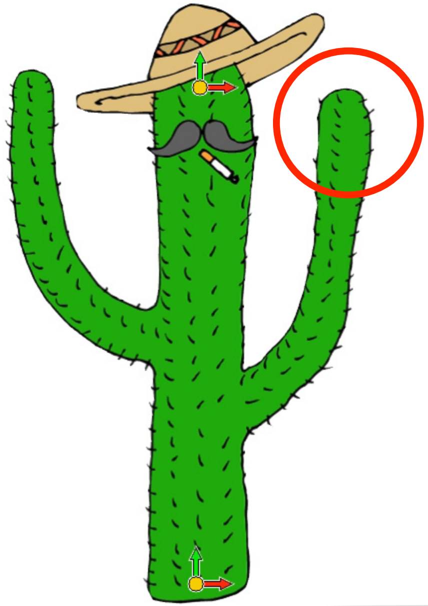
$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

Spurious extrema cause distracting artifacts

unconstrained Δ^2
[Botsch & Kobbelt 2004]

$$\mathbf{x}_i' = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

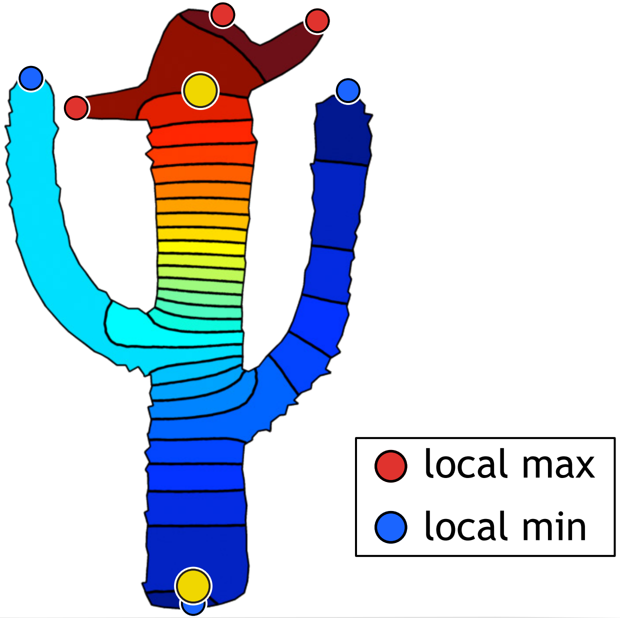


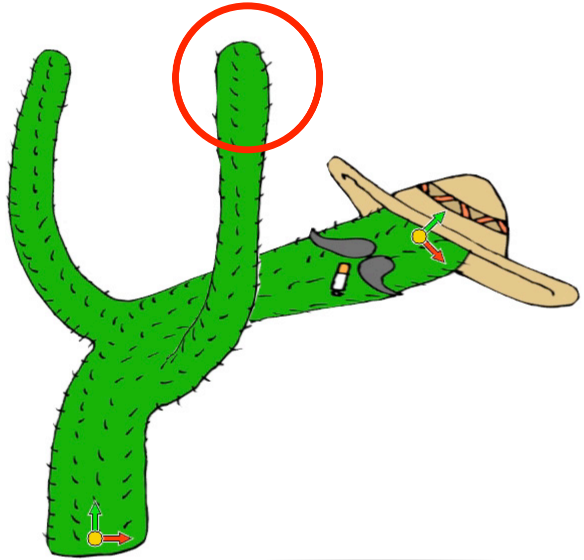


Spurious extrema cause distracting artifacts

unconstrained Δ^2
[Botsch & Kobbelt 2004]

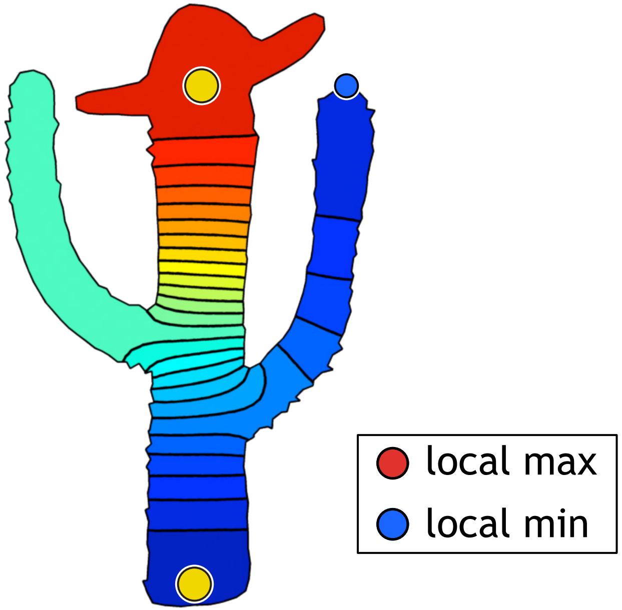
$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$



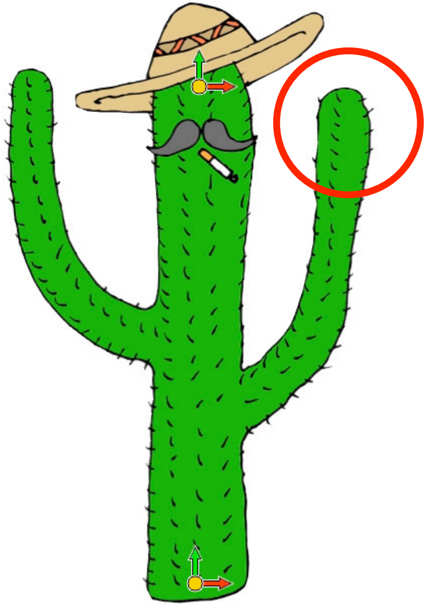


Bounds help, but don't solve problem

bounded Δ^2
[Jacobson et al. 2011]



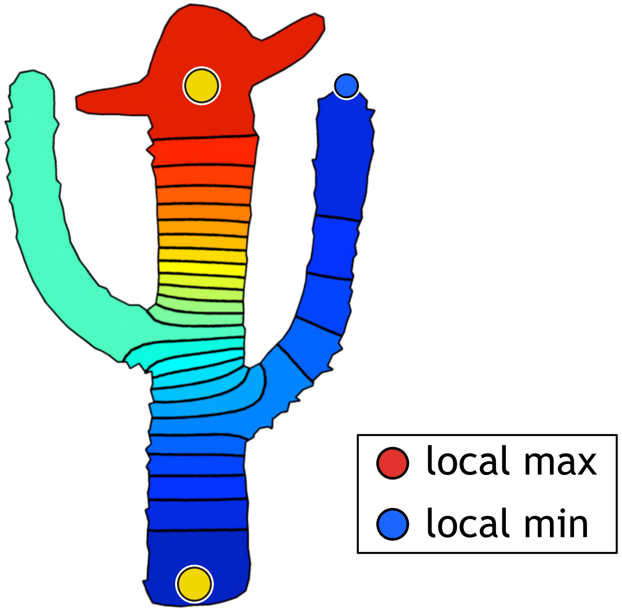
$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

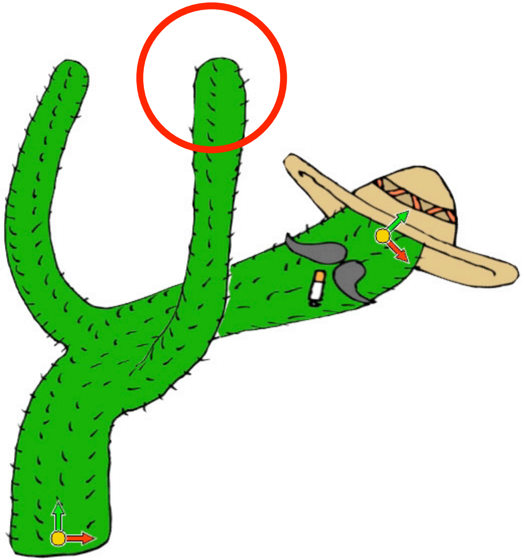


Bounds help, but don't solve problem

bounded Δ^2
[Jacobson et al. 2011]

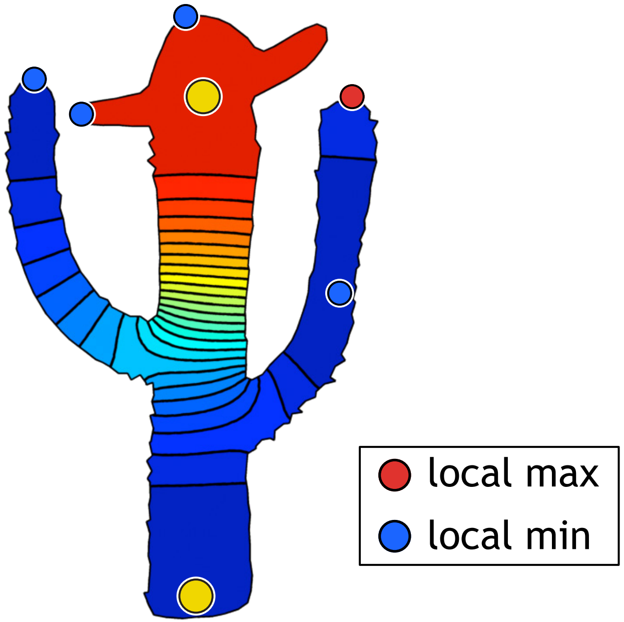
$$\mathbf{x}_i' = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$



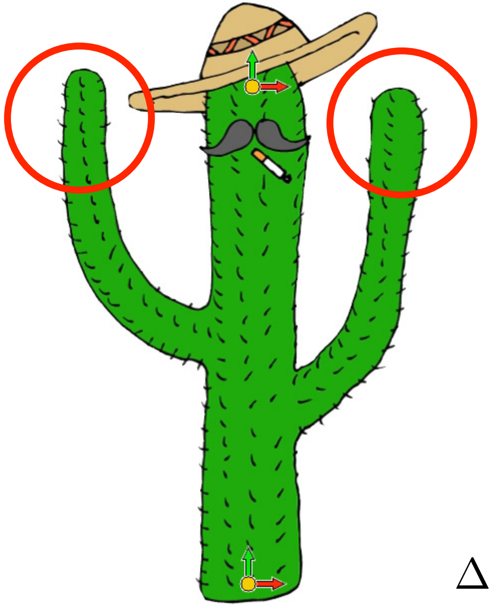


Gets worse with higher-order smoothness

bounded Δ^4
[Jacobson et al. 2011]



$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

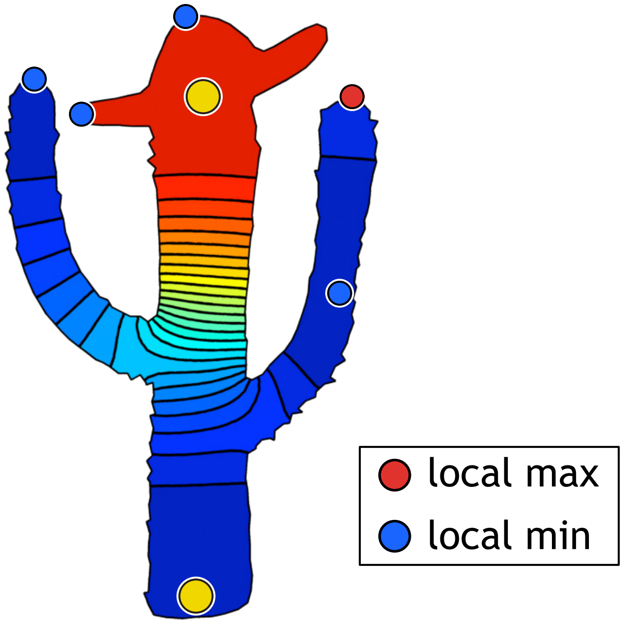


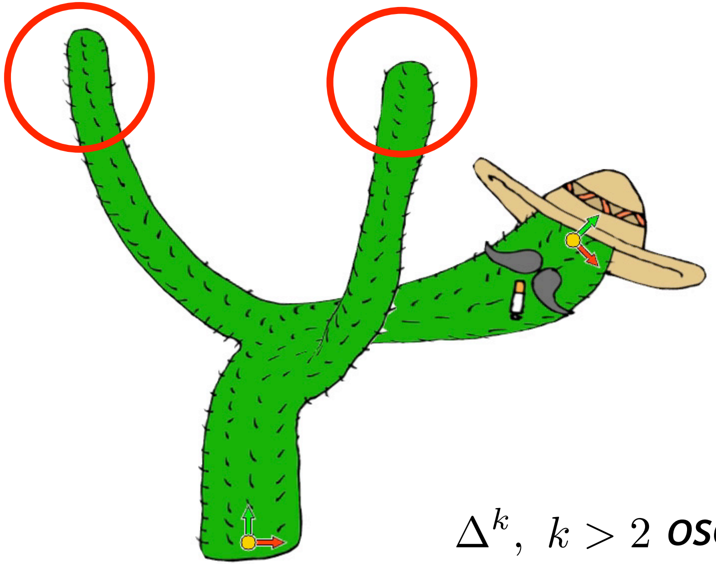
$\Delta^k,\ k > 2$ *oscillate too much*

Gets worse with higher-order smoothness

bounded Δ^4
[Jacobson et al. 2011]

$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

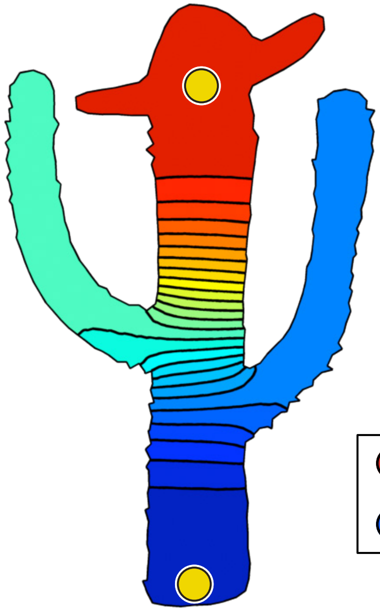




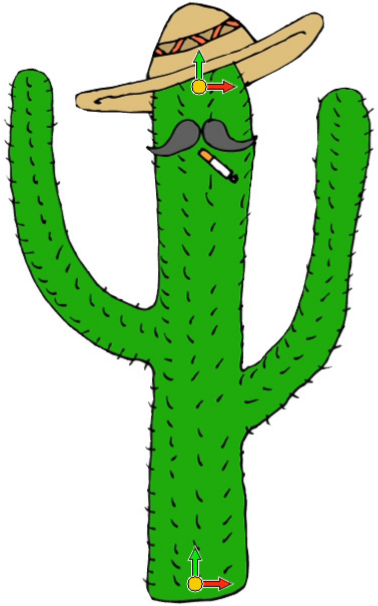
$\Delta^k,\ k > 2$ *oscillate too much*

We *explicitly* prohibit spurious extrema

our Δ^4

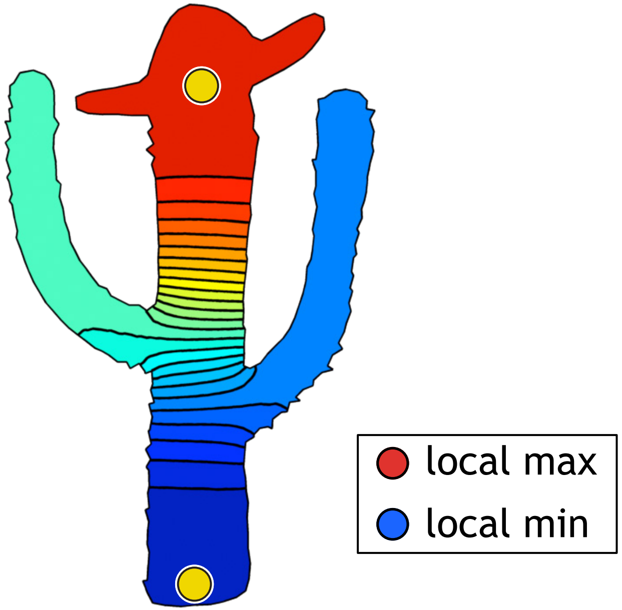


$$\mathbf{x}_i' = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

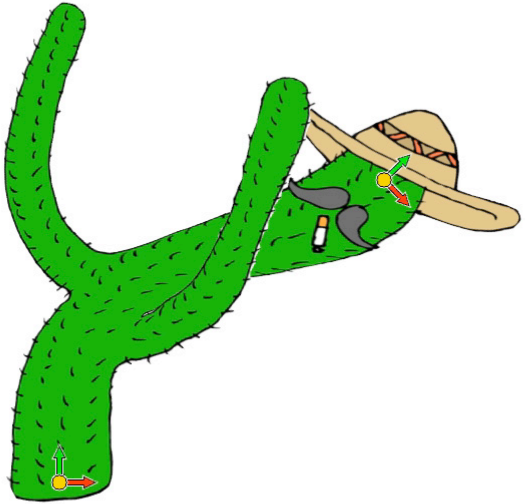


We *explicitly* prohibit spurious extrema

our Δ^4



$$\mathbf{x}'_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$



Same functions used for color interpolation

$$\mathbf{x}_i' = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\, T_j\, \mathbf{x}_i$$

igl
ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Same functions used for color interpolation

$$\mathbf{c}_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\,\mathbf{c}_j$$

Same functions used for color interpolation

unconstrained Δ^2
[Finch et al. 2011]

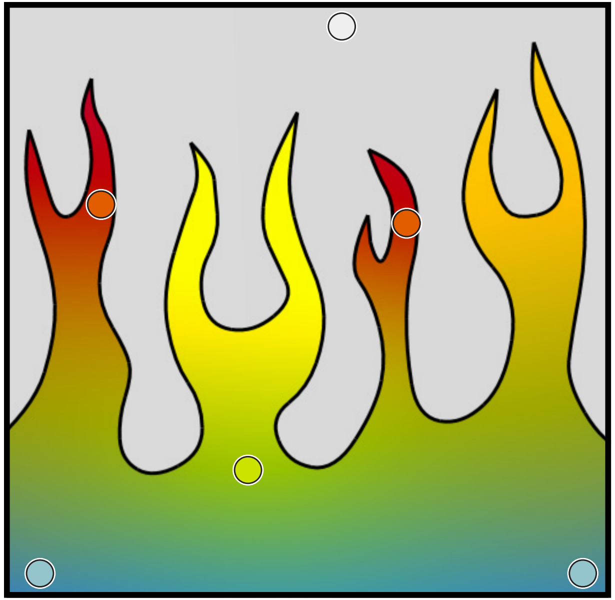
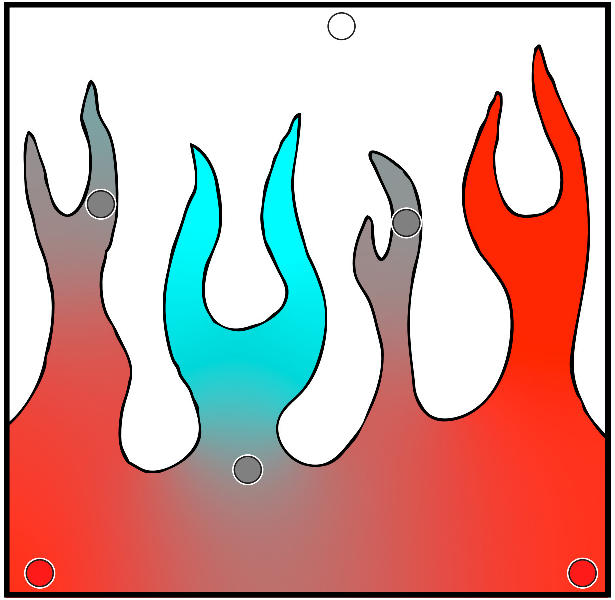


Image courtesy Mark Finch

$$\mathbf{c}_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\mathbf{c}_j$$

Same functions used for color interpolation

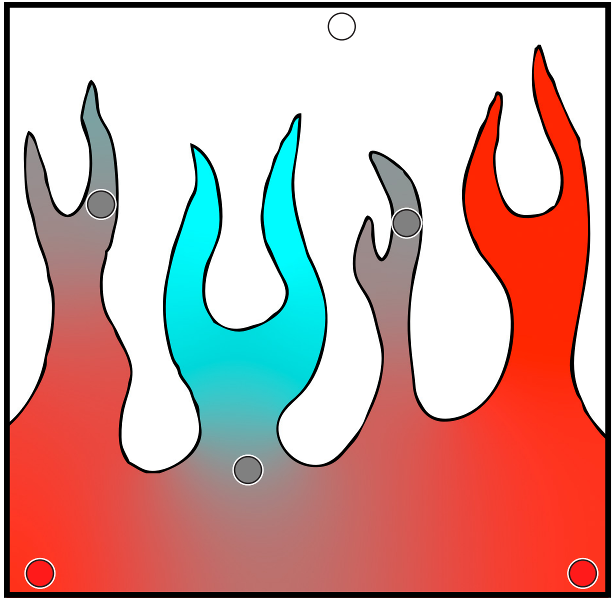
unconstrained Δ^2
[Finch et al. 2011]



$$\mathbf{c}_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\mathbf{c}_j$$

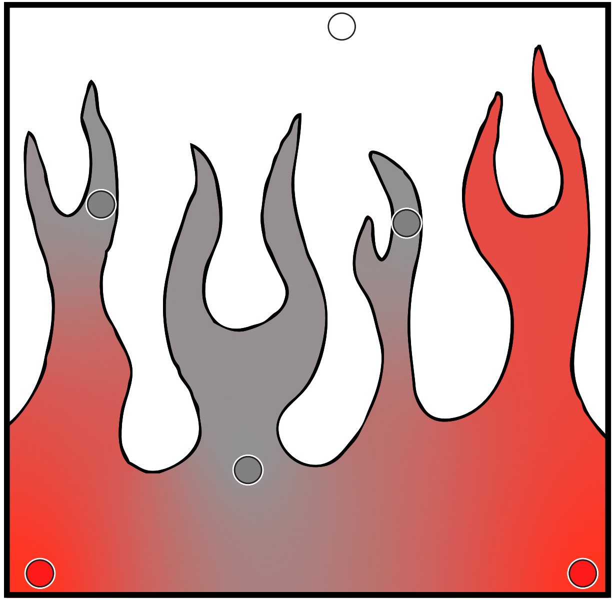
Same functions used for color interpolation

unconstrained Δ^2
[Finch et al. 2011]



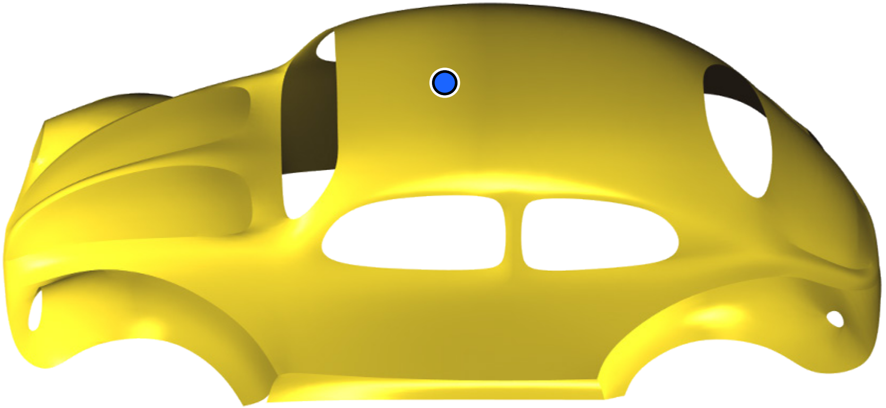
$$\mathbf{c}_i = \sum_{j=1}^{H} f_j(\mathbf{x}_i)\mathbf{c}_j$$

Our Δ^2



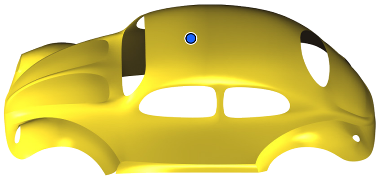
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

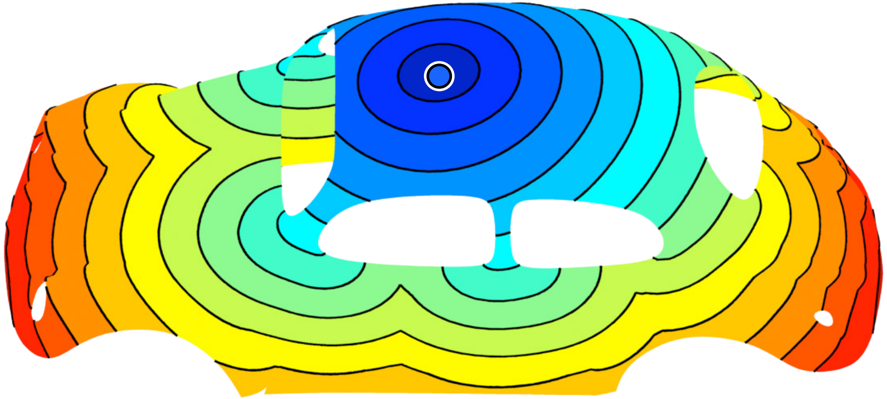
Want same control when smoothing data



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Want same control when smoothing data

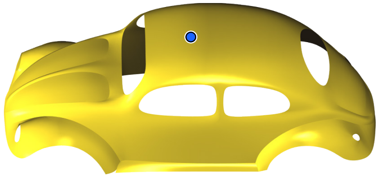


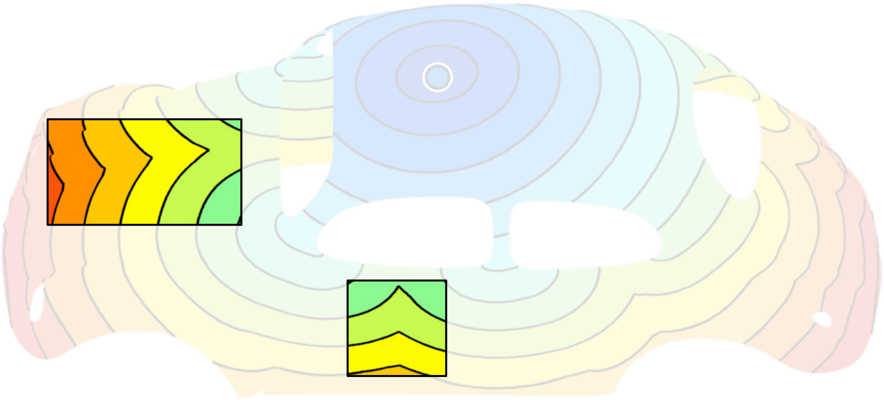


Exact, but sharp geodesic

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

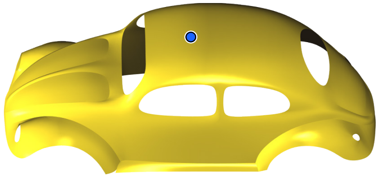
Want same control when smoothing data

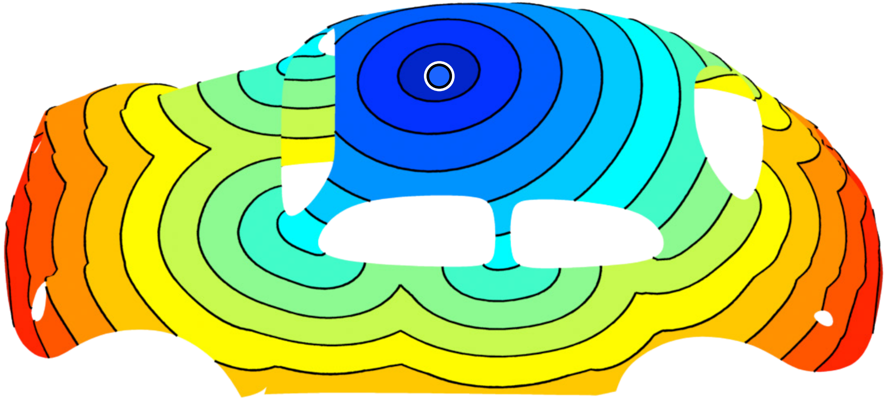




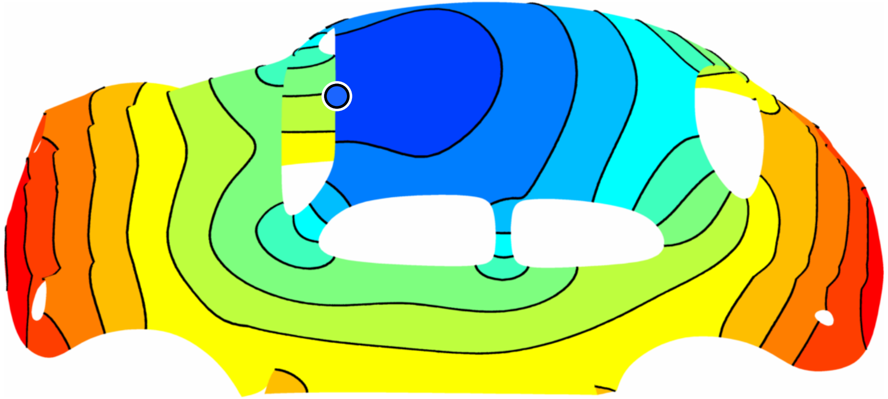
Exact, but sharp geodesic

Want same control when smoothing data



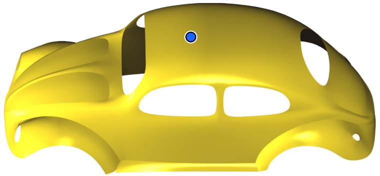


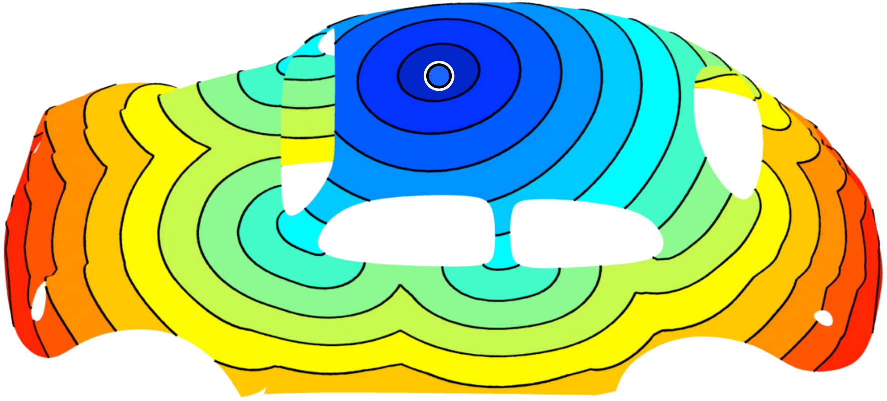
Exact, but sharp geodesic



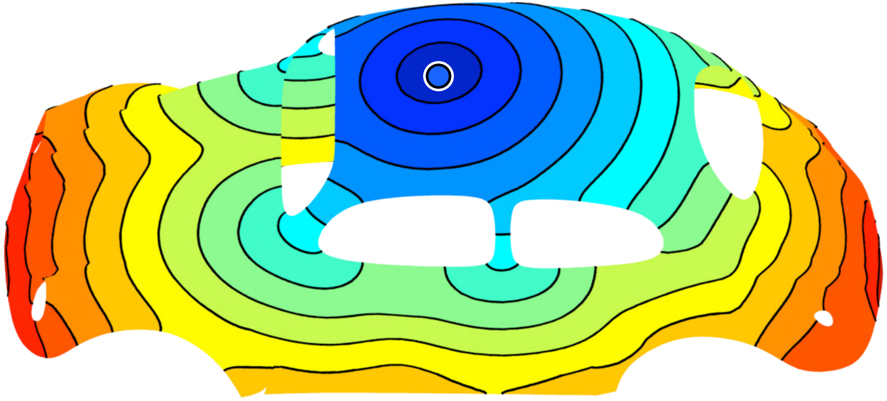
Smooth, but extrema are lost

Want same control when smoothing data





Exact, but sharp geodesic



Smooth *and* maintain extrema

Ideal discrete problem is intractable

$$\arg\min_{f} \; E(f)$$

Interpolation functions:

$$E_L(f) = \int_{\mathcal{M}} \|\nabla^k f\|^2 dV, \quad k = 2, 3, \ldots$$

Ideal discrete problem is intractable

$$\arg\min_{f} \; E(f)$$

Data smoothing:

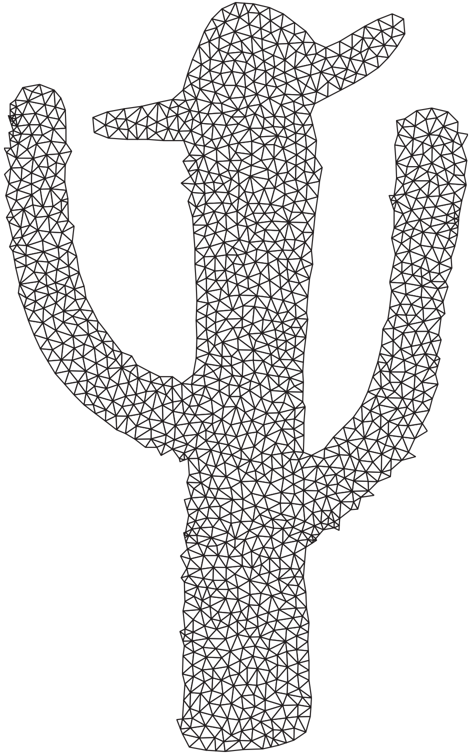
$$E_L(f) = \int_{\mathcal{M}} \|\nabla^k f\|^2 dV, \quad k = 2, 3, \ldots$$

$$E_D(f) = \sum_{i \in \mathcal{M}} \|h_i - f_i\|^2$$

$$E(f) = \gamma_L E_L(f) + \gamma_D E_D(f)$$

Ideal discrete problem is intractable

$$\arg\min_{f} \; E(f)$$

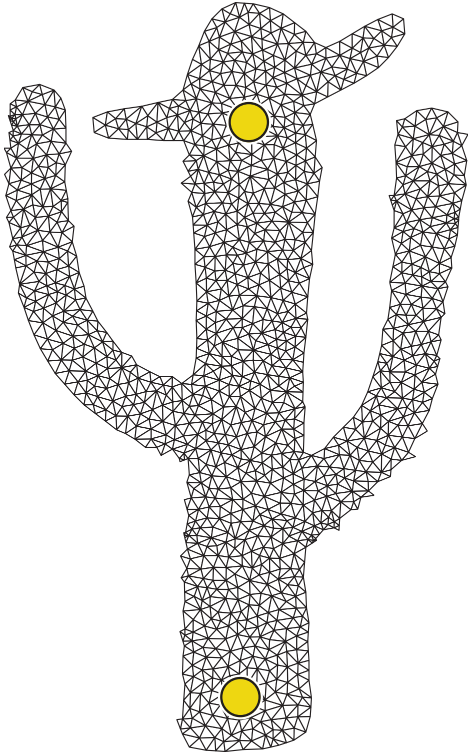


Ideal discrete problem is intractable

$$\arg\min_f \; E(f)$$

$$\text{s.t.} \quad f_{\max} = known$$

$$f_{\min} = known$$



Ideal discrete problem is intractable

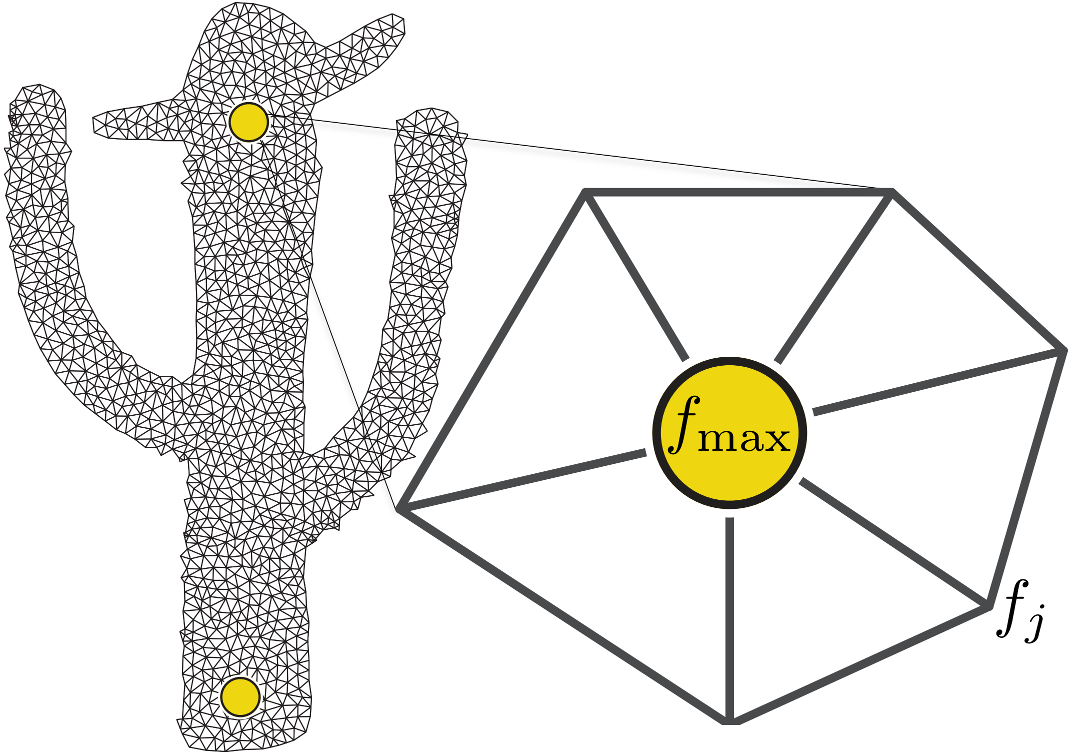
$$\arg\min_{f} \; E(f)$$

s.t. $f_{\max} = known$

$f_{\min} = known$

linear $f_j < f_{\max}$

$f_j > f_{\min}$



Ideal discrete problem is intractable

$$\arg\min_{f} \; E(f)$$

s.t.

$$f_{\max} = known$$

$$f_{\min} = known$$

linear

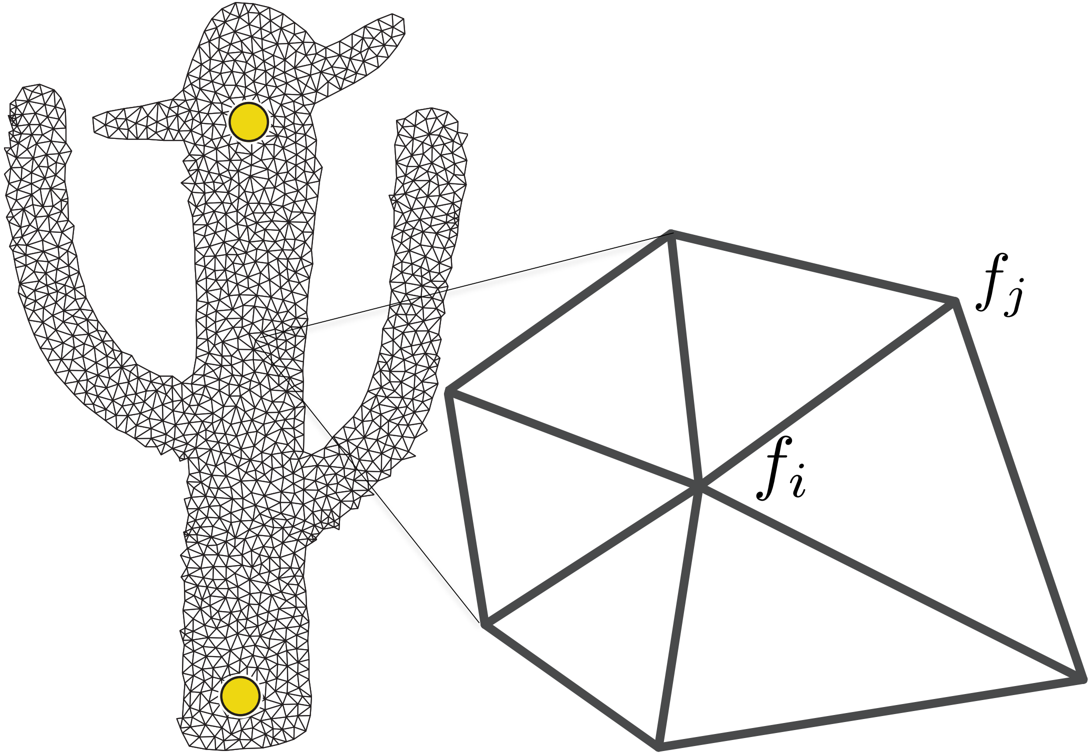
$$f_j < f_{\max}$$

$$f_j > f_{\min}$$

nonlinear

$$f_i > \min_{j \in \mathcal{N}(i)} f_j$$

$$f_i < \max_{j \in \mathcal{N}(i)} f_j$$



Assume we have a feasible solution

$$\arg\min_f \; E(f)$$

$$\text{s.t.} \quad f_{\max} = known$$

$$f_{\min} = known$$

linear

$$f_j < f_{\max}$$

$$f_j > f_{\min}$$

nonlinear

$$f_i > \min_{j \in \mathcal{N}(i)} f_j$$

$$f_i < \max_{j \in \mathcal{N}(i)} f_j$$

"Representative function" u

$$u_j < u_{\max}$$

$$u_j > u_{\min}$$

handles

$$u_i > \min_{j \in \mathcal{N}(i)} u_j$$

$$u_i < \max_{j \in \mathcal{N}(i)} u_j$$

interior

Assume we have a feasible solution

"Representative function" $\mathcal{U}$

handles	$u_j < u_{\max}$ $u_j > u_{\min}$
interior	$u_i > \min\limits_{j \in \mathcal{N}(i)} u_j$ $u_i < \max\limits_{j \in \mathcal{N}(i)} u_j$

Copy "monotonicity" of representative

$$\arg\min_f \; E(f)$$

$$\text{s.t.} \quad f_{\max} = known$$

$$f_{\min} = known$$

$$(f_i - f_j)(u_i - u_j) > 0 \qquad \text{linear} \qquad \forall (i,j) \in \mathcal{E}$$

At least one edge in either direction per vertex

Rewrite as conic optimization

QP

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{F}\,\mathbf{f}\|^2 + \mathbf{c}^\mathsf{T}\mathbf{f} + \text{const} \\ \text{subject to} \quad & \mathbf{A}_{leq}^\mathsf{T}\mathbf{f} \leq \mathbf{b}_{leq}, \\ & \mathbf{f} \leq \mathbf{u}_f, \quad \mathbf{f} \geq \mathbf{l}_f \end{aligned}$$

Conic

$$\begin{aligned} \underset{\left[\begin{array}{ccc}\mathbf{f}^\mathsf{T} & \mathbf{t}^\mathsf{T} & v\end{array}\right]}{\text{minimize}} \quad & \left[\begin{array}{ccc}\mathbf{c}^\mathsf{T} & 0 & 1\end{array}\right] \left[\begin{array}{c}\mathbf{f} \\ \mathbf{t} \\ v\end{array}\right] + \text{const} \\ \text{subject to} \quad & \left[\begin{array}{ccc}\mathbf{F} & -\mathbf{I} & 0 \\ \mathbf{A}_{leq}^\mathsf{T} & 0 & 0\end{array}\right] \left[\begin{array}{c}\mathbf{f} \\ \mathbf{t} \\ v\end{array}\right] \geq \left[\begin{array}{c}0 \\ -\infty\end{array}\right] \\ & \left[\begin{array}{ccc}\mathbf{F} & -\mathbf{I} & 0 \\ \mathbf{A}_{leq}^\mathsf{T} & 0 & 0\end{array}\right] \left[\begin{array}{c}\mathbf{f} \\ \mathbf{t} \\ v\end{array}\right] \leq \left[\begin{array}{c}0 \\ \mathbf{b}_{leq}\end{array}\right] \\ & \left[\begin{array}{c}\mathbf{f} \\ \mathbf{t} \\ v\end{array}\right] \leq \left[\begin{array}{c}\mathbf{u}_f \\ \infty \\ \infty\end{array}\right] \\ & \left[\begin{array}{c}\mathbf{f} \\ \mathbf{t} \\ v\end{array}\right] \geq \left[\begin{array}{c}\mathbf{l}_f \\ -\infty \\ 0\end{array}\right] \\ & 2v \geq \sum_i t_i^2 \end{aligned}$$

Optimize with MOSEK

We always have harmonic representative

$$\arg\min_u \ \frac{1}{2}\int_\Omega \|\nabla u\|^2 dV$$

We always have harmonic representative

$$\arg\min_u \ \frac{1}{2}\int_\Omega \|\nabla u\|^2 dV$$

$$\text{s.t.} \quad u_{\max} = 1$$

We always have harmonic representative

$$\arg\min_{u} \ \frac{1}{2}\int_{\Omega} \|\nabla u\|^2 dV$$

$$\text{s.t.} \qquad u_{\max} = 1$$

$$\text{s.t.} \qquad u_{\min} = 0$$

We always have harmonic representative

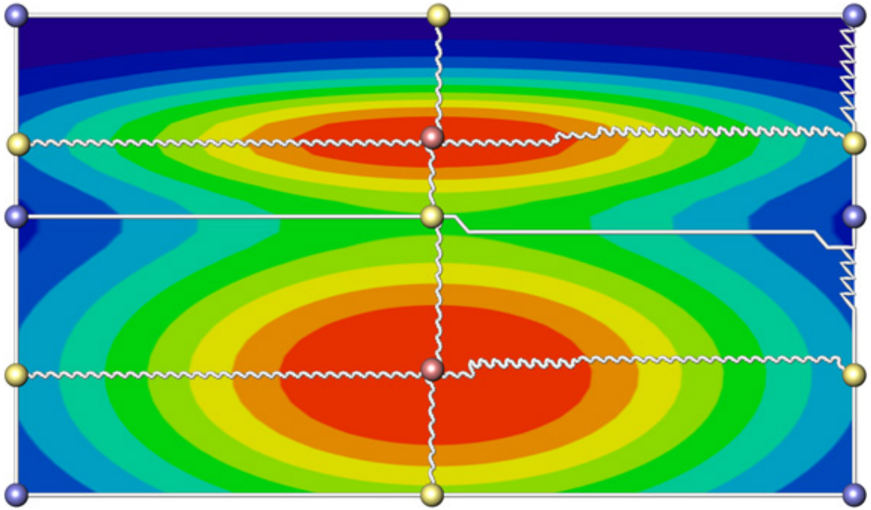
$$\mathop{\arg\min}_{u} \ \frac{1}{2}\int_{\Omega} \|\nabla u\|^2 dV$$

$$\text{s.t.} \quad u_{\max} = 1$$

$$\text{s.t.} \quad u_{\min} = 0$$

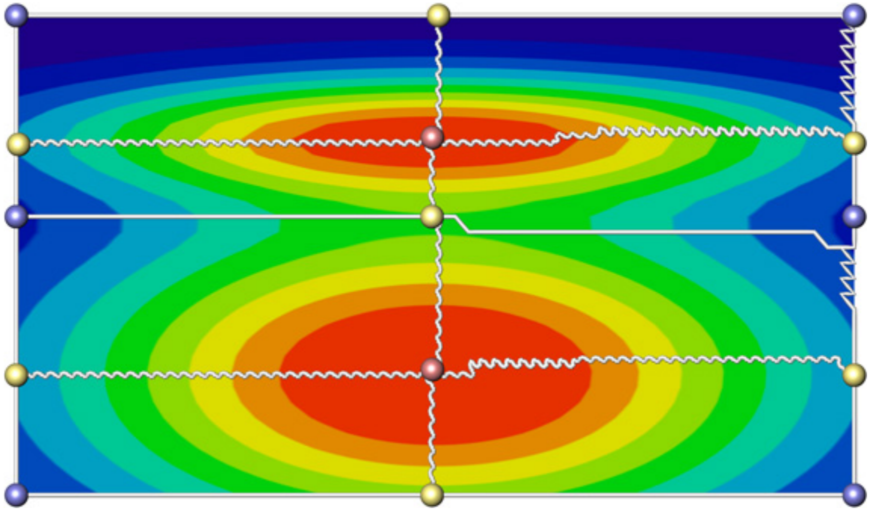
Works well when no input function exists

Data energy may fight harmonic representative

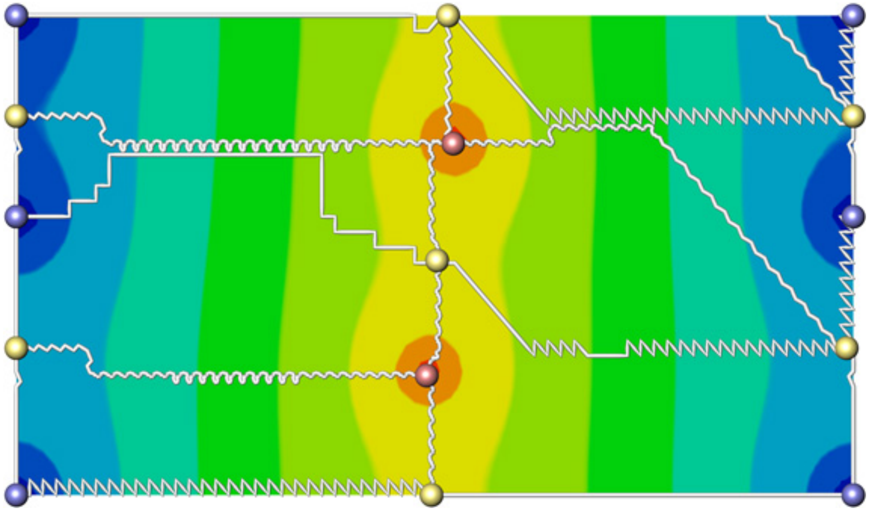


Anisotropic input data

Data energy may fight harmonic representative

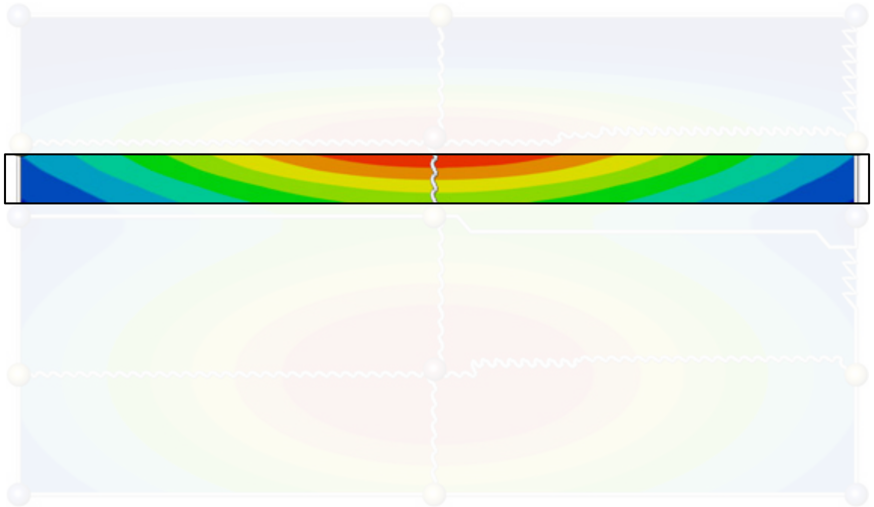


Anisotropic input data

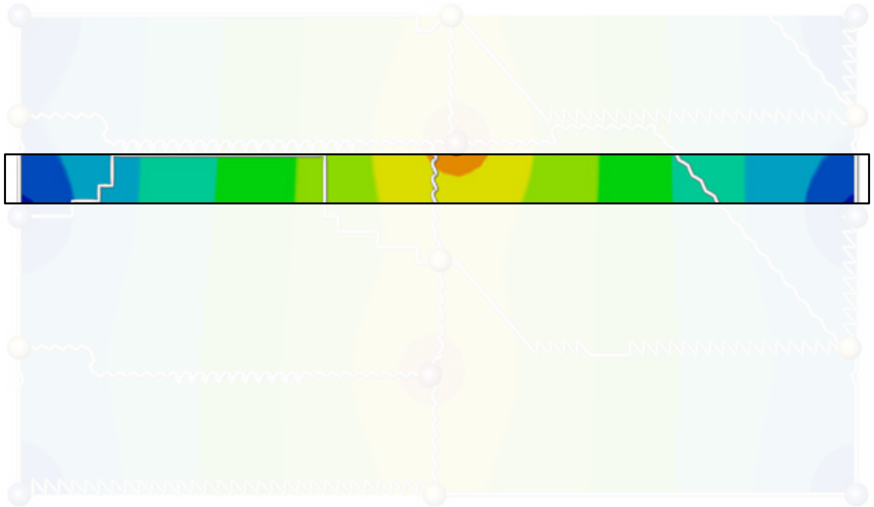


Harmonic representative

Data energy may fight harmonic representative

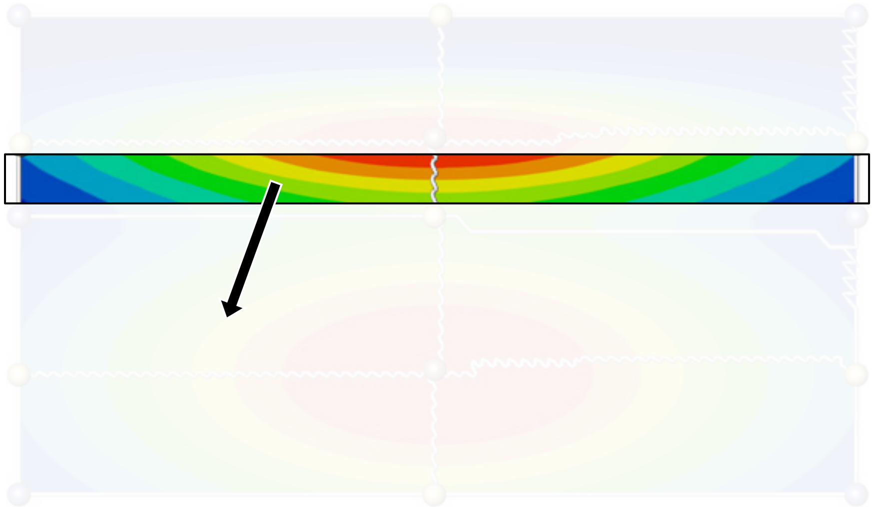


Anisotropic input data

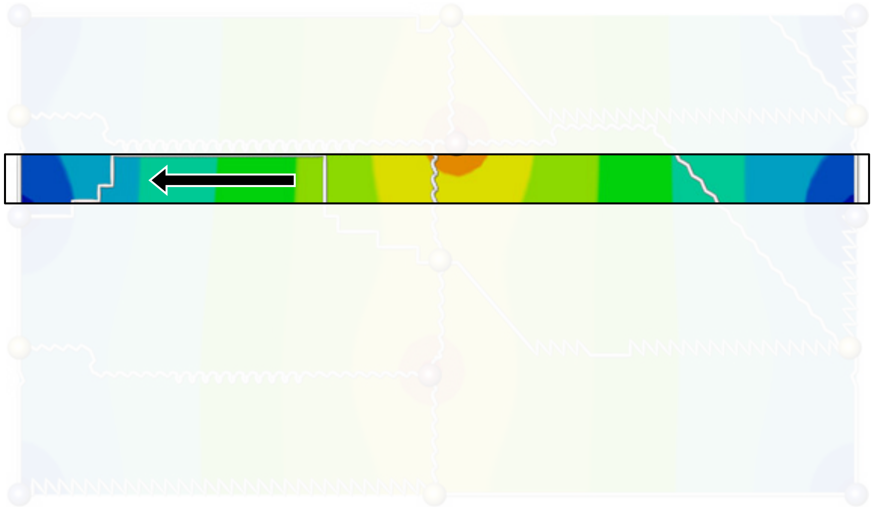


Harmonic representative

Data energy may fight harmonic representative

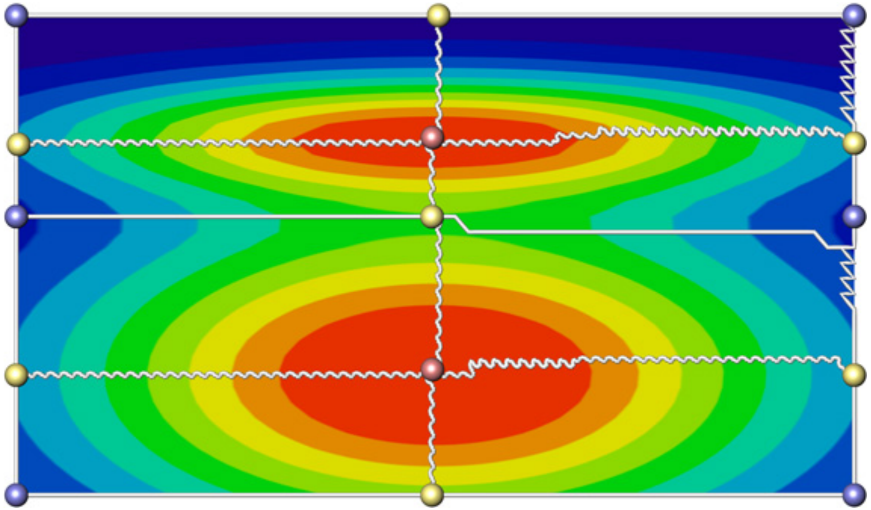


Anisotropic input data

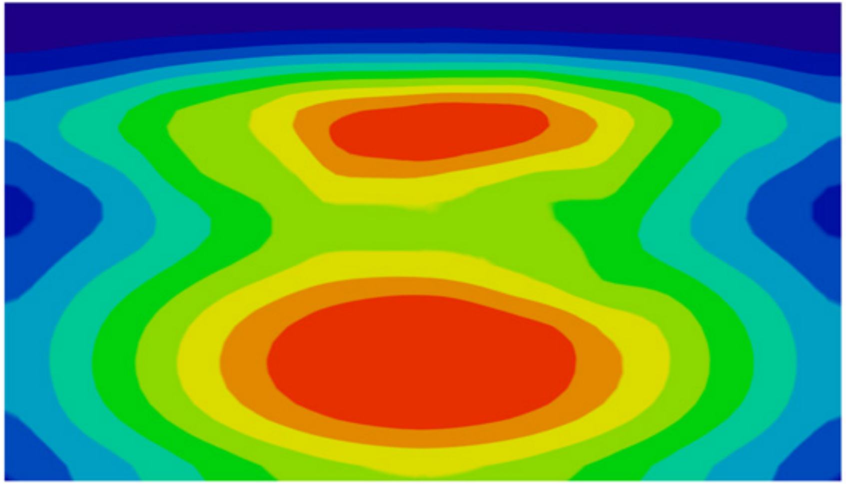


Harmonic representative

Data energy may fight harmonic representative

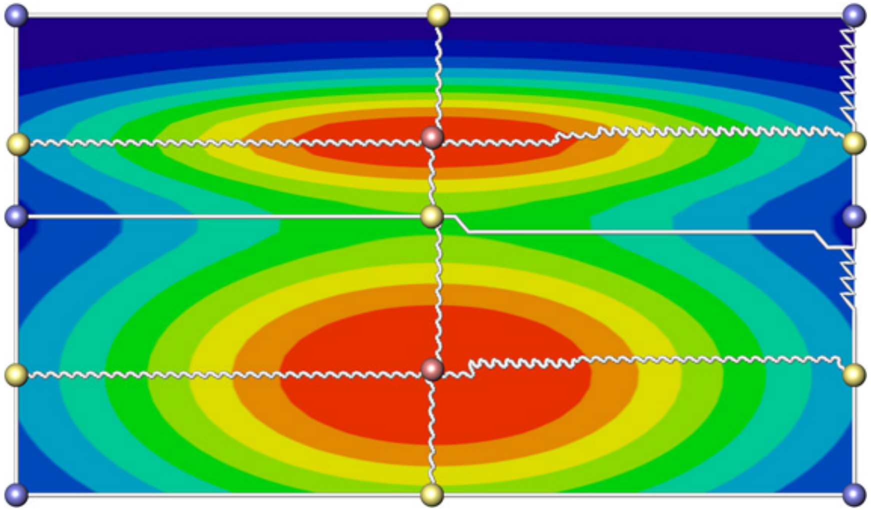


Anisotropic input data

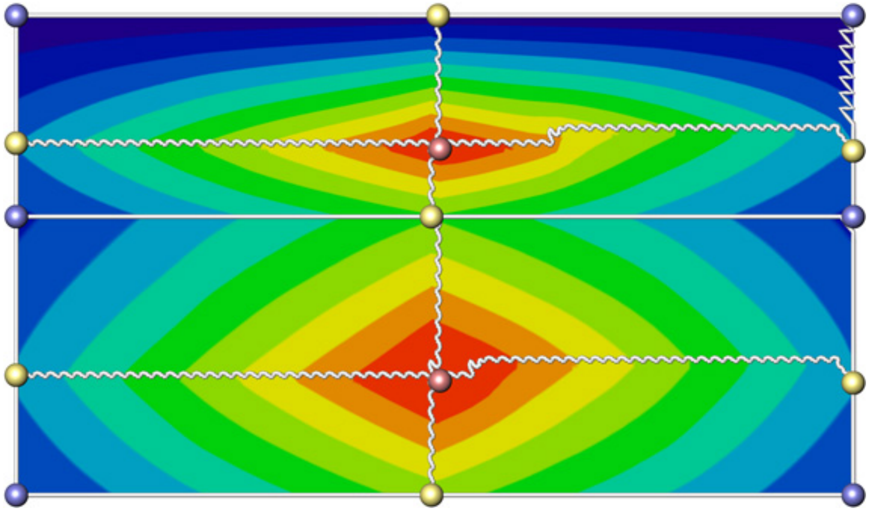


Resulting solution with large γ_D

If data exists, copy topology, too

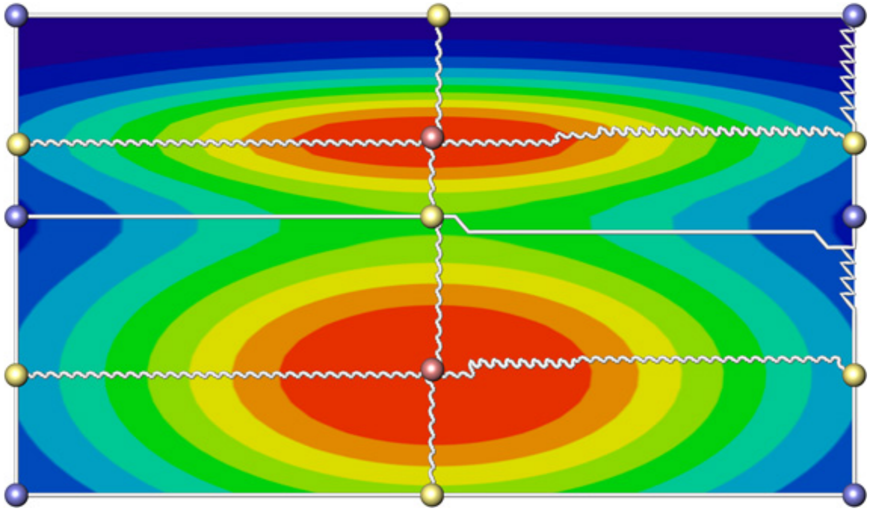


Anisotropic input data

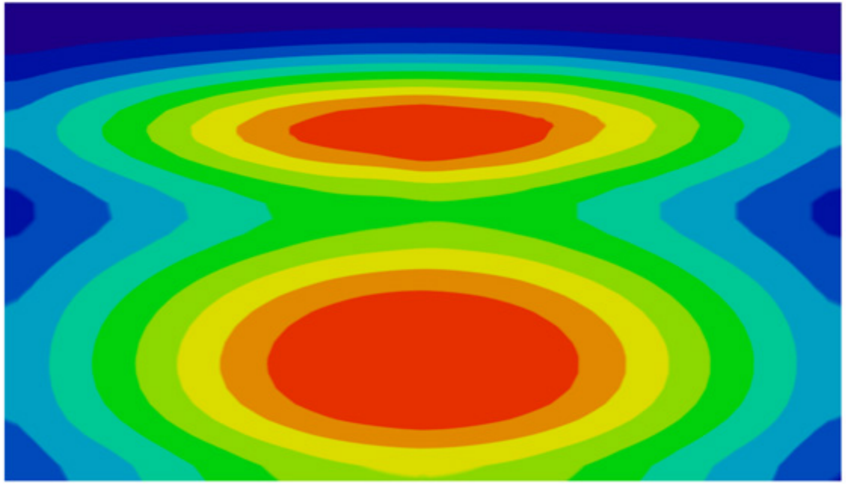


[Weinkauf et al. 2010] representative

If data exists, copy topology, too



Anisotropic input data



Resulting solution with large γ_D

Final algorithm is simple and efficient

- *Data smoothing*: topology-aware representative
  - Morse-smale + linear solve ~milliseconds

Final algorithm is simple and efficient

- *Data smoothing*: topology-aware representative
  - Morse-smale + linear solve ~milliseconds
- *Interpolation*: harmonic representative
  - Linear solve ~milliseconds

Final algorithm is simple and efficient

- *Data smoothing*: topology-aware representative
  - Morse-smale + linear solve ~milliseconds
- *Interpolation*: harmonic representative
  - Linear solve ~milliseconds
- Conic optimization
  - 2D ~milliseconds, 3D ~seconds

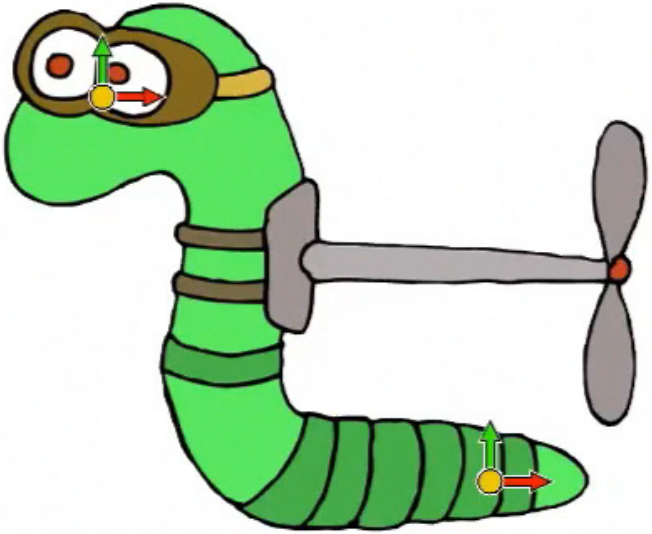
Final algorithm is simple and efficient

- *Data smoothing*: topology-aware representative
  - Morse-smale + linear solve ~milliseconds
- *Interpolation*: harmonic representative
  - Linear solve ~milliseconds
- Conic optimization
  - 2D ~milliseconds, 3D ~seconds

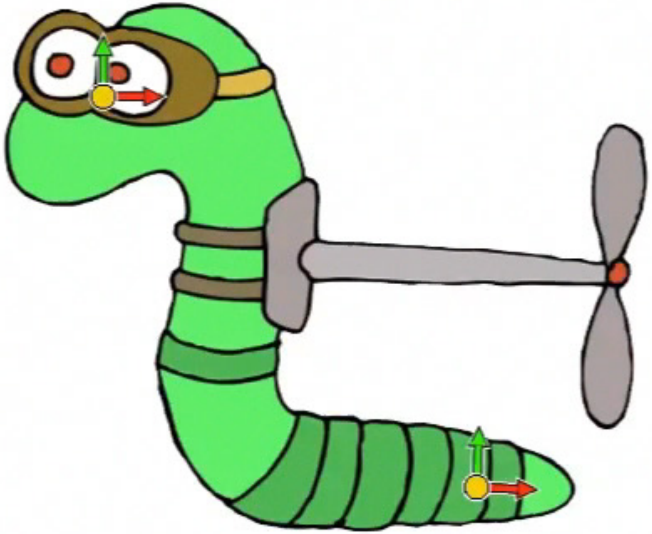
Interpolation: functions are **precomputed**

We preserve troublesome appendages

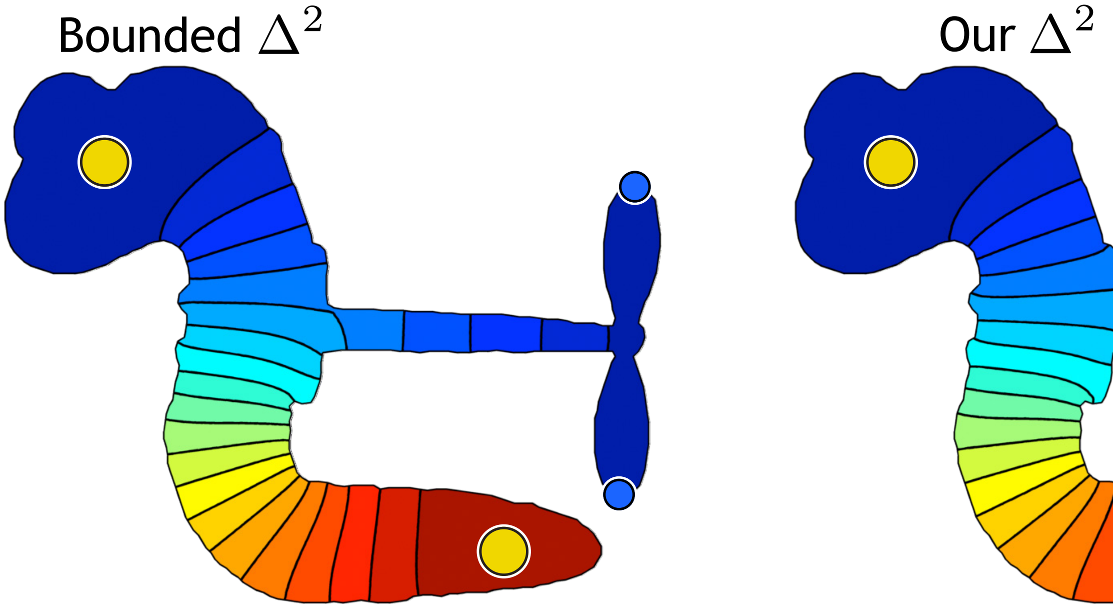
Bounded Δ^2



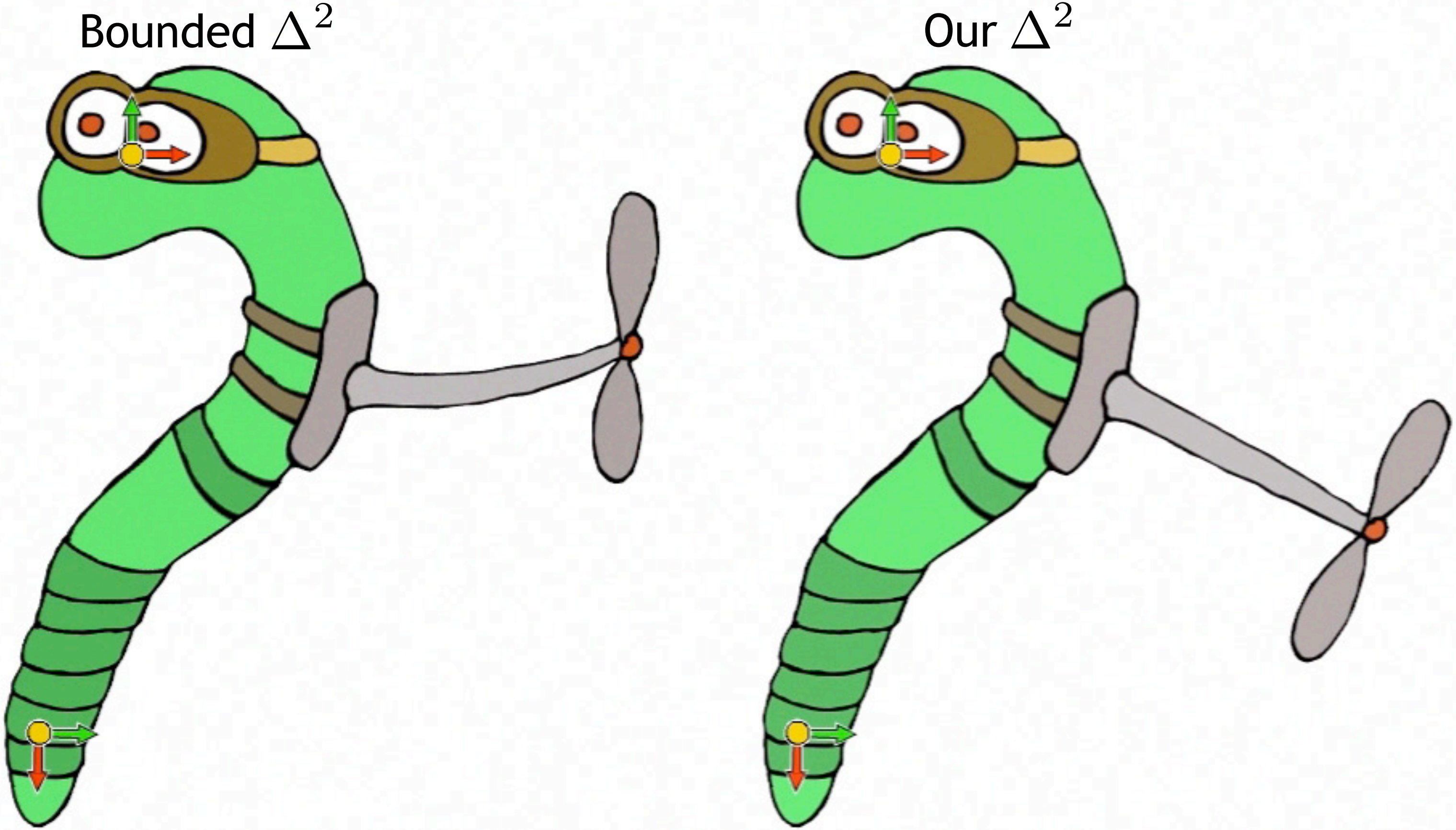
Our Δ^2



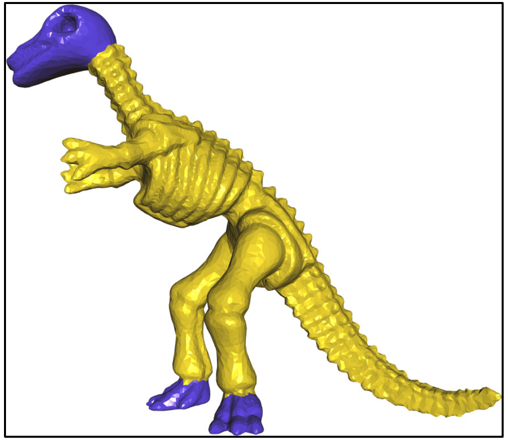
We preserve troublesome appendages

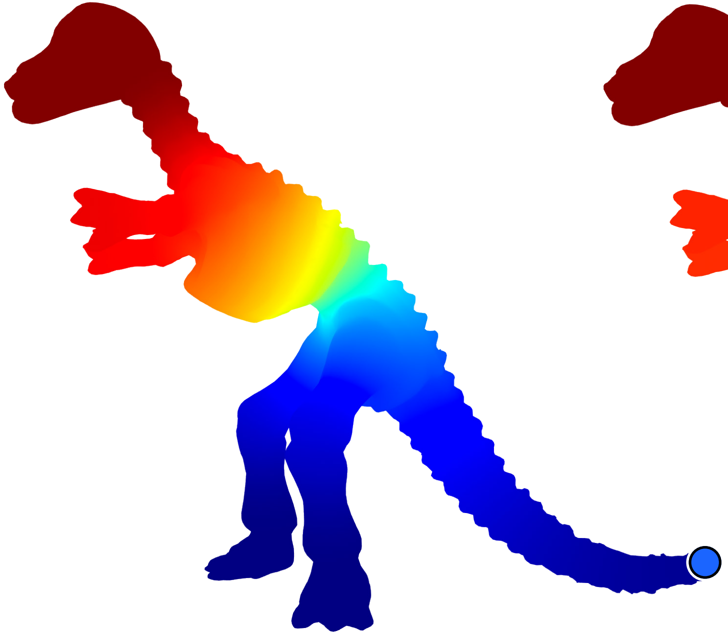


We preserve troublesome appendages

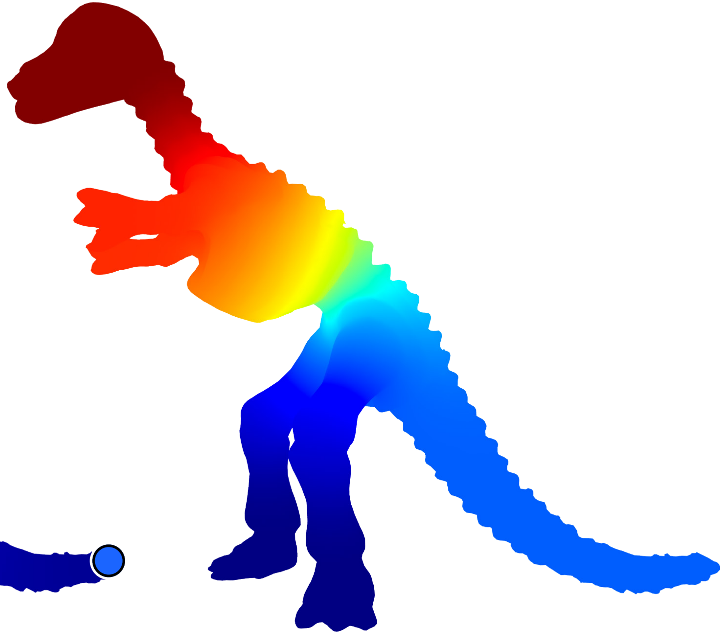


Our weights attach appendages to body



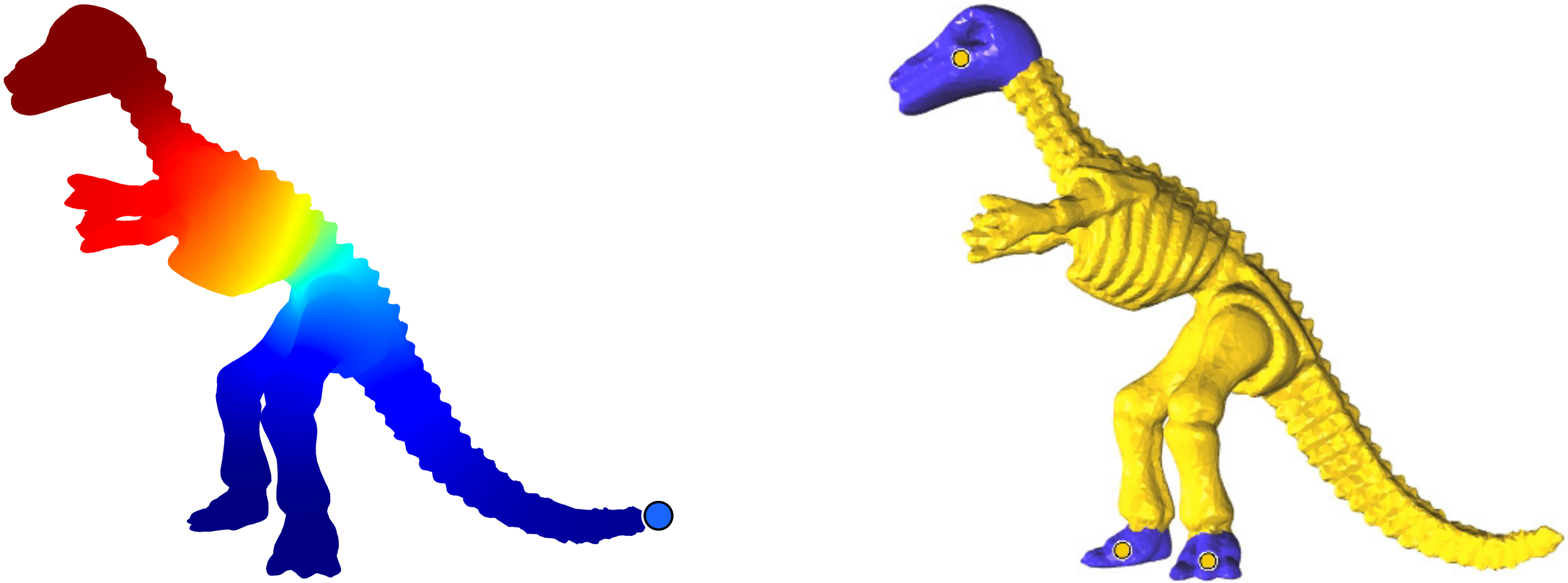


[Botsch & Kobbelt 2004, Jacobson et al. 2011]



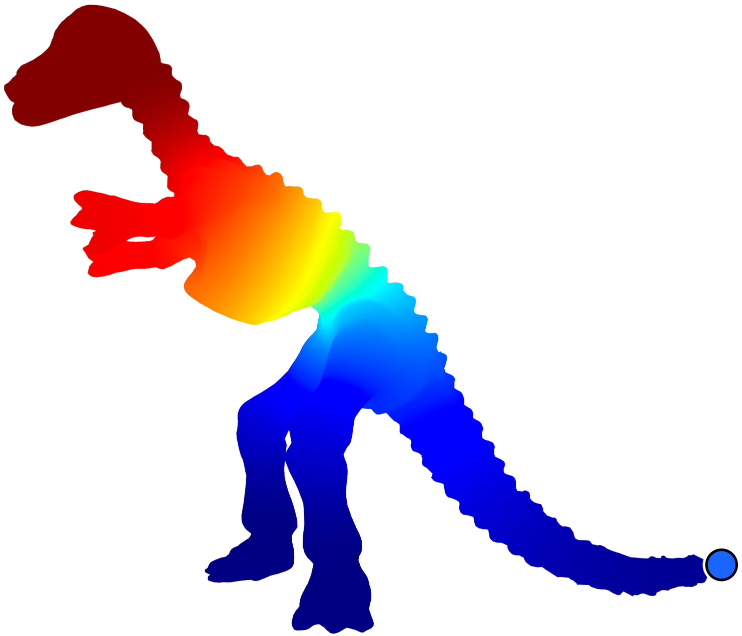
Our method

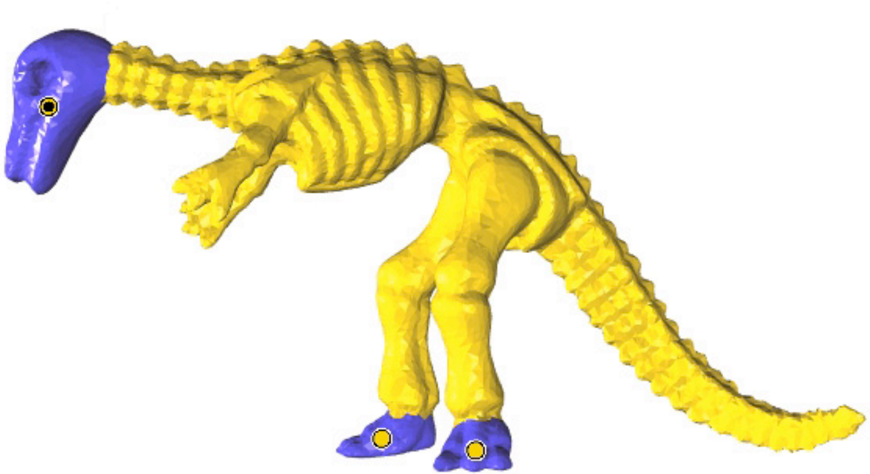
Extrema glue appendages to far-away handles



[Botsch & Kobbelt 2004, Jacobson et al. 2011]

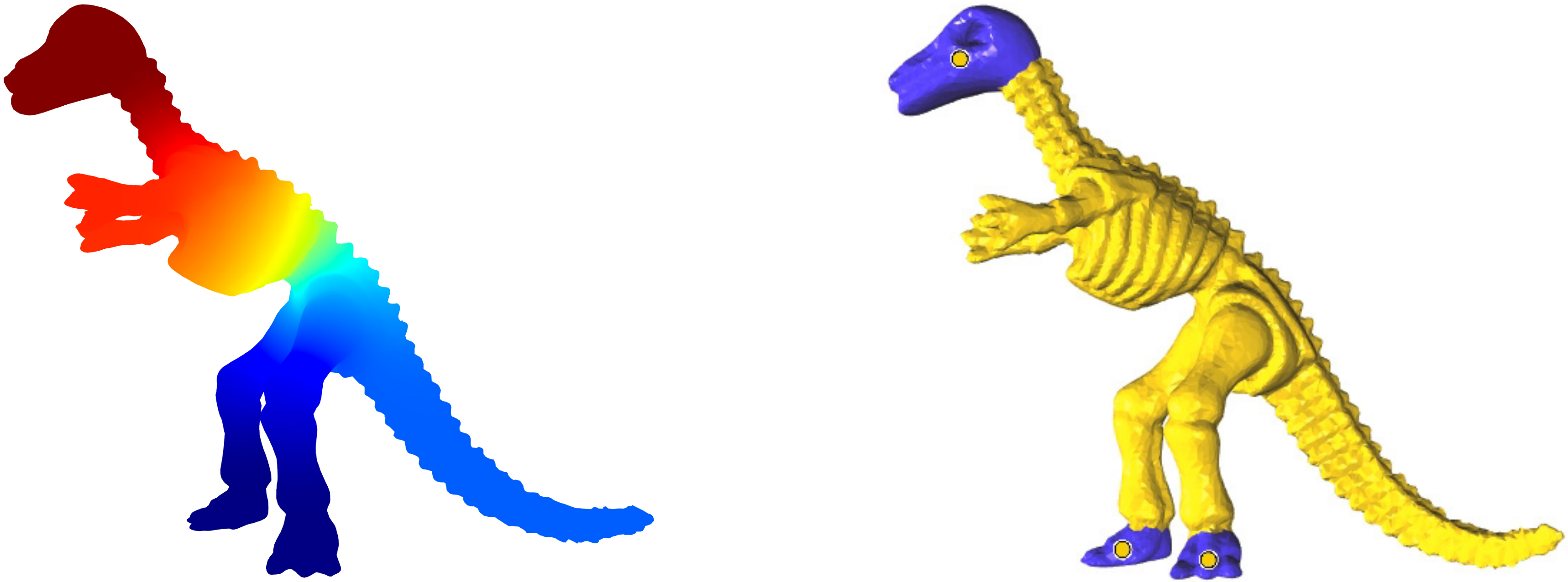
Extrema glue appendages to far-away handles





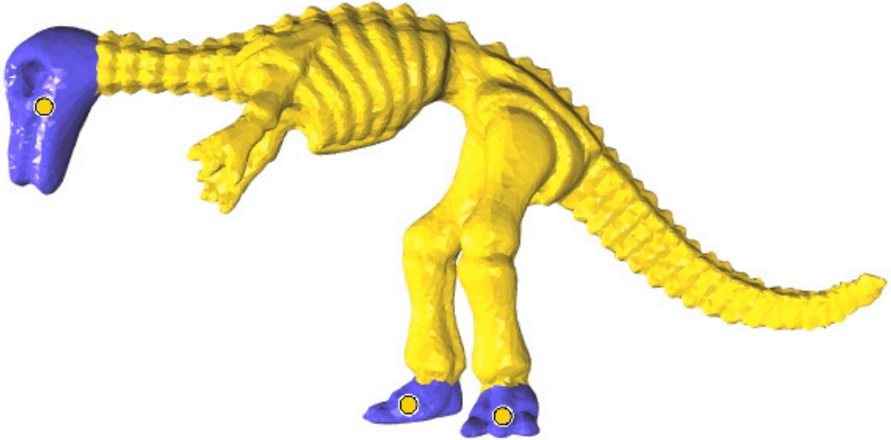
[Botsch & Kobbelt 2004, Jacobson et al. 2011]

Our weights attach appendages to body



Our method

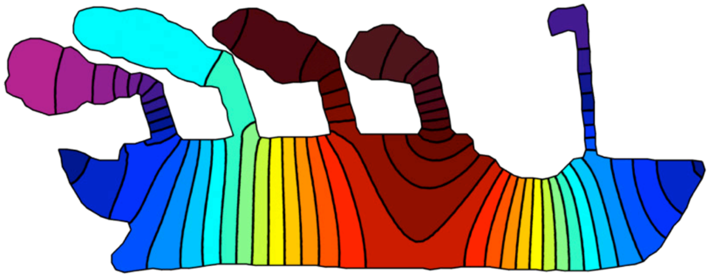
Our weights attach appendages to body



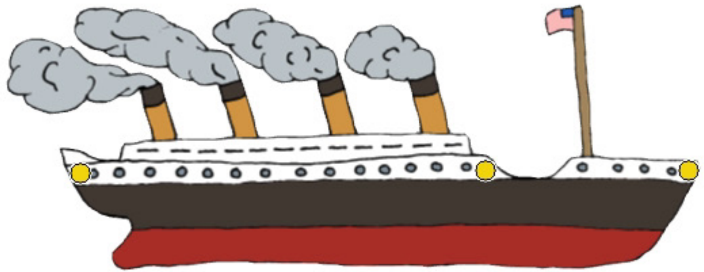
Our method

Extrema distort small features

Unconstrained Δ^2 [Botsch & Kobbelt 2004]

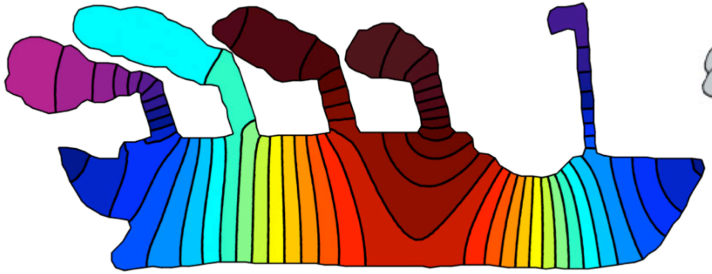


weight of middle point

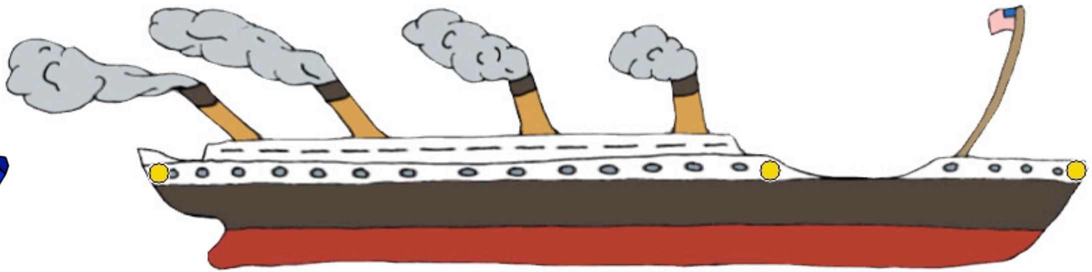


Extrema distort small features

Unconstrained Δ^2 [Botsch & Kobbelt 2004]



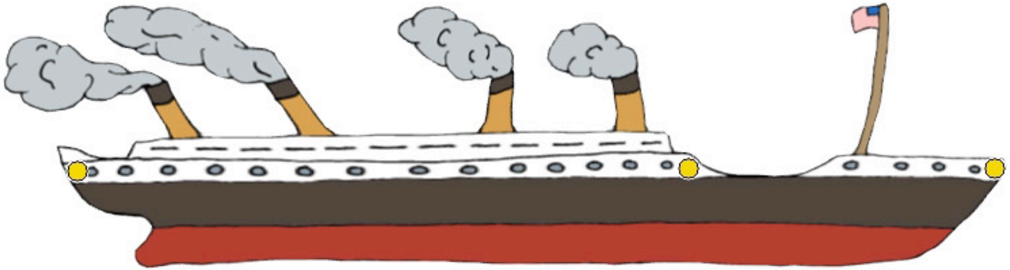
weight of middle point



Extrema distort small features

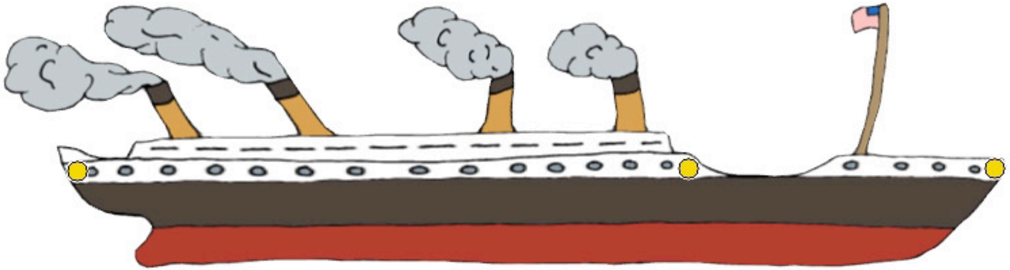
Bounded Δ^2 [Jacobson et al. 2011]

weight of middle point

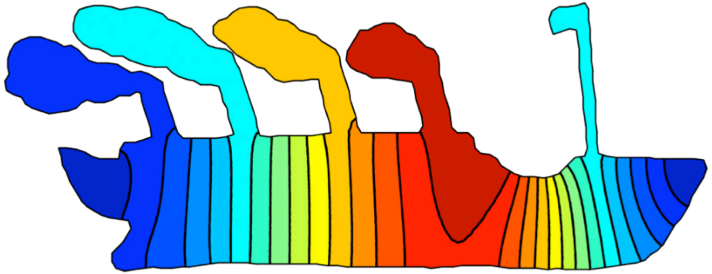


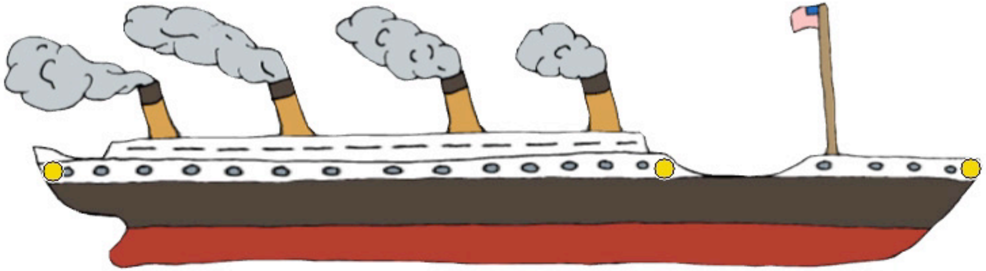
"Monotonicity" helps preserve small features

Bounded Δ^2 [Jacobson et al. 2011]

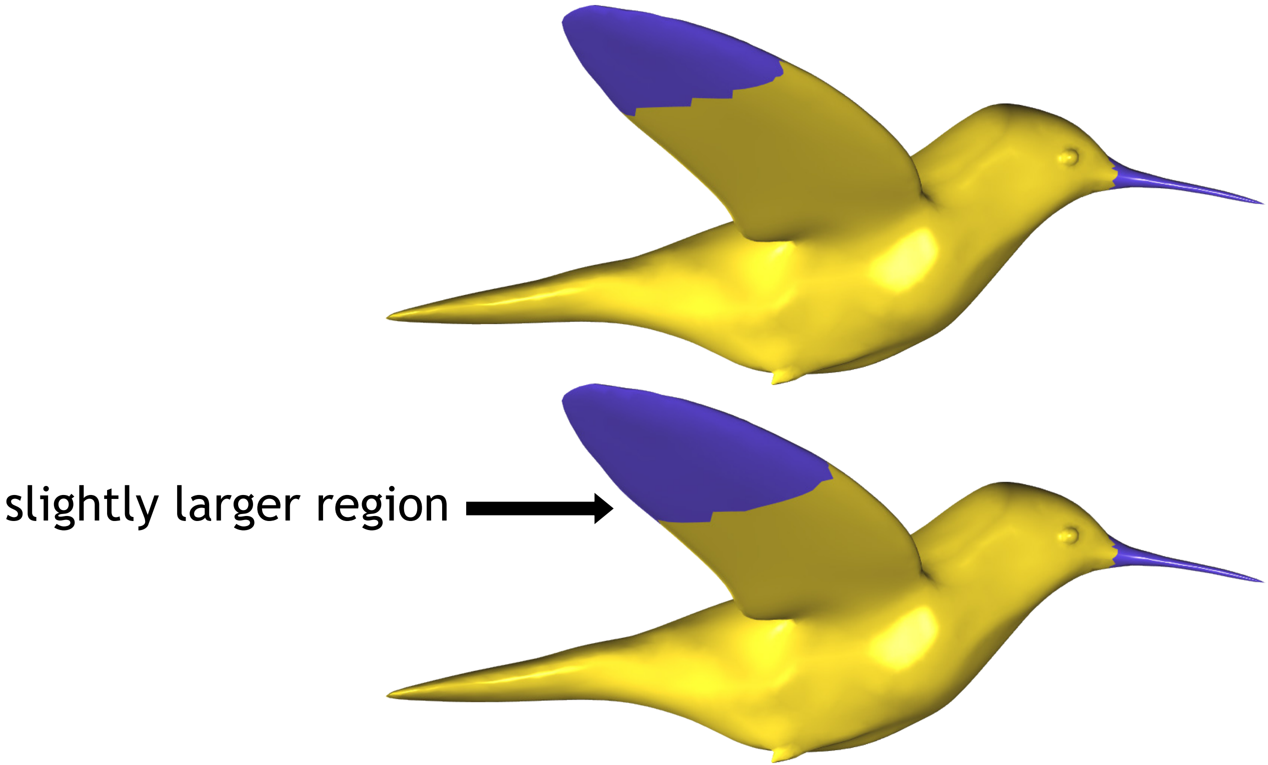


Our Δ^2

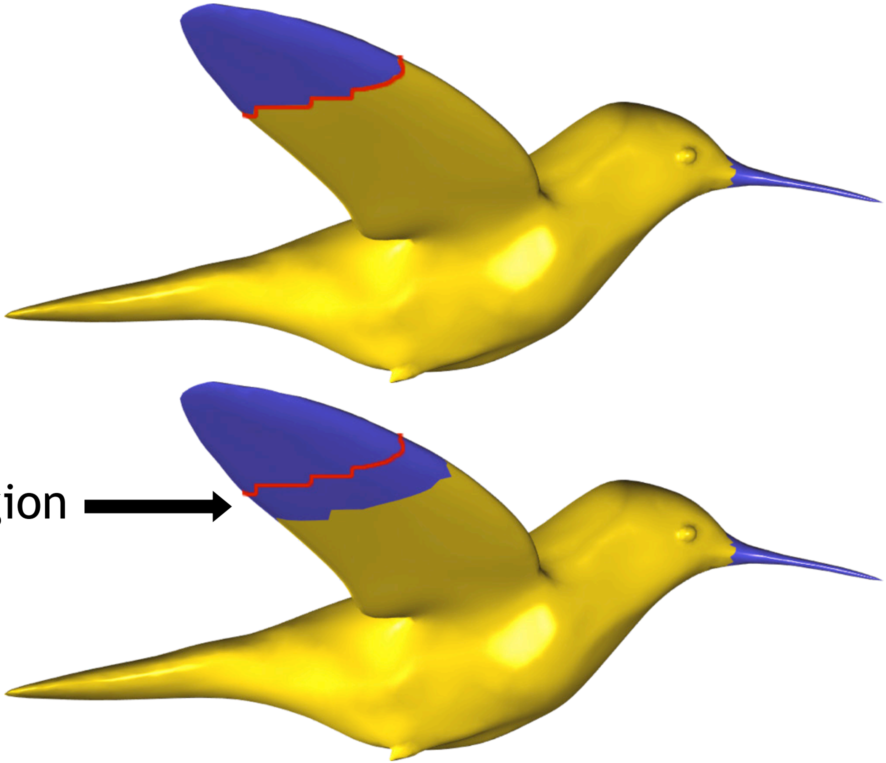




Spurious extrema are unstable, may “flip”

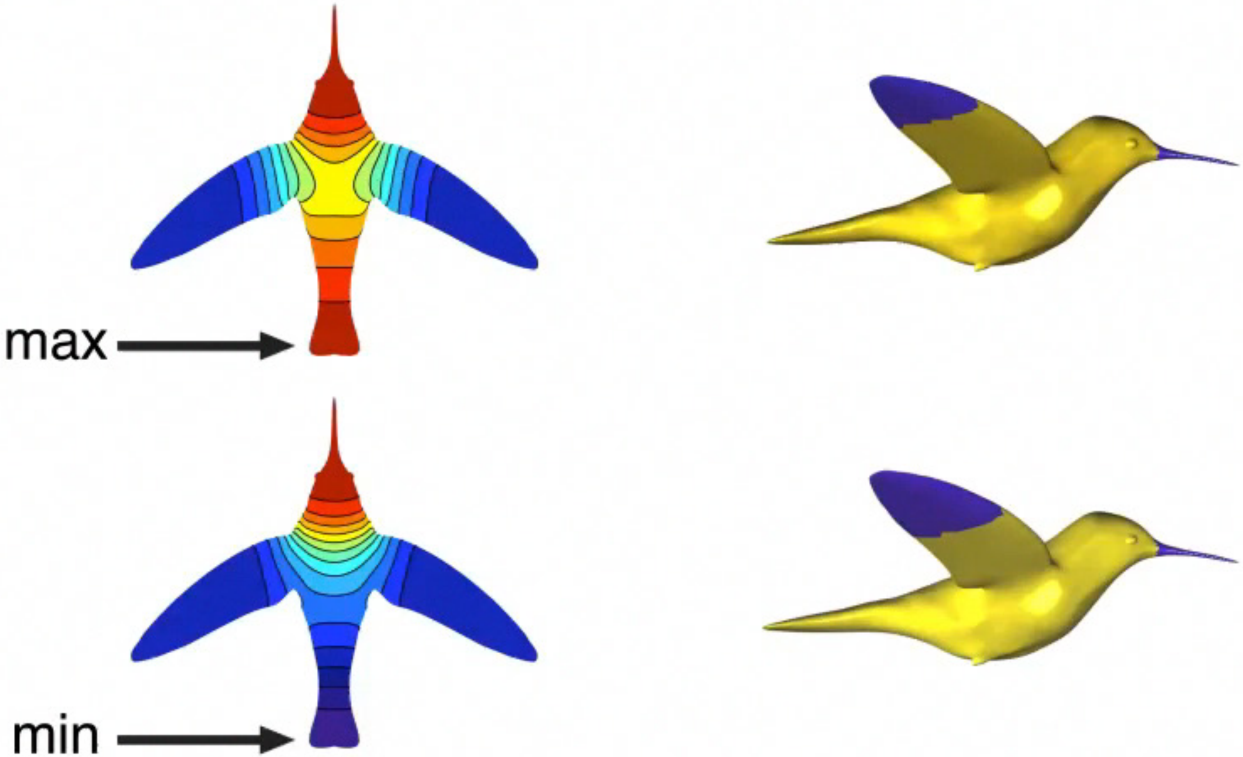


Spurious extrema are unstable, may “flip”



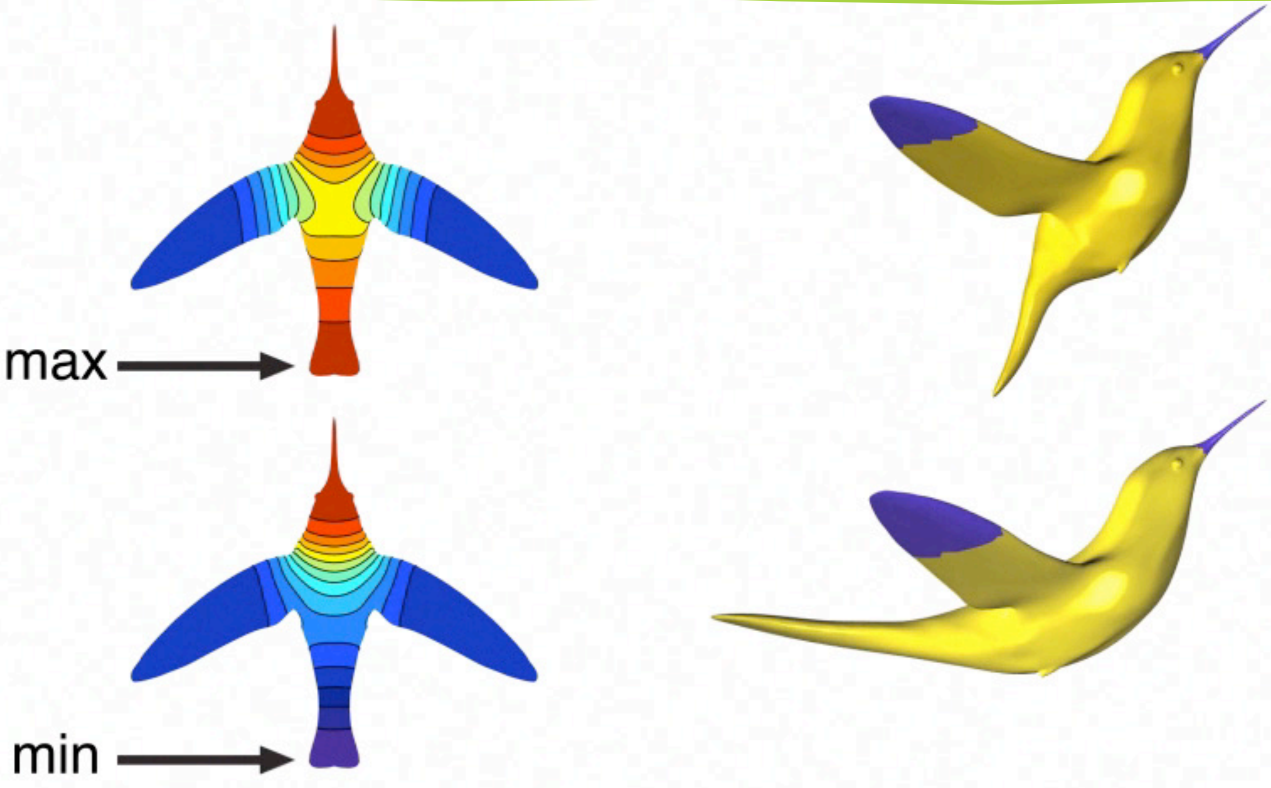
slightly larger region

Spurious extrema are unstable, may "flip"



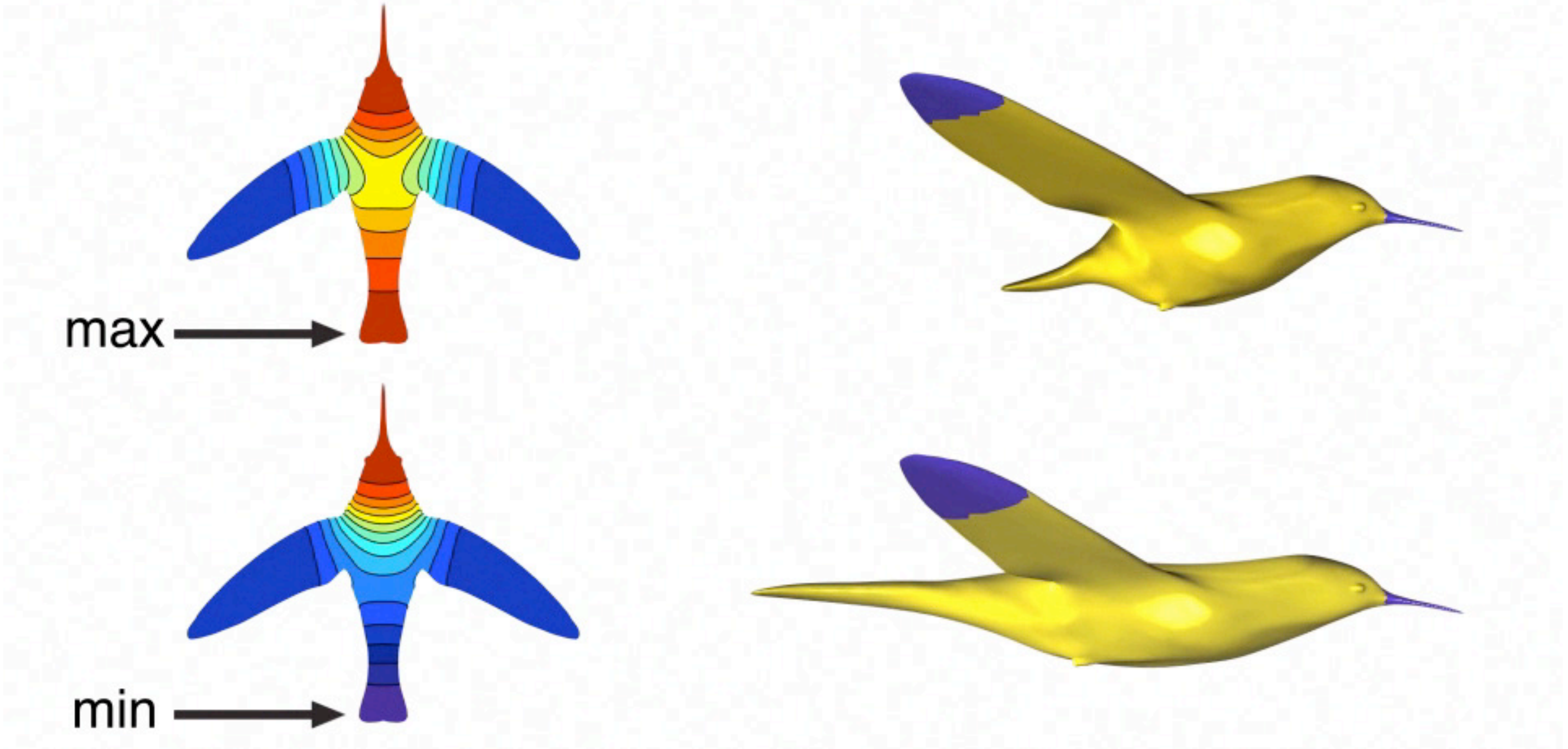
Unconstrained Δ^3 [Botsch & Kobbelt, 2004]

Spurious extrema are unstable, may “flip”



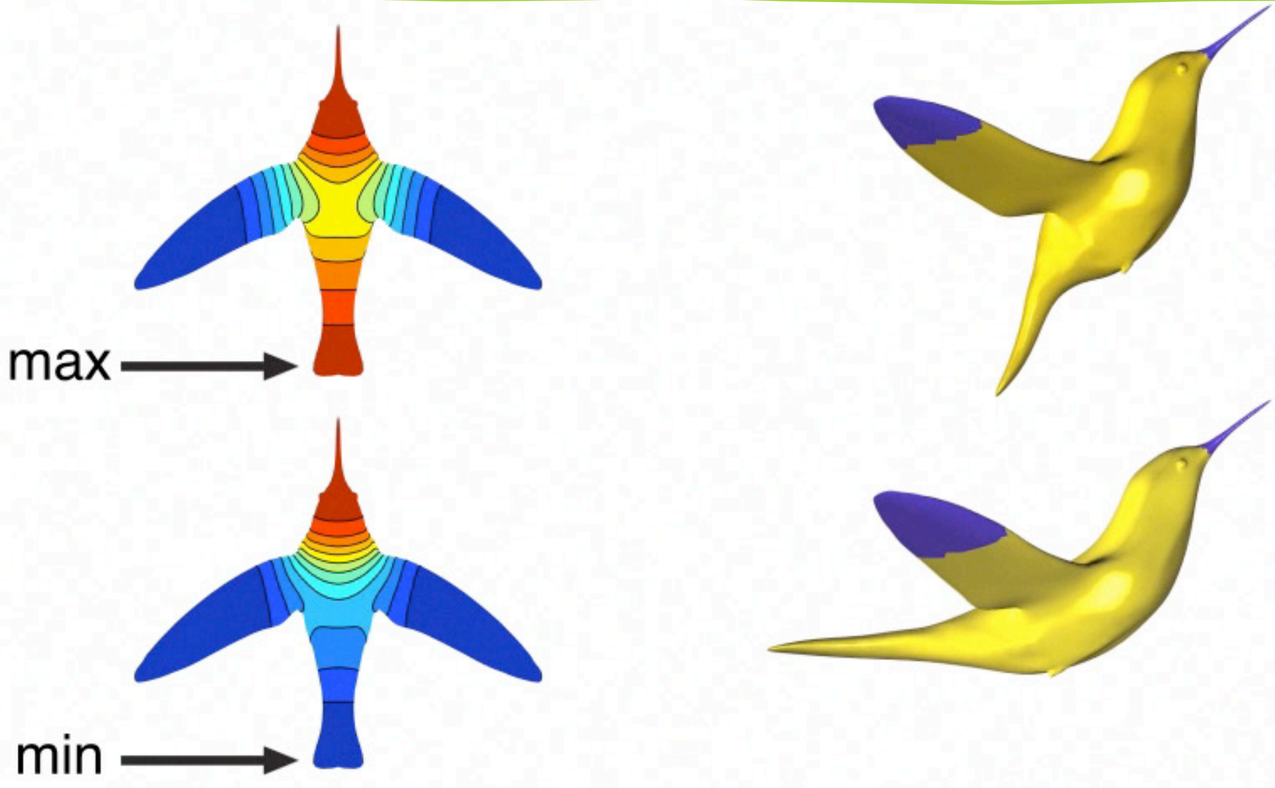
Unconstrained Δ^3 [Botsch & Kobbelt, 2004]

Spurious extrema are unstable, may "flip"



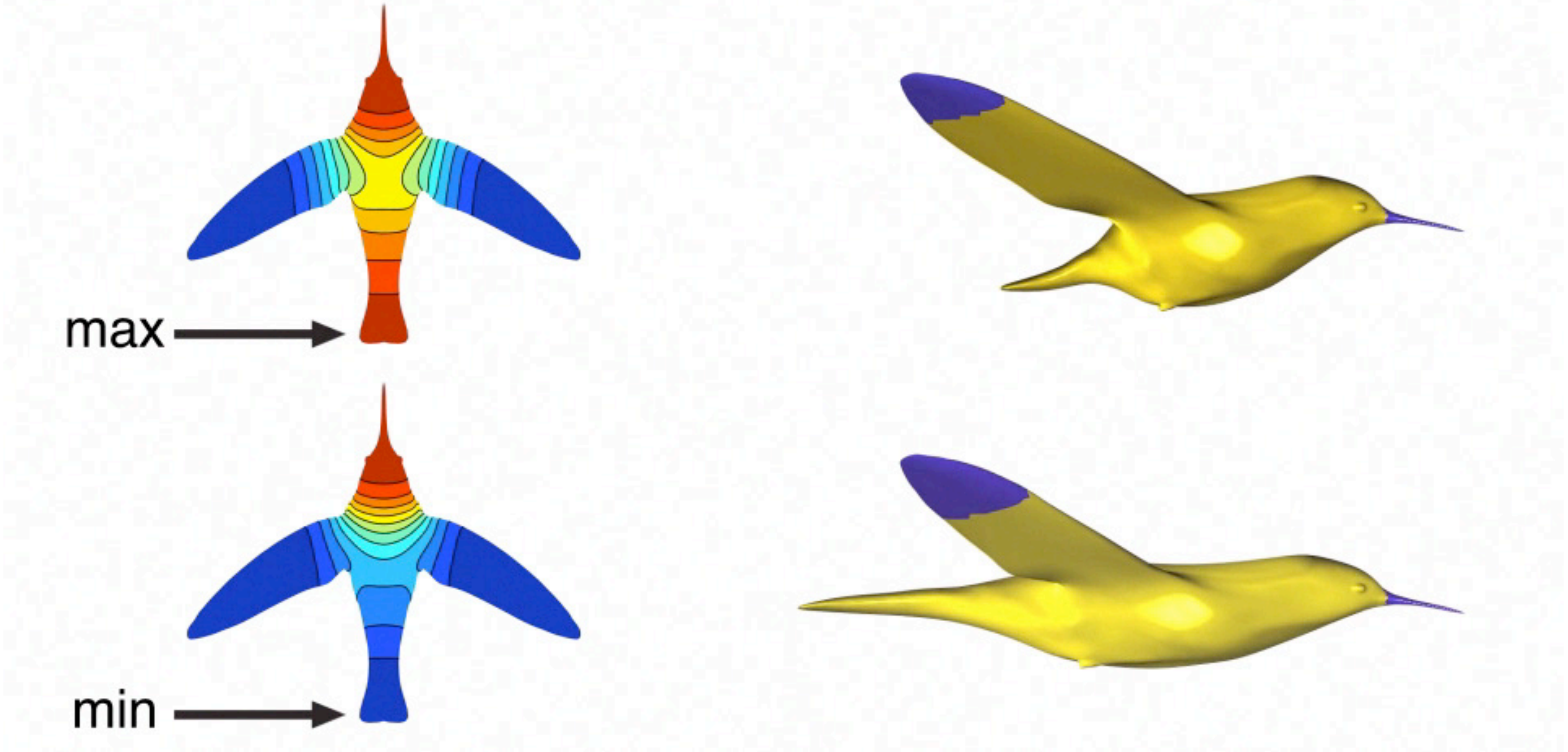
Unconstrained Δ^3 [Botsch & Kobbelt, 2004]

Spurious extrema are unstable, may "flip"



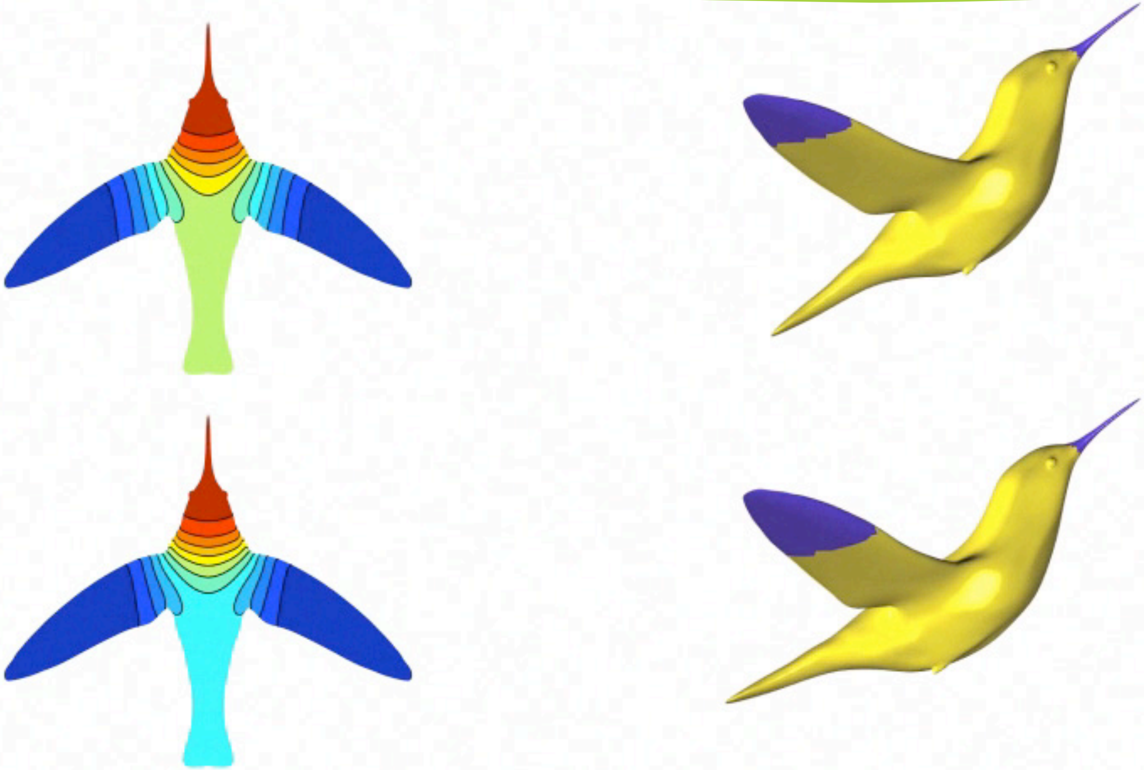
Bounded Δ^3

Spurious extrema are unstable, may "flip"



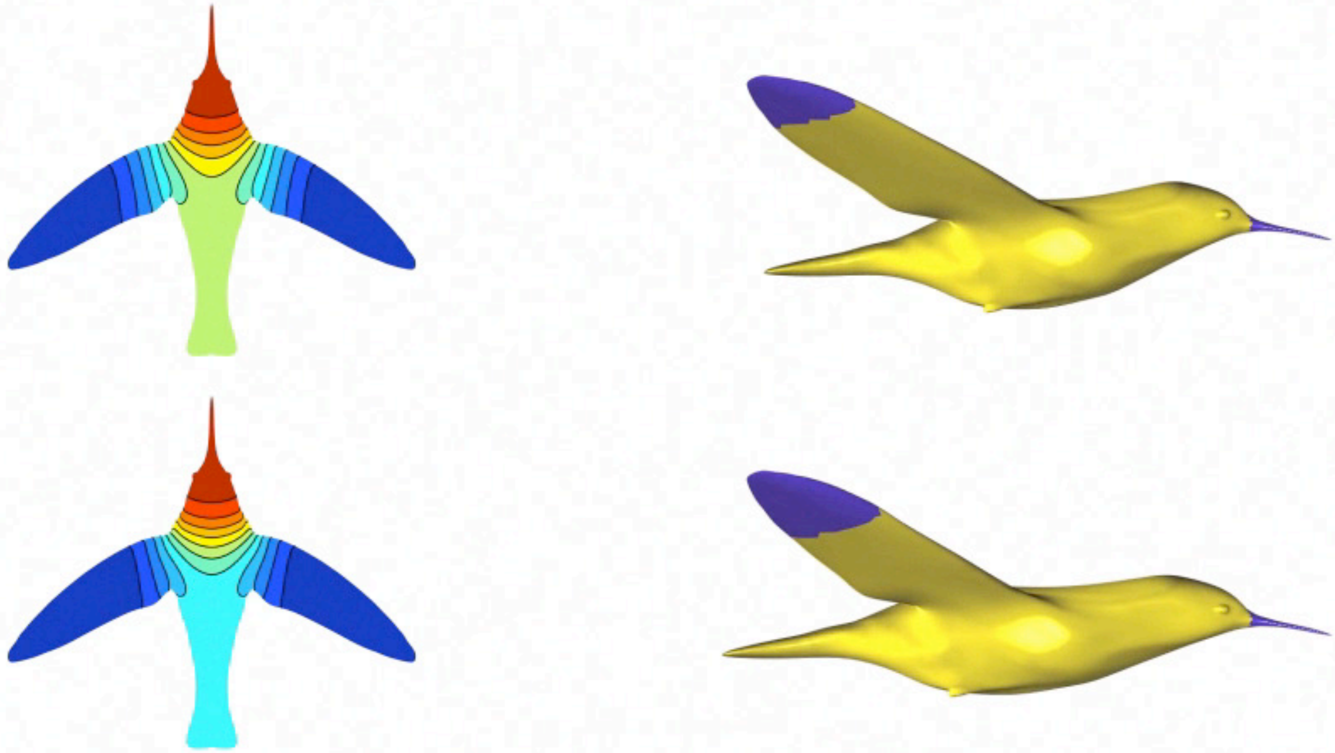
Bounded Δ^3

Lack of extrema leads to more stability



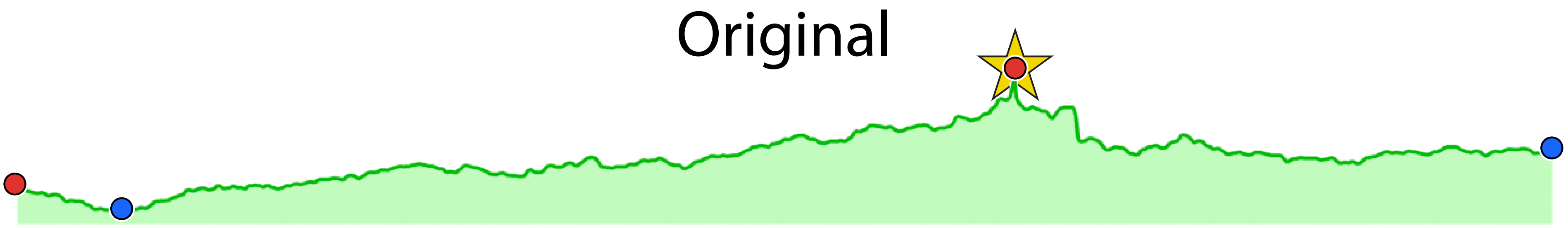
Our Δ^3

Lack of extrema leads to more stability



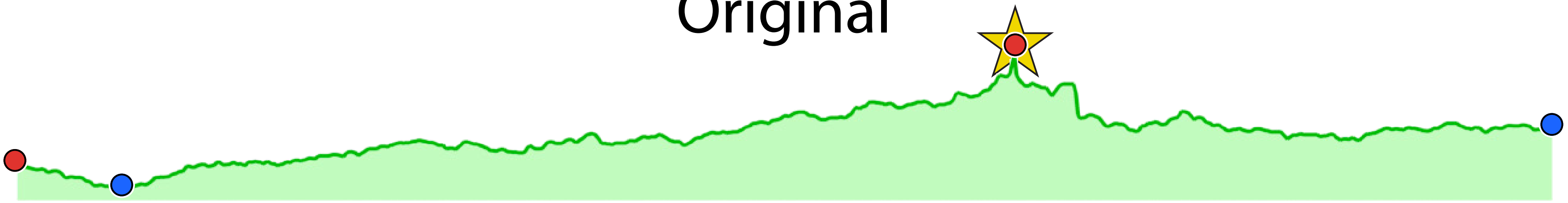
Our Δ^3

Even control continuity at extrema

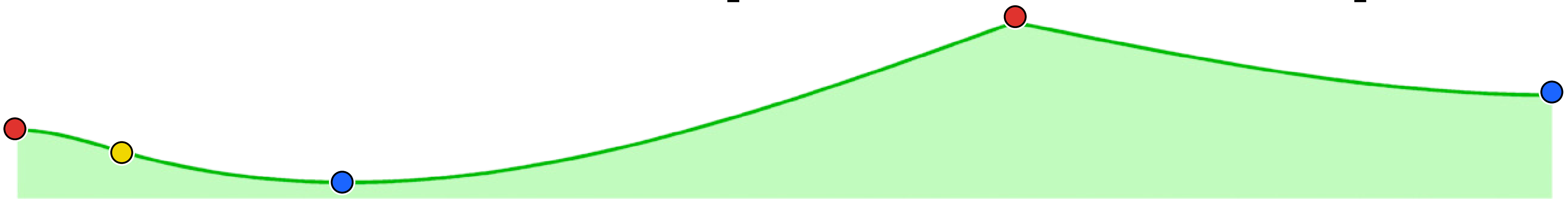


Even control continuity at extrema

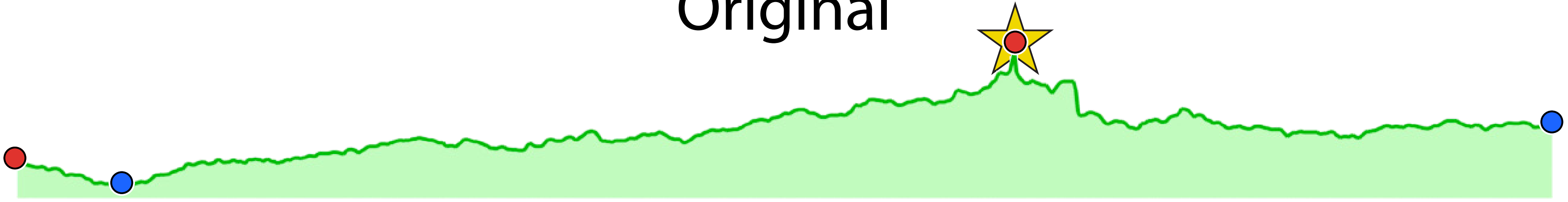
Original

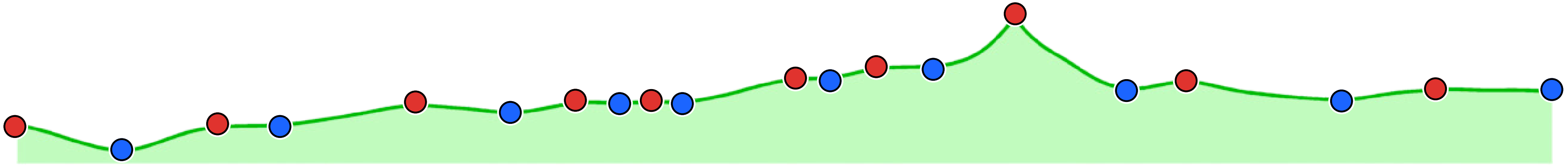


Direct extension of [Botsch & Kobbelt 2004]

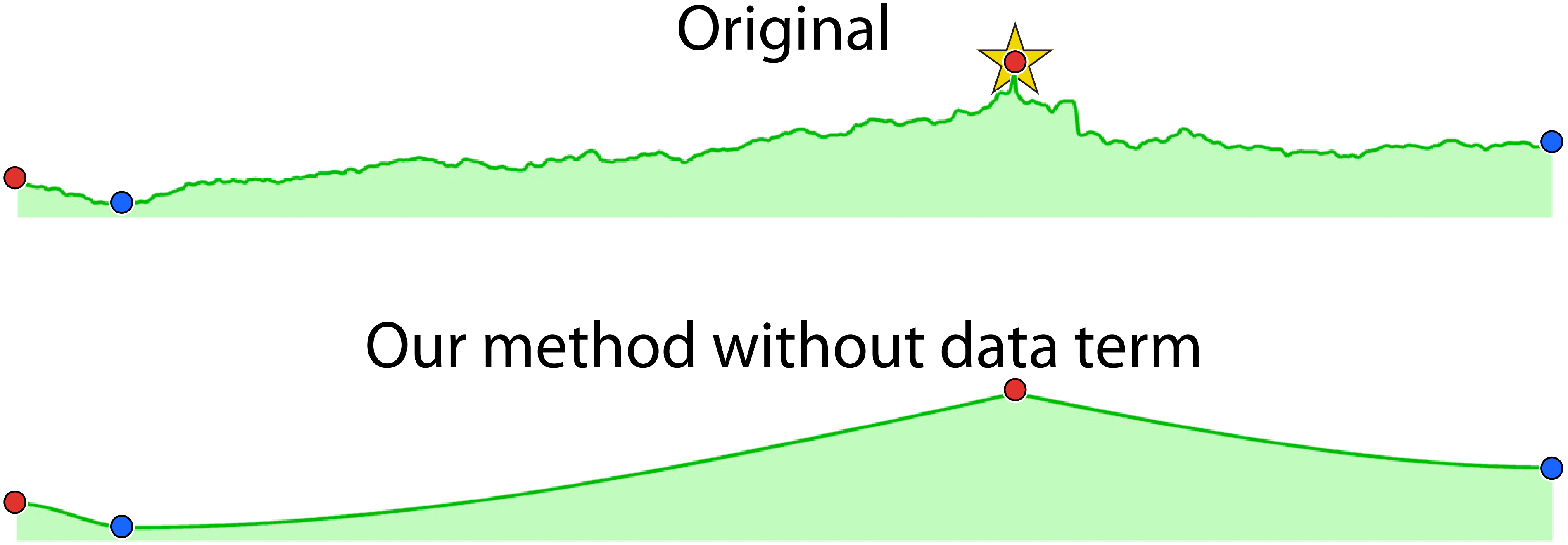


Even control continuity at extrema

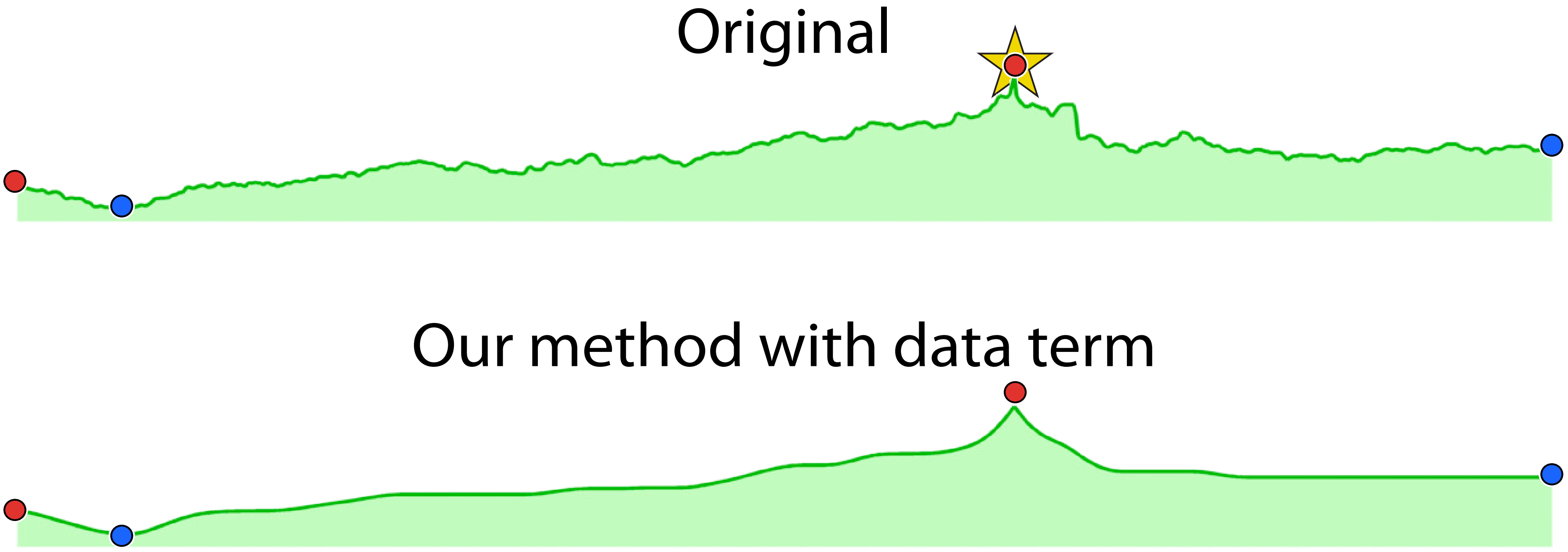




Even control continuity at extrema

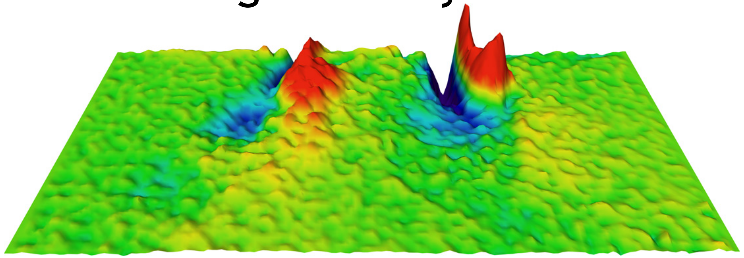


Even control continuity at extrema



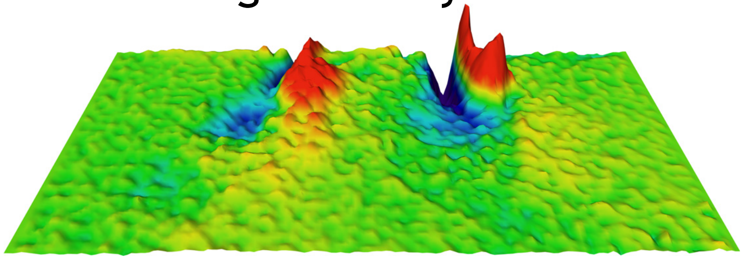
Reproduces results of Weinkauf et al. 2010…

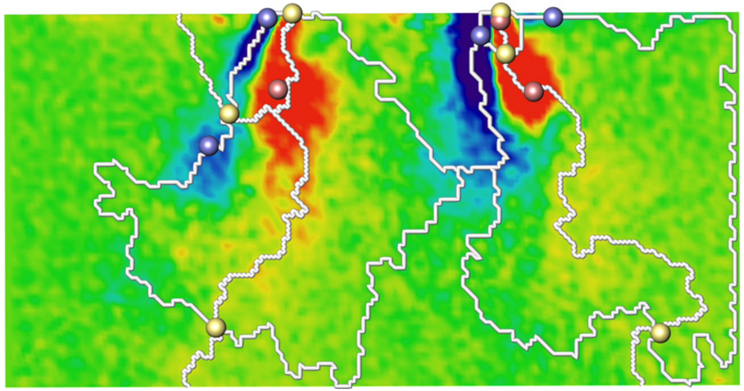
Original noisy data



Reproduces results of Weinkauf et al. 2010…

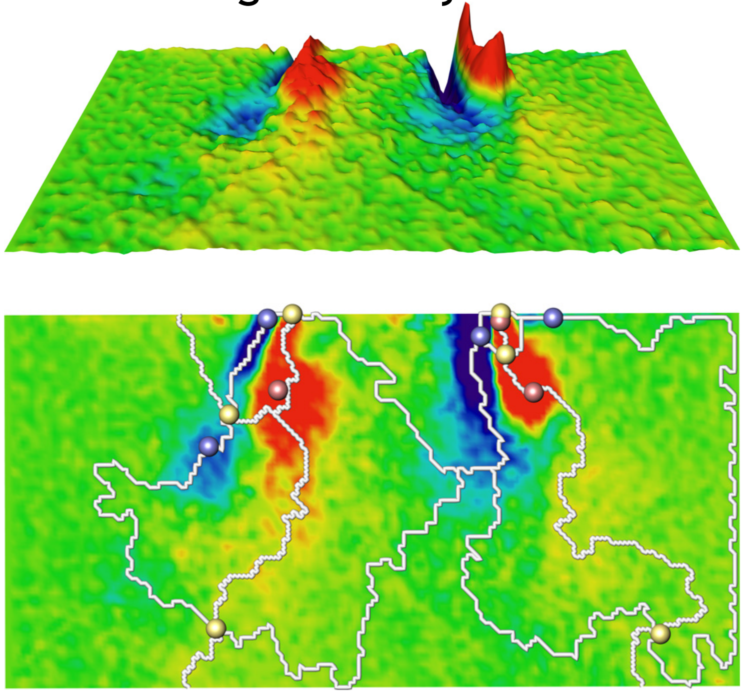
Original noisy data



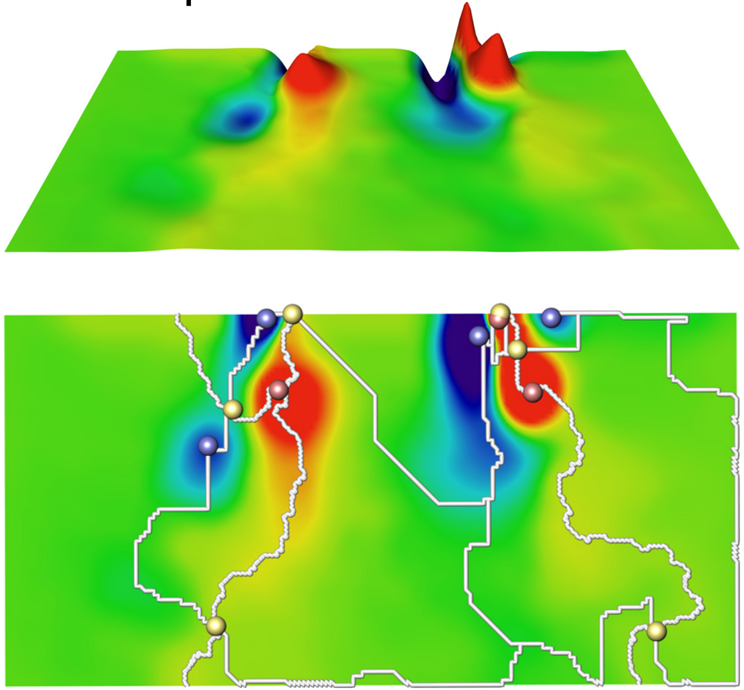


Reproduces results of Weinkauf et al. 2010…

Original noisy data

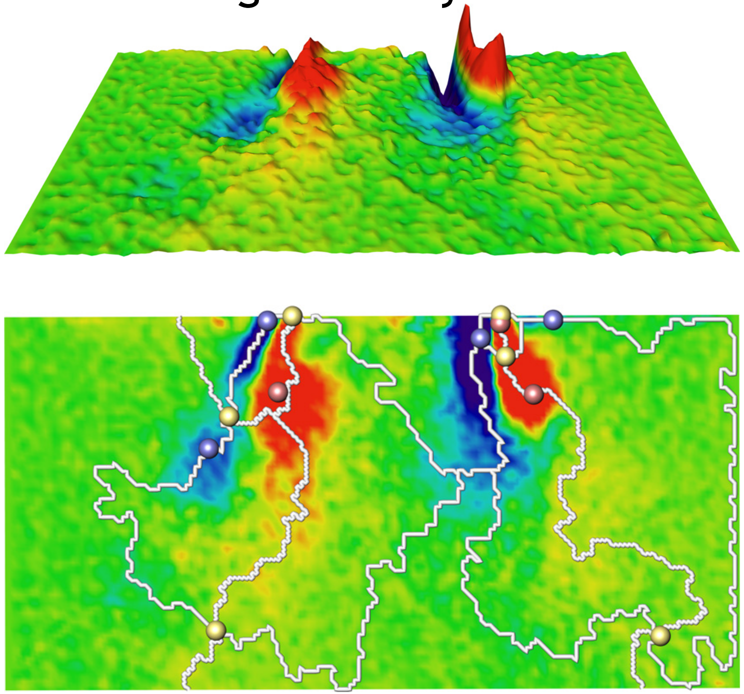


Simplified and smoothed

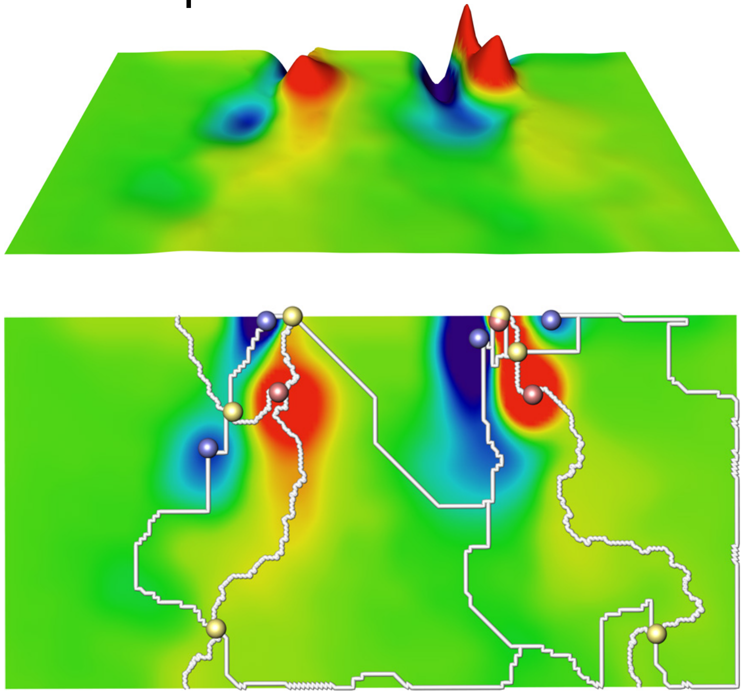


Reproduces results of Weinkauf et al. 2010…

Original noisy data

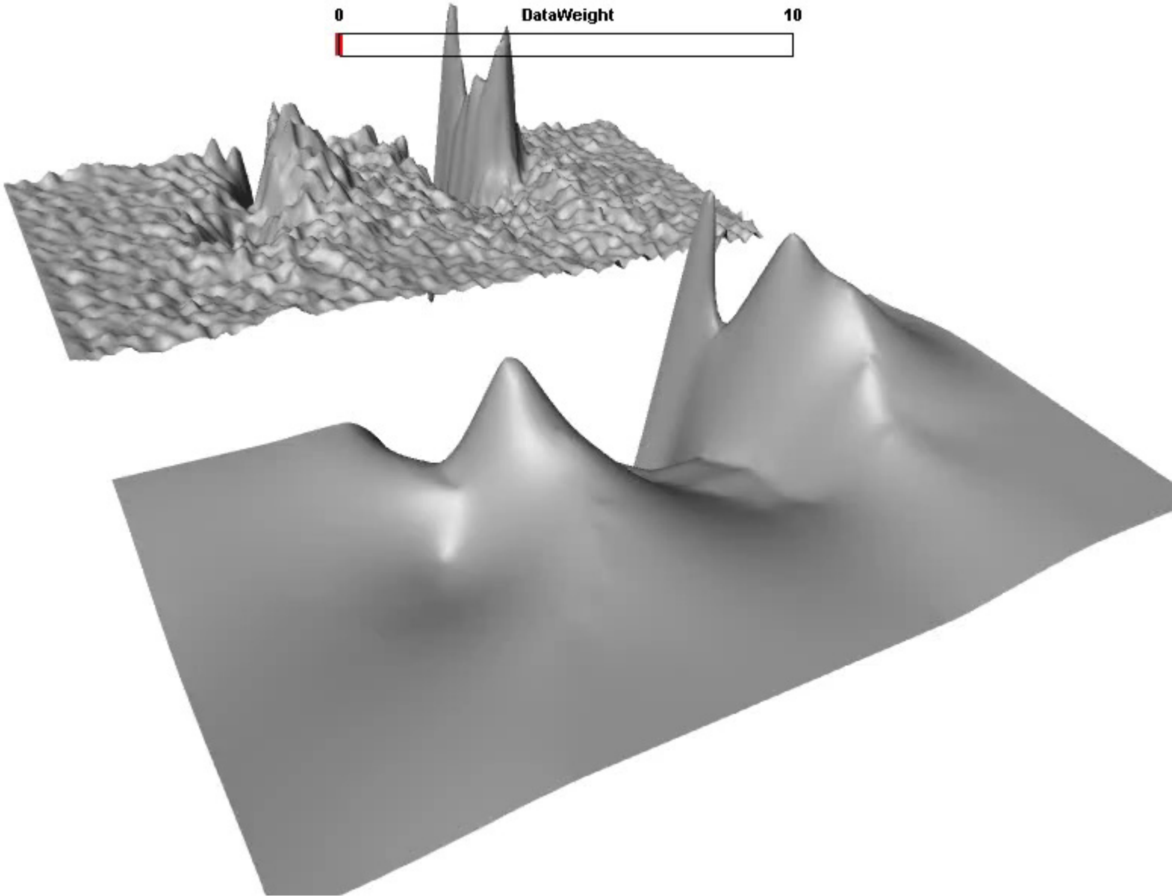


Simplified and smoothed



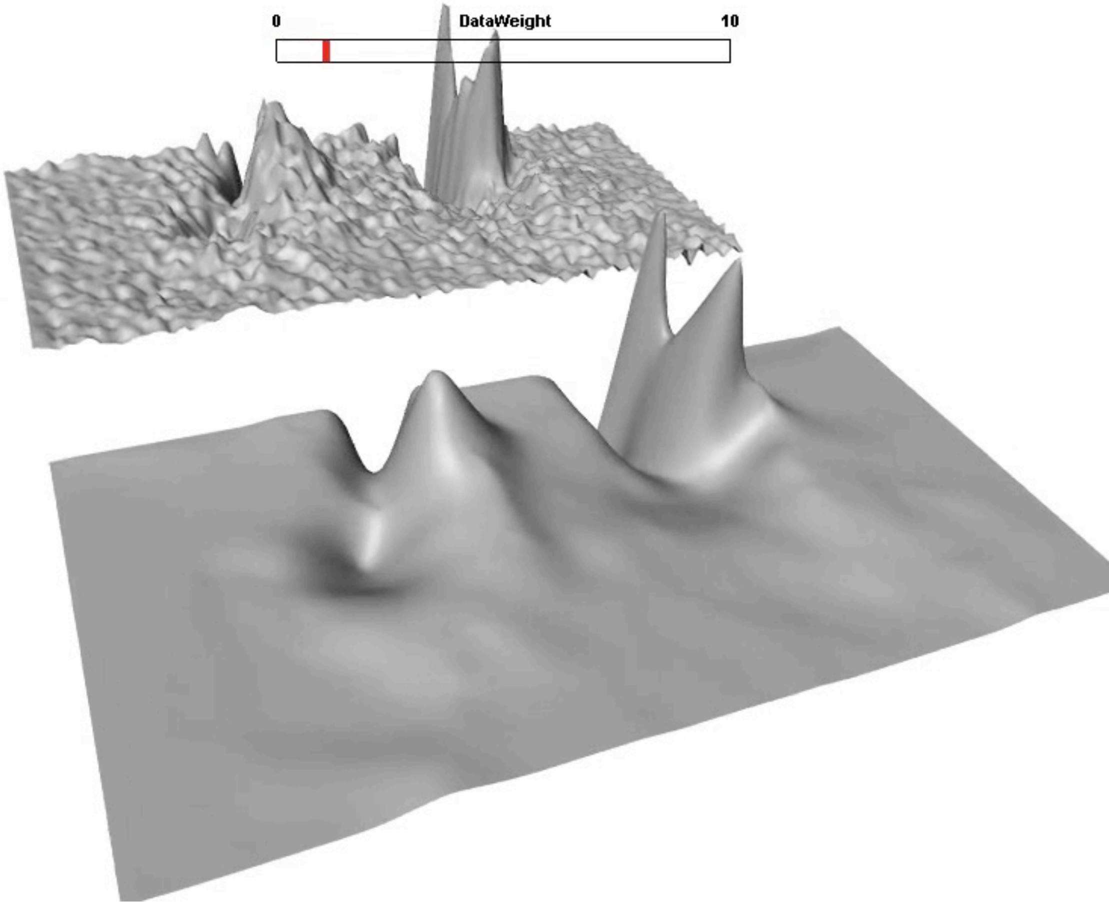
… but 1000 times faster

30K vertices
5 seconds per solve



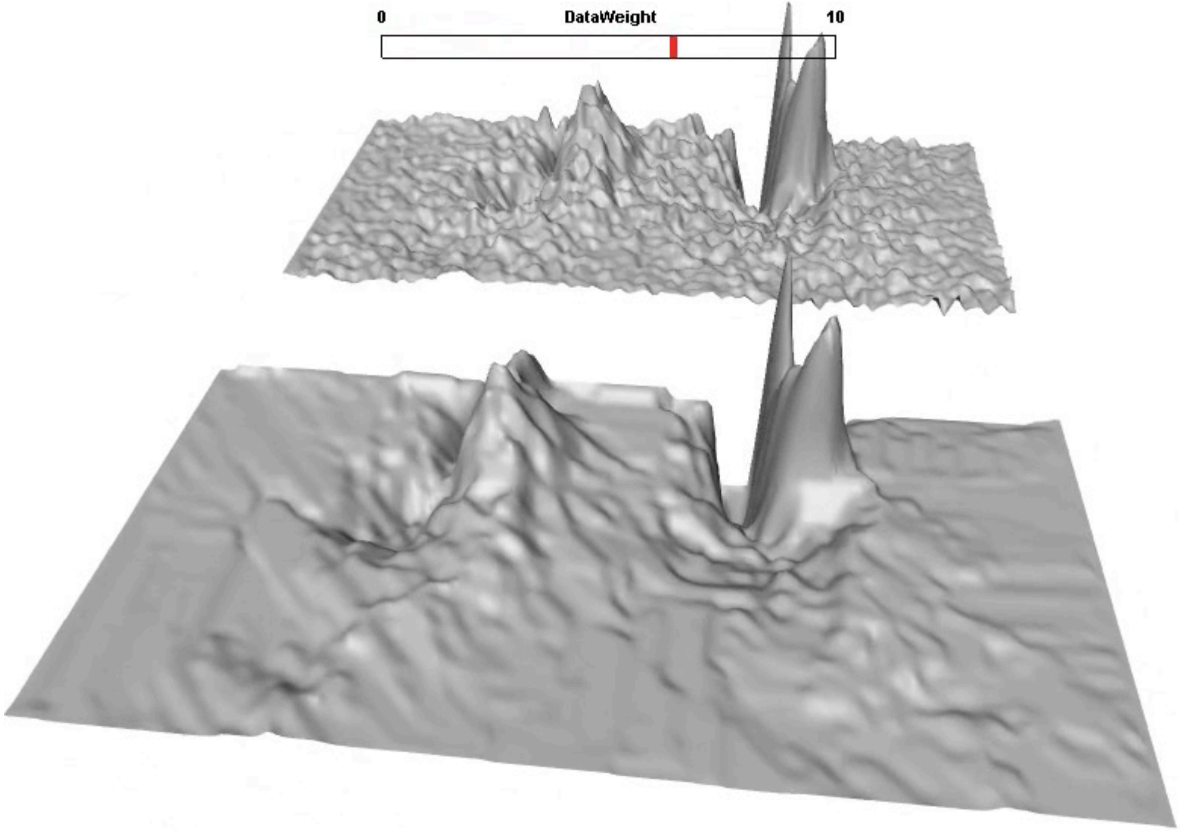
… but 1000 times faster

30K vertices
5 seconds per solve



… but 1000 times faster

30K vertices
5 seconds per solve



Conclusion: Important to control extrema

- Copy “monotonicity” of harmonic functions
- *Reduces* search-space, but optimization is tractable

Future work and discussion

- Larger, but still tractable subspace?
  - Consider all valid harmonic functions?

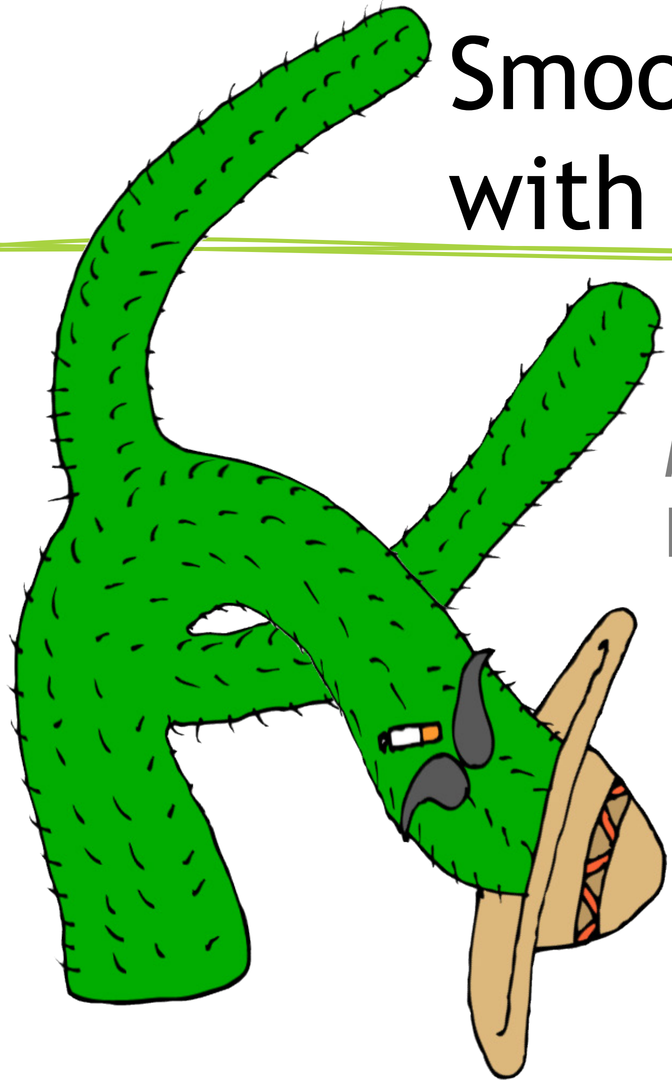
Future work and discussion

- Larger, but still tractable subspace?
  - Consider all valid harmonic functions?
- Continuous formulation?

Acknowledgements

We thank Kenshi Takayama for his valuable feedback. This work was supported in part by an SNF award 200021_137879 and by a gift from Adobe Systems.

Smooth Shape-Aware Functions with Controlled Extrema



MATLAB Demo:
http://igl.ethz.ch/projects/monotonic/

Alec Jacobson (jacobson@inf.ethz.ch)
Tino Weinkauf
Olga Sorkine