Controlled-Topology Filtering

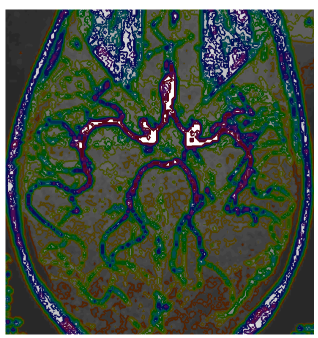
Yotam Gingold
& Denis Zorin

new york university mrl

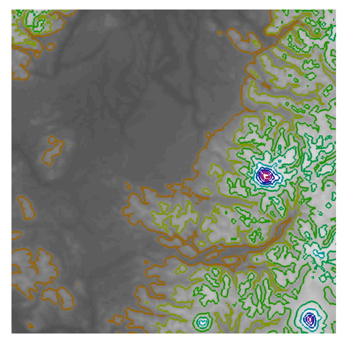
Motivation

Many applications require extraction of isolines & isosurfaces (contours) from scalar functions

- MRI, CT, terrain data, scientific computing



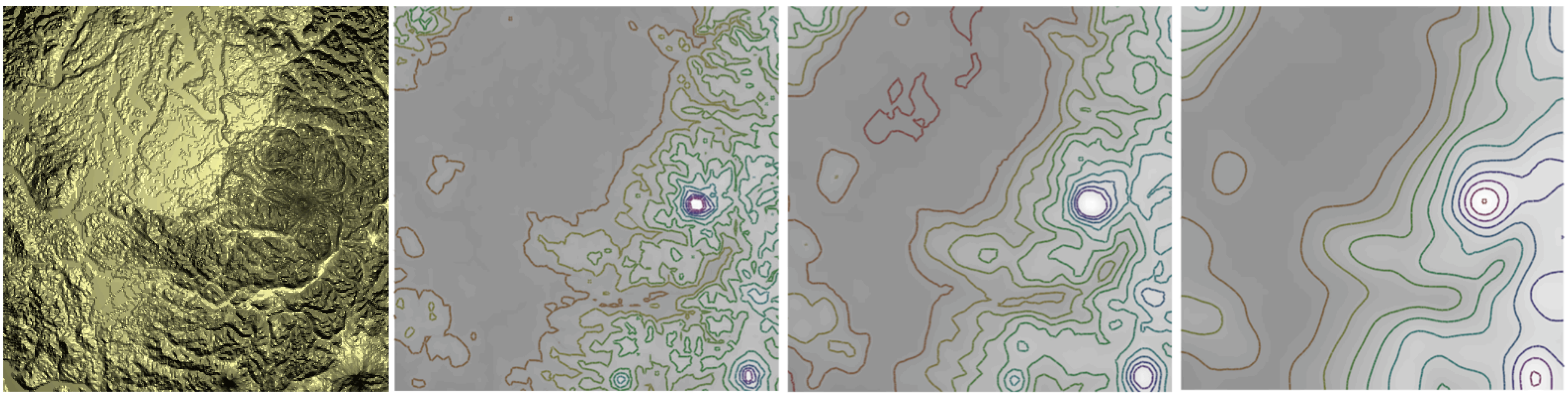
Cow CT Scan



Puget Sound

Motivation

Want to filter (smooth, sharpen) all contours at once



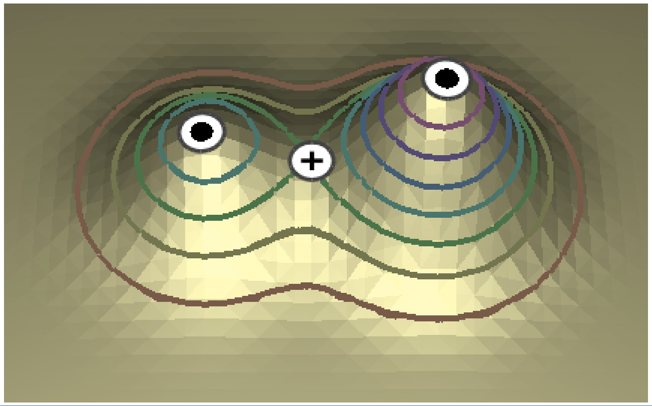
- Downsampling
- Noise reduction

Contour-line topology

Number of contours at all isovalues

- In 2D fields, value can be height

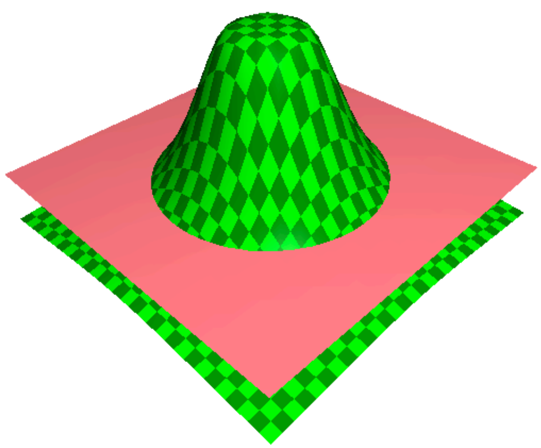
Topology changes occur when value equals value of a critical point (min, max, saddle)

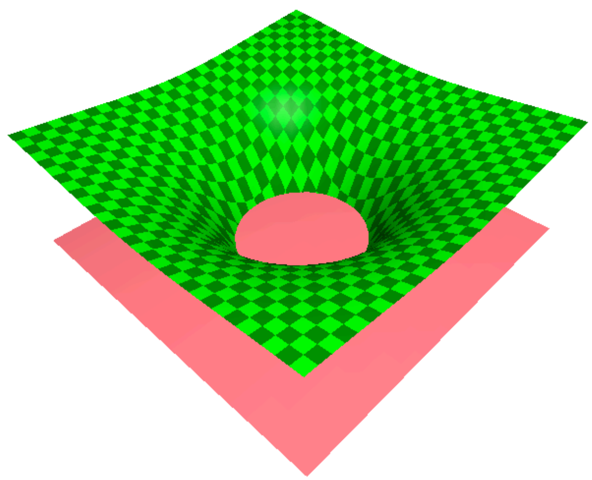


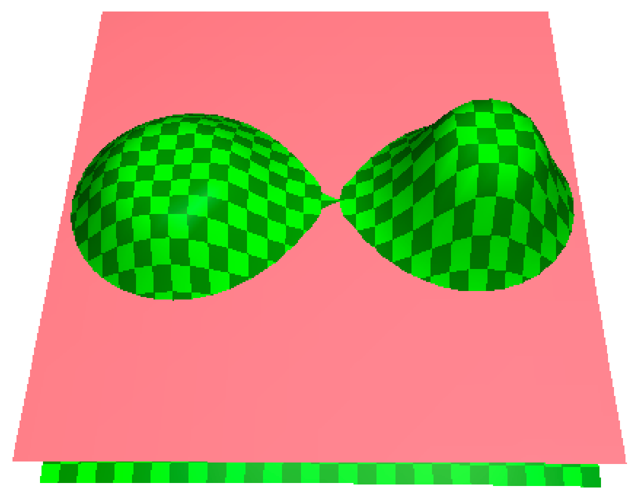
Critical Points

Maxima and minima correspond to hills and valleys

Saddles join two hills or valleys



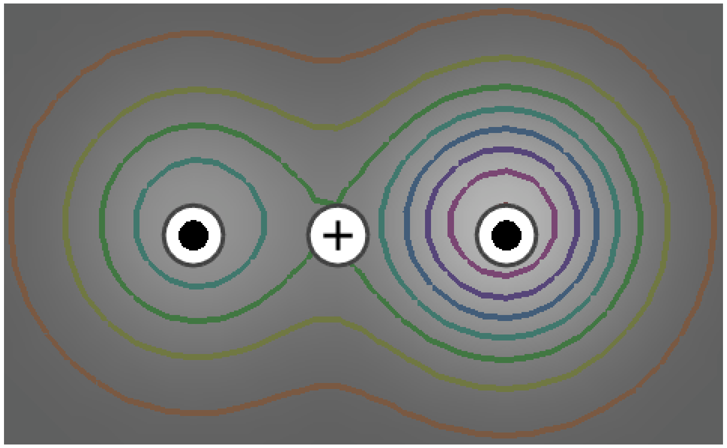




Features

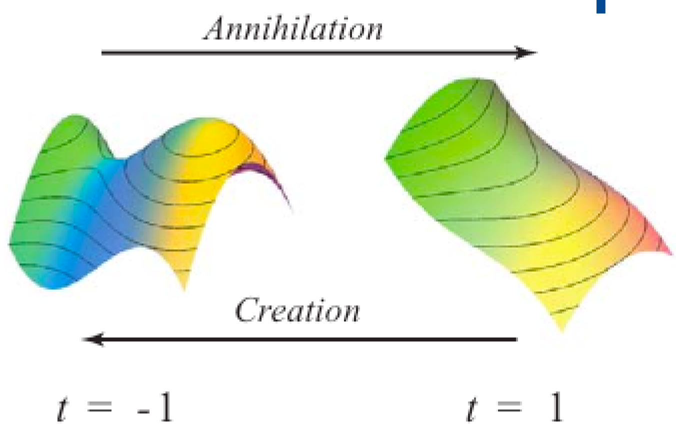
Correspond to critical contours passing through saddles

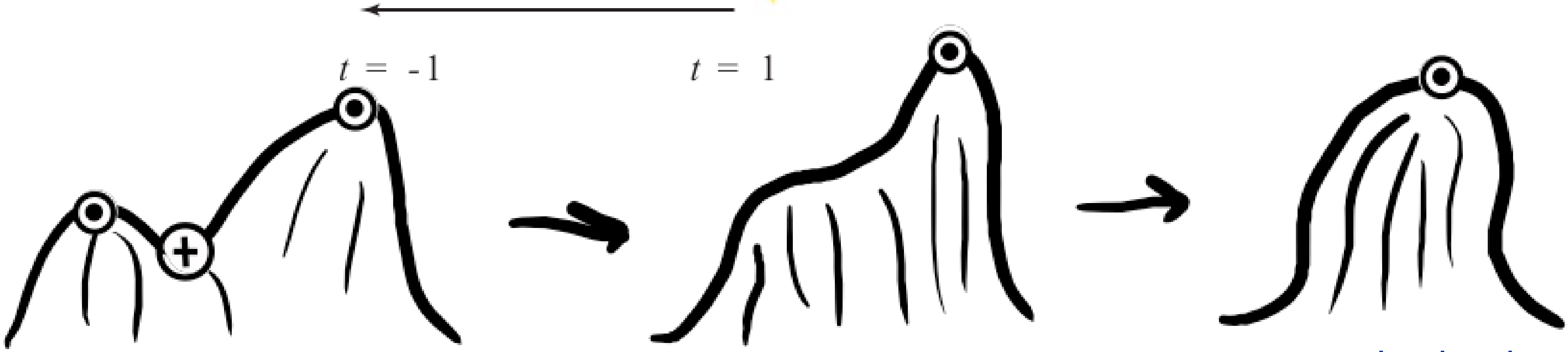
- A saddle divides a contour in two
- The interior of a contour containing an unpaired extremum is a feature



Topological Events

Features appear/disappear in saddle-min/max pairs

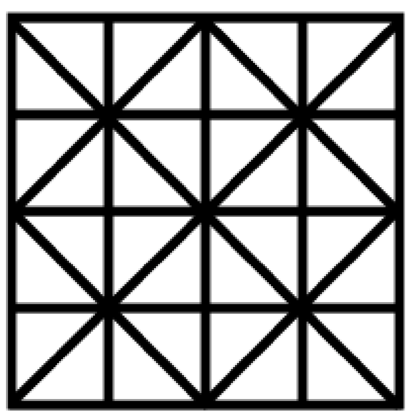


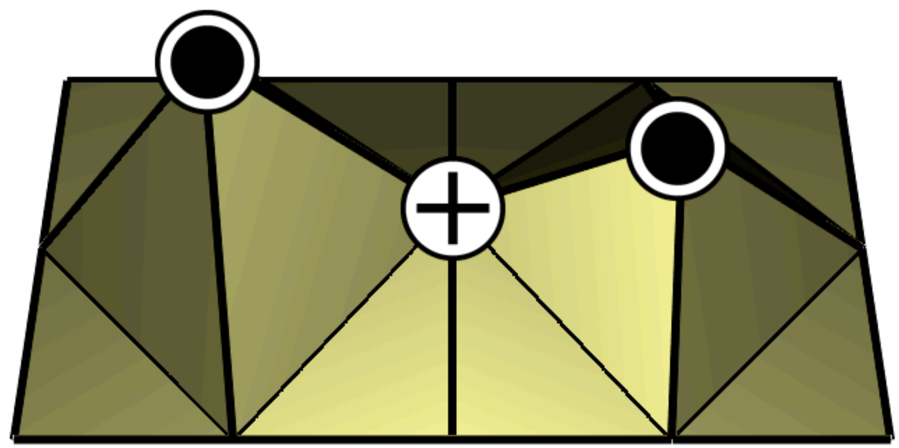


new york university mrl

Piecewise linear data

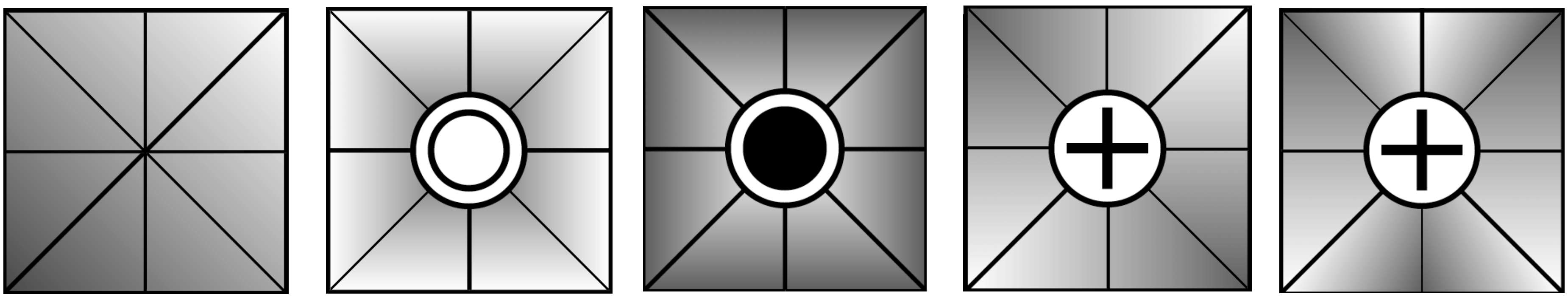
- Scalar values defined on a regular grid
- Simplicial mesh guarantees critical points only at vertices
- Some complex critical points can be stable





Piecewise linear data

Find critical points by comparing value with neighbors



regular **min** **max** **saddle (simple)** **saddle (monkey)**

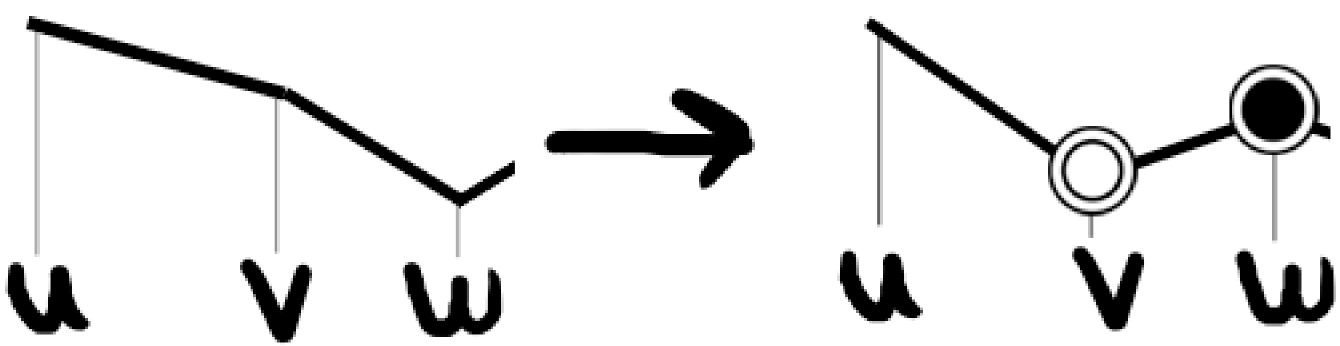
- Break ties with arbitrary, consistent perturbation

PL topological events

Only happens when 2 adjacent vertices change relative height

- The edge *flips* relative to equality

1D

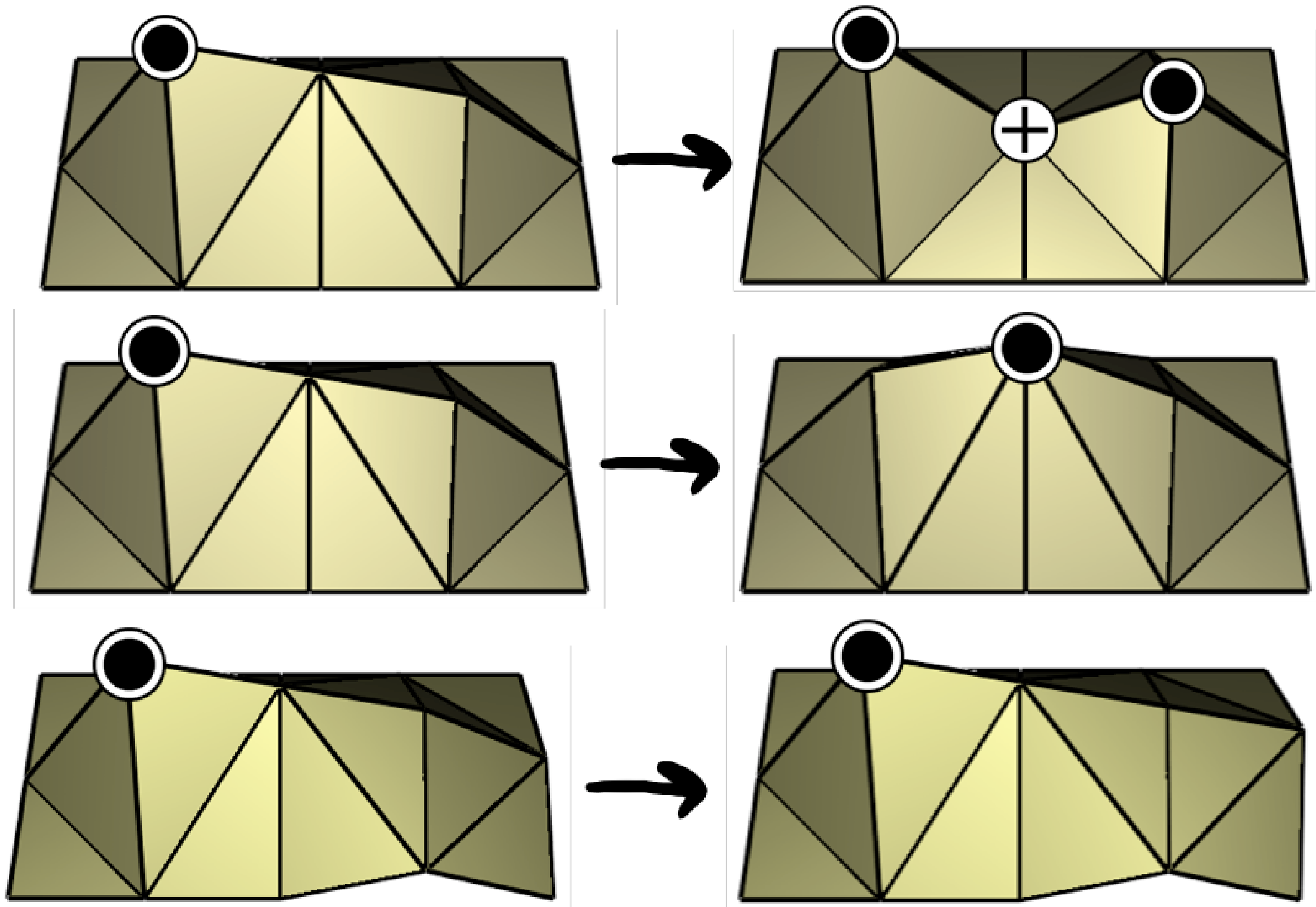


PL topological events

Merge/Split

Move

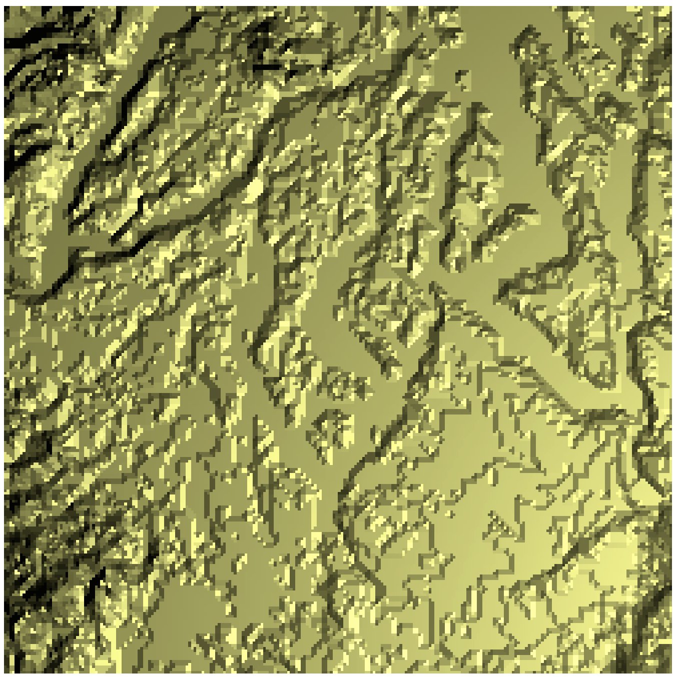
Nothing

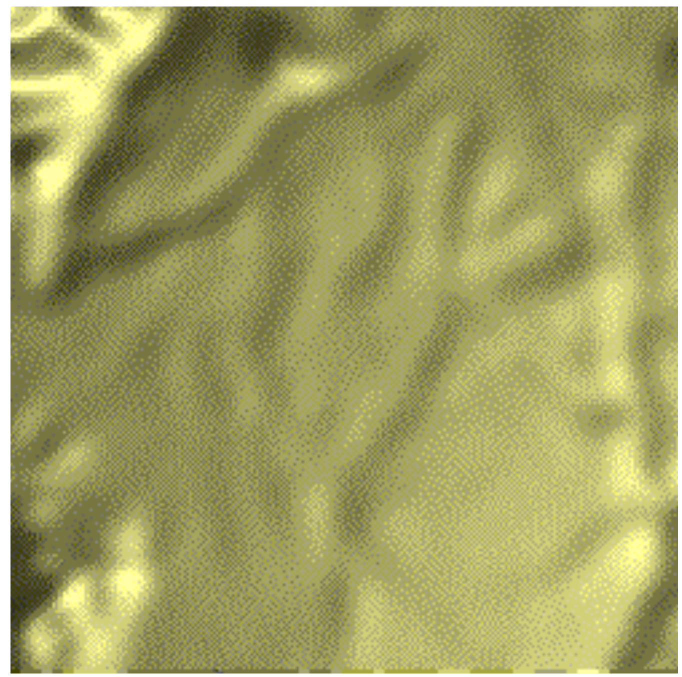


new york university mrl

Laplacian Smoothing

$$I_t = \Delta I$$





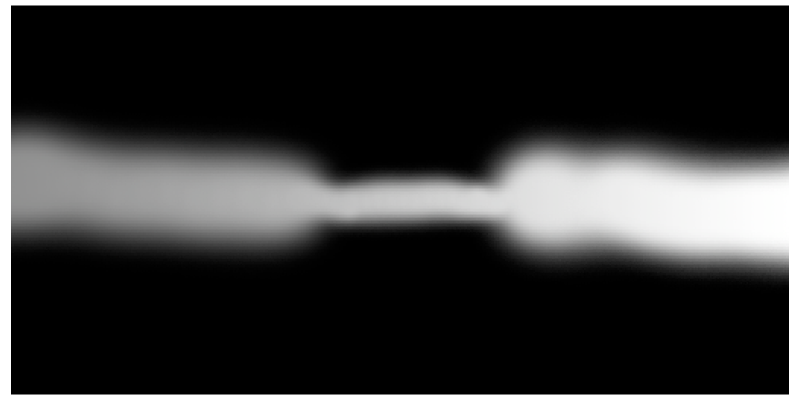
new york university mrl

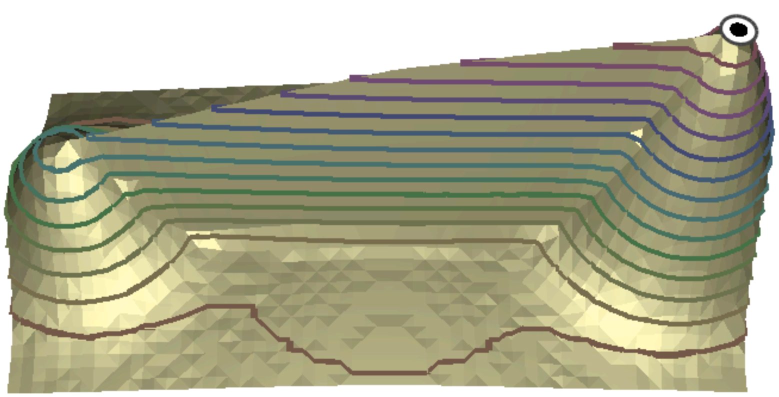
Laplacian Smoothing

Can create features

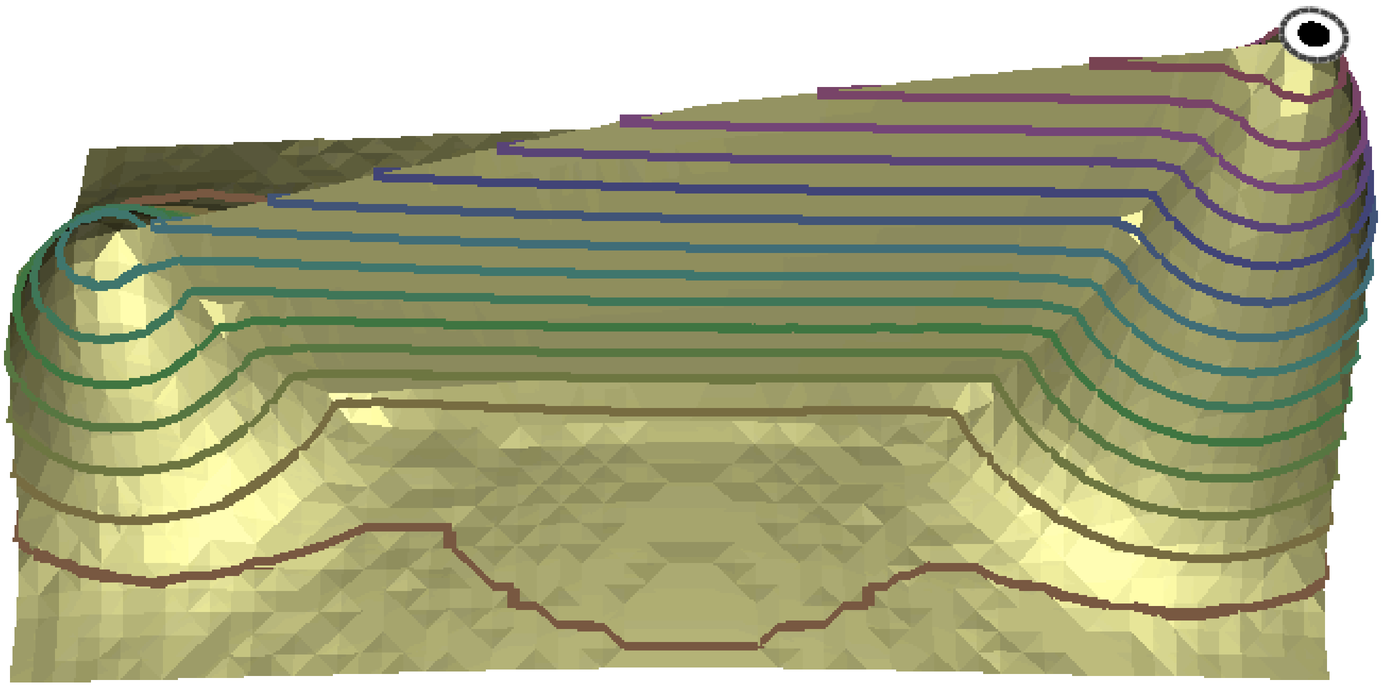
Ridge Bridge **blood vessels**

as in

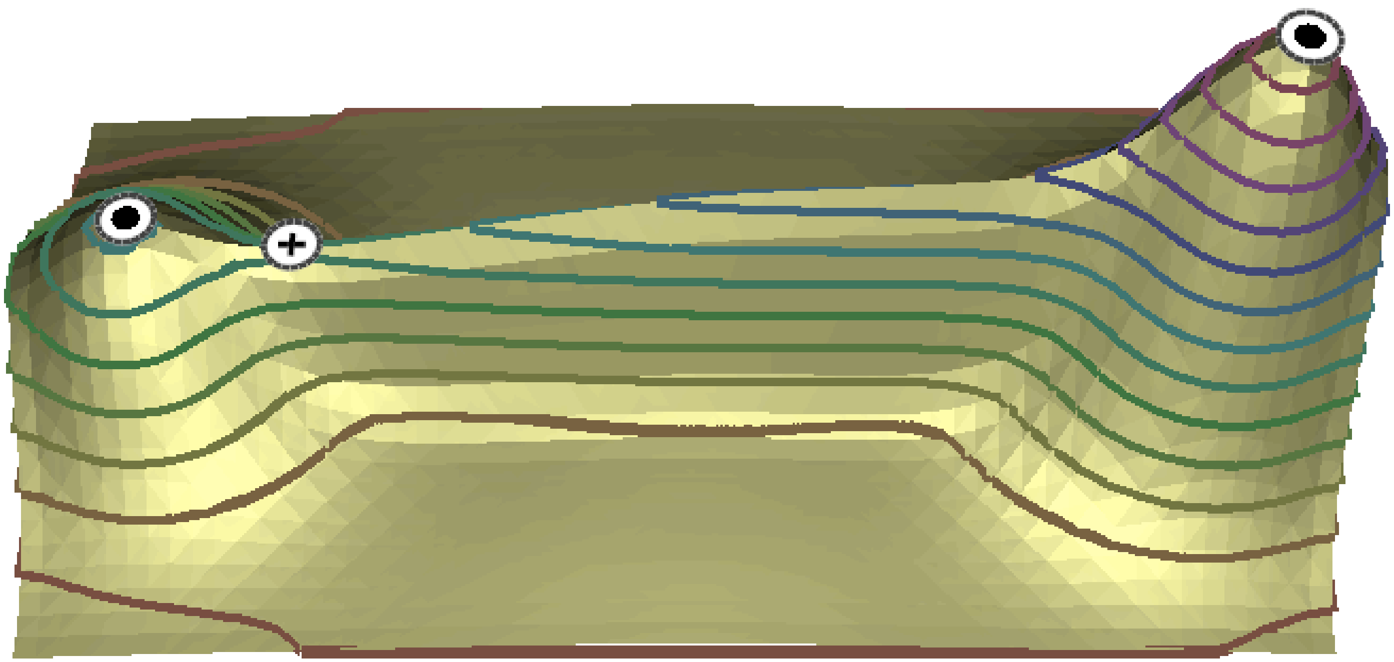




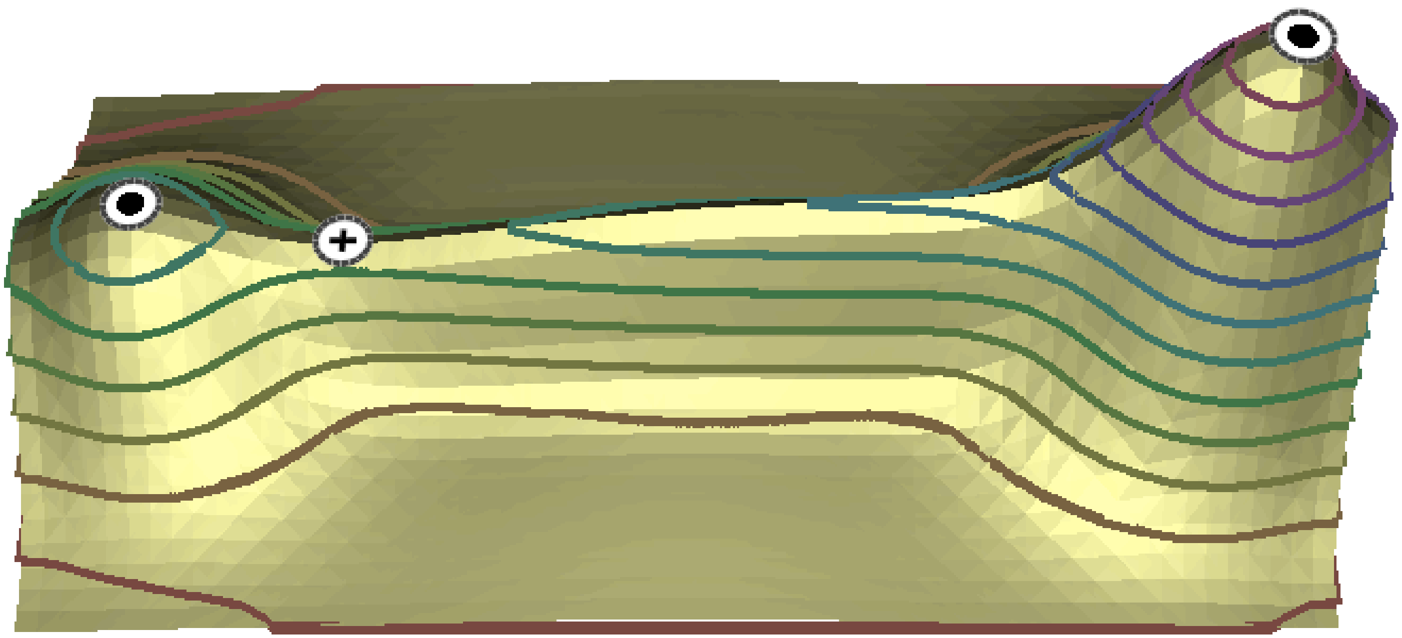
Ridge Bridge (t = 0.0)



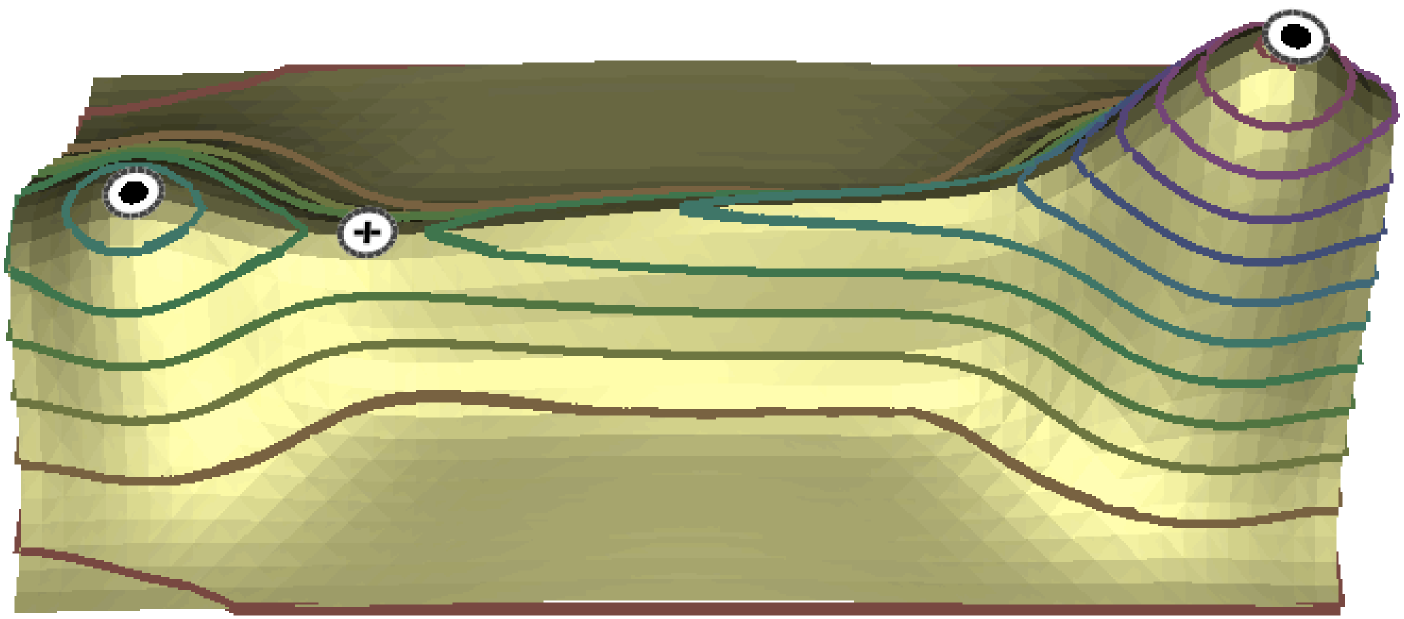
Ridge Bridge (3.5)



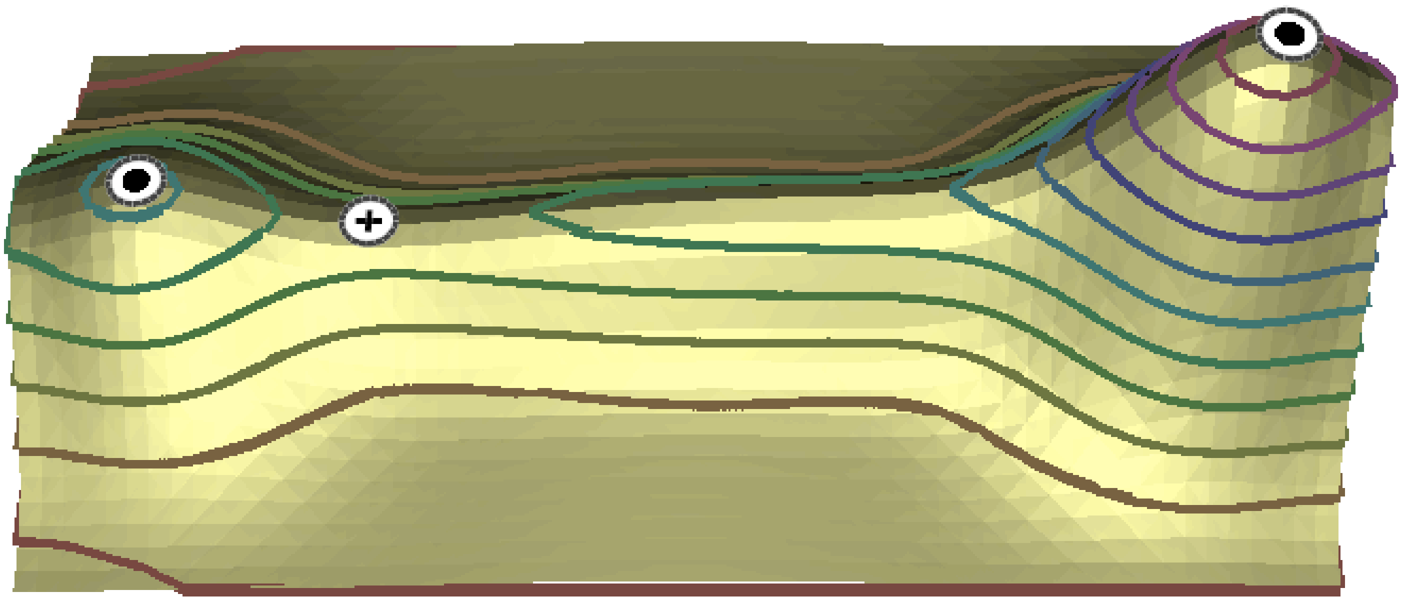
Ridge Bridge (7.0)



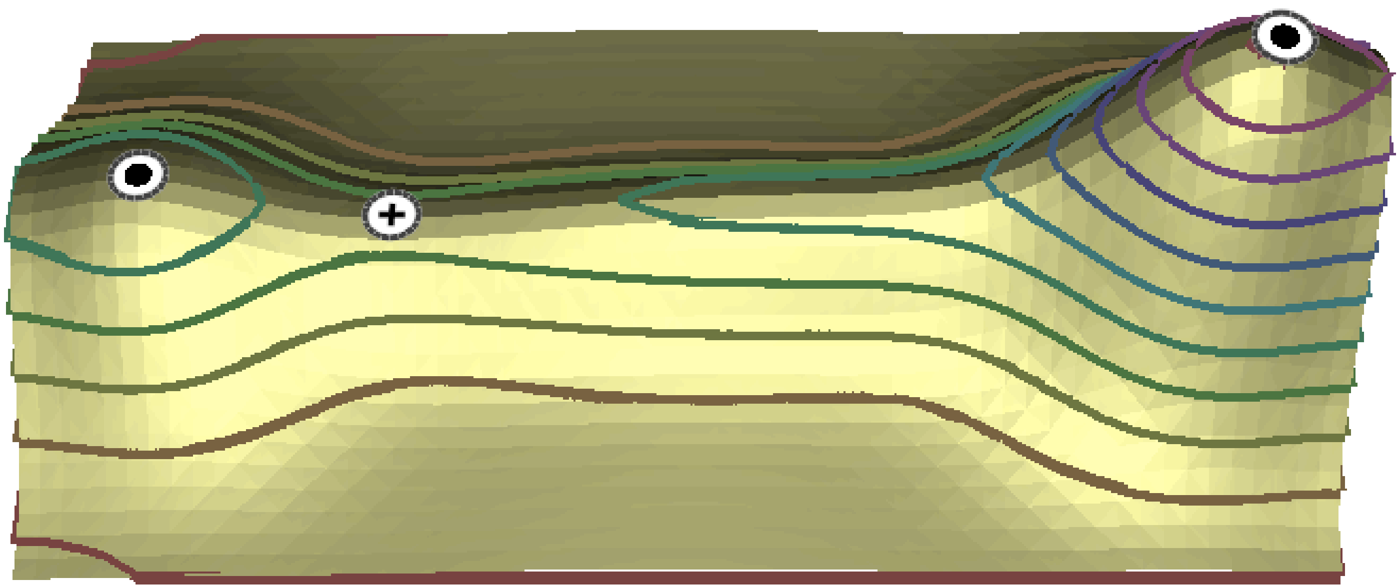
Ridge Bridge (10.5)



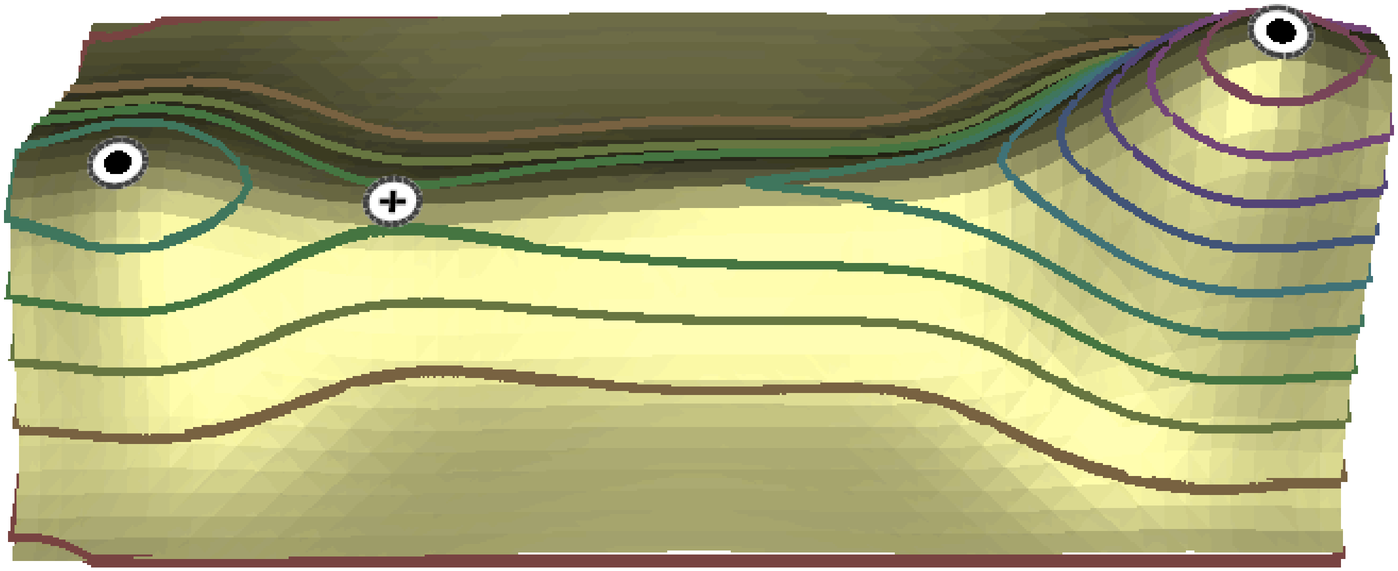
Ridge Bridge (14.0)



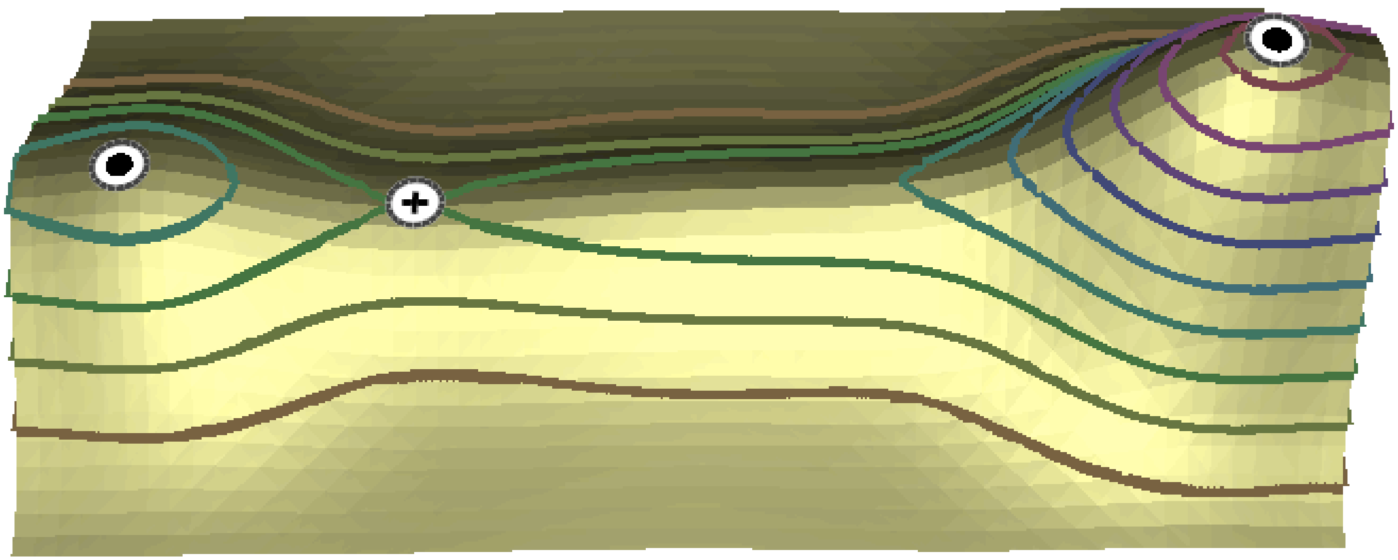
Ridge Bridge (17.5)



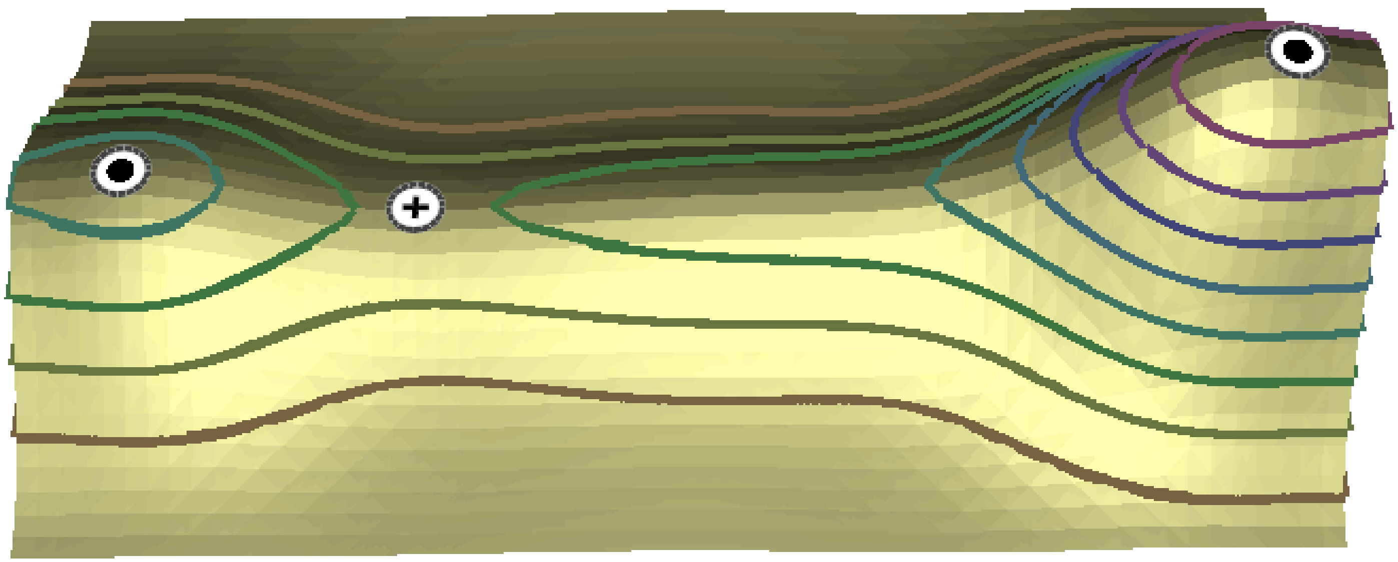
Ridge Bridge (21.0)



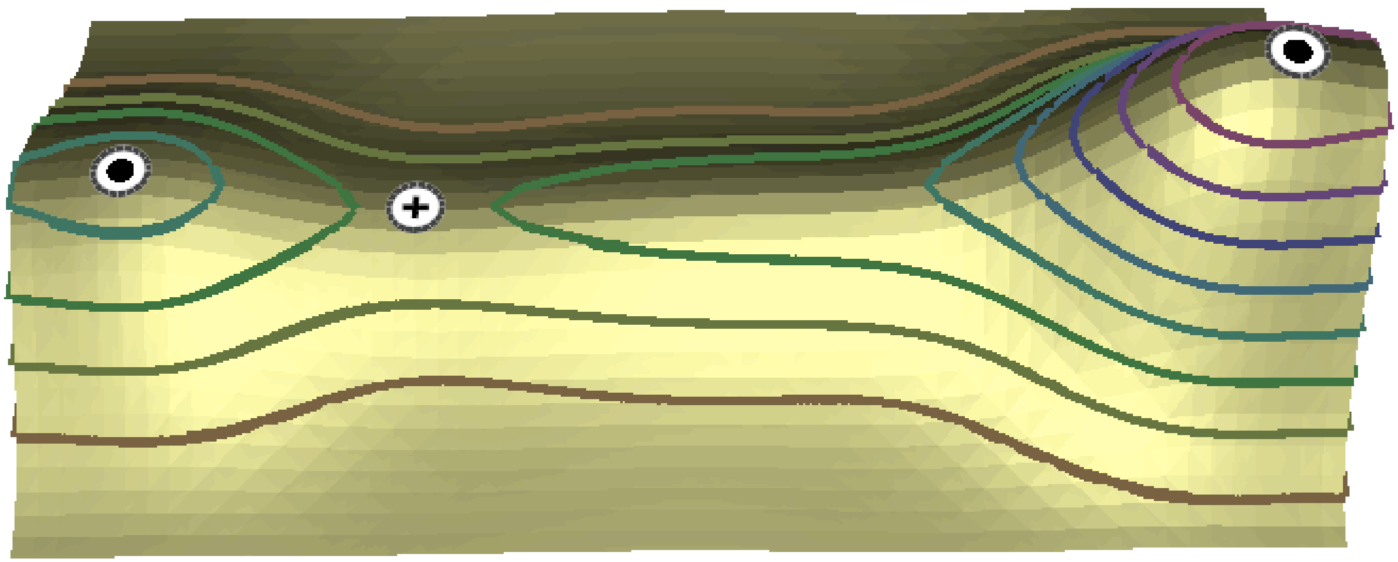
Ridge Bridge (24.5)



Ridge Bridge (28.0)



Ridge Bridge (28.0)

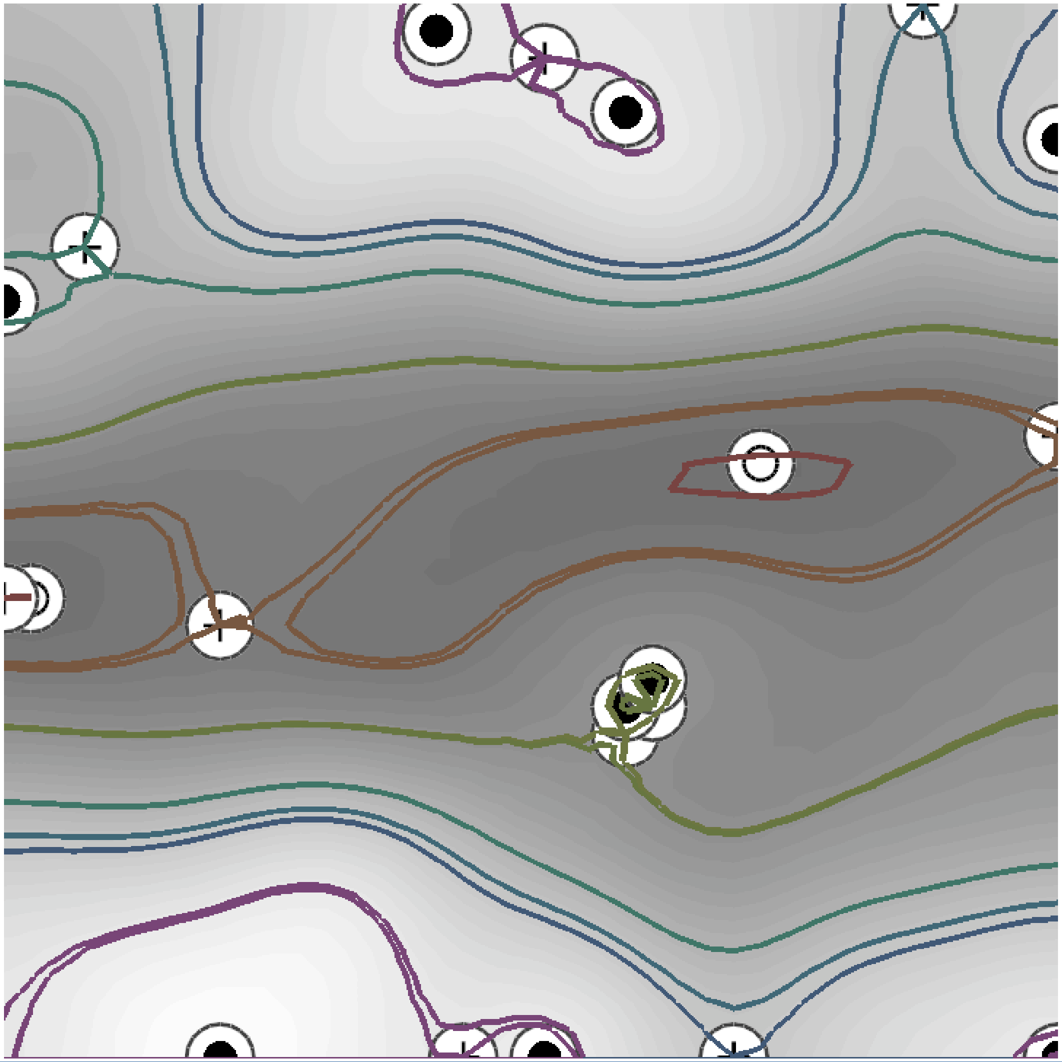


2/3

Ridge Bridge (28.0)

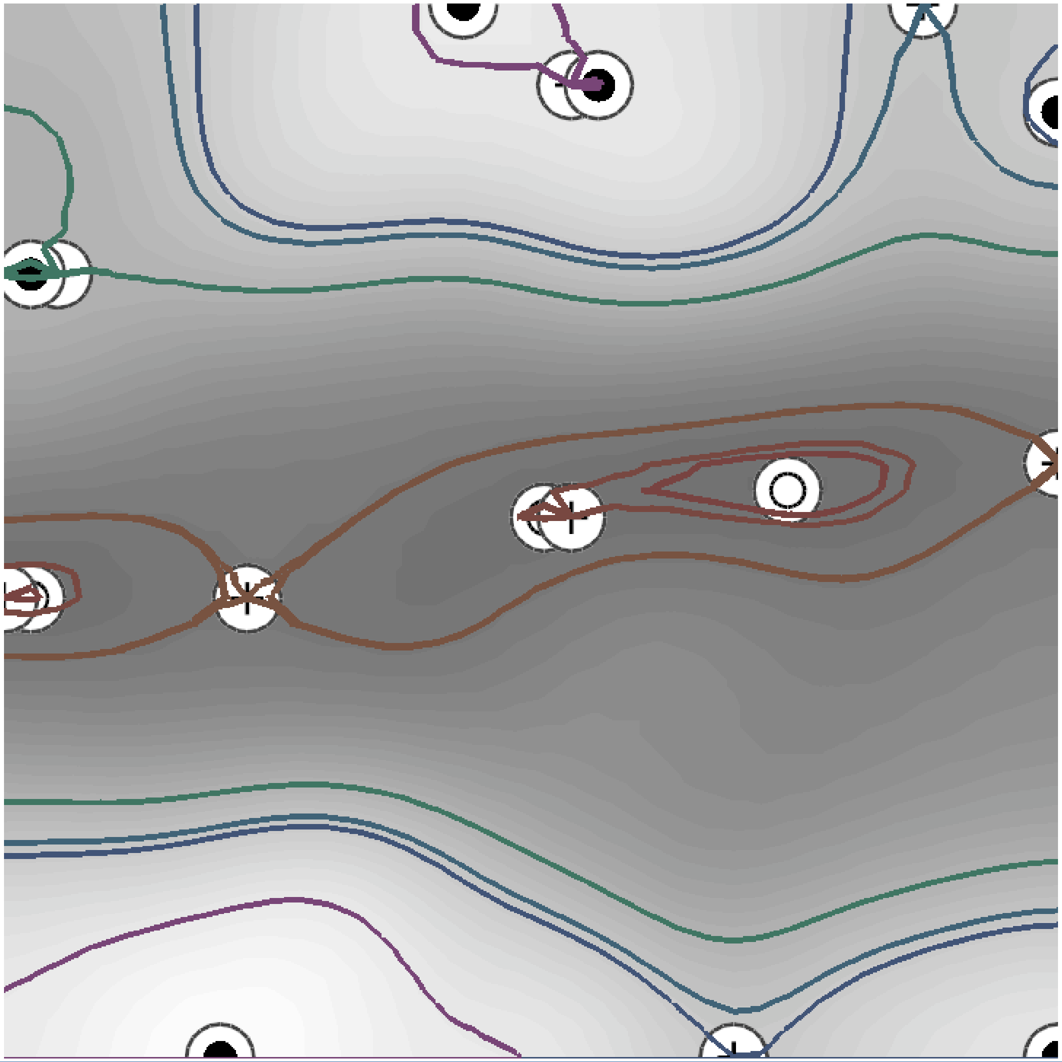
3/3

Puget Sound (3.5)

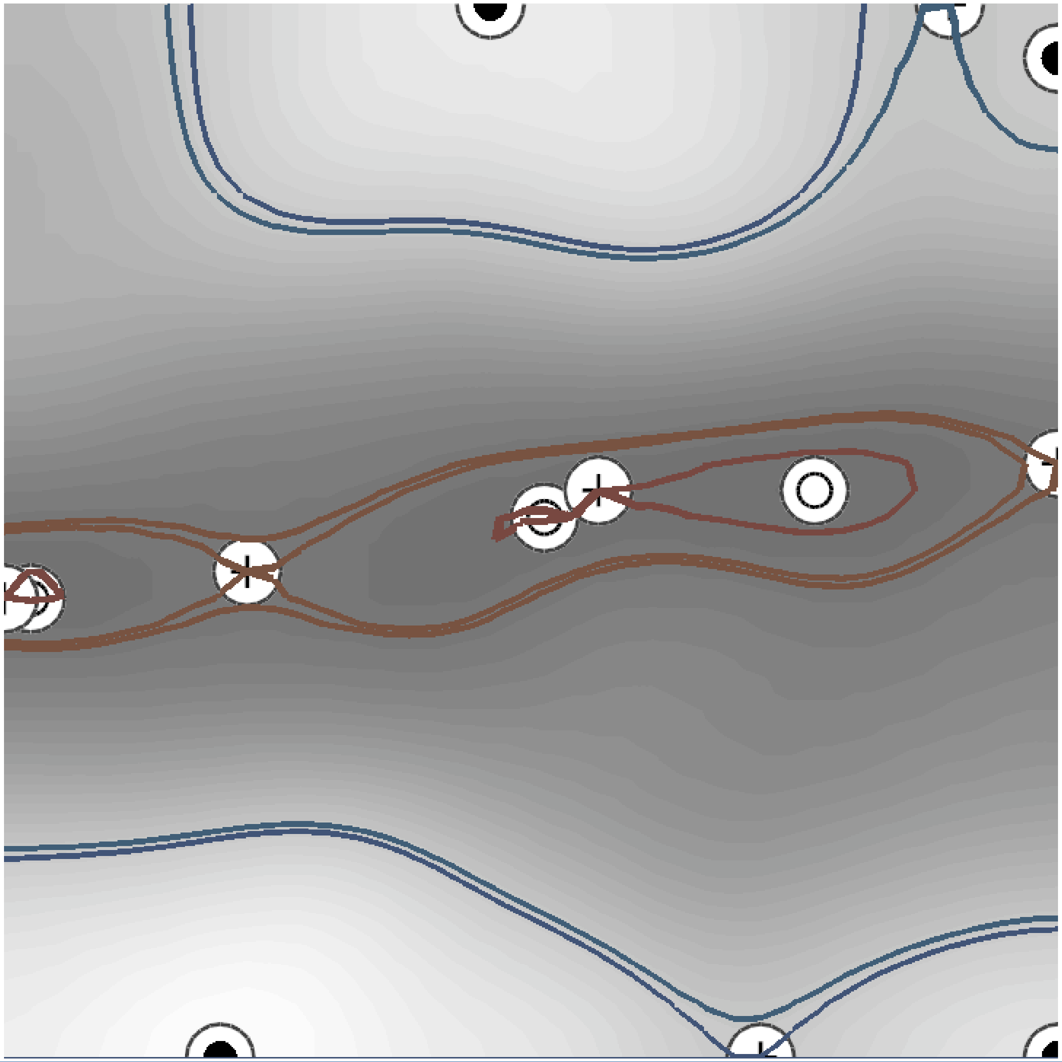


new york university mrl

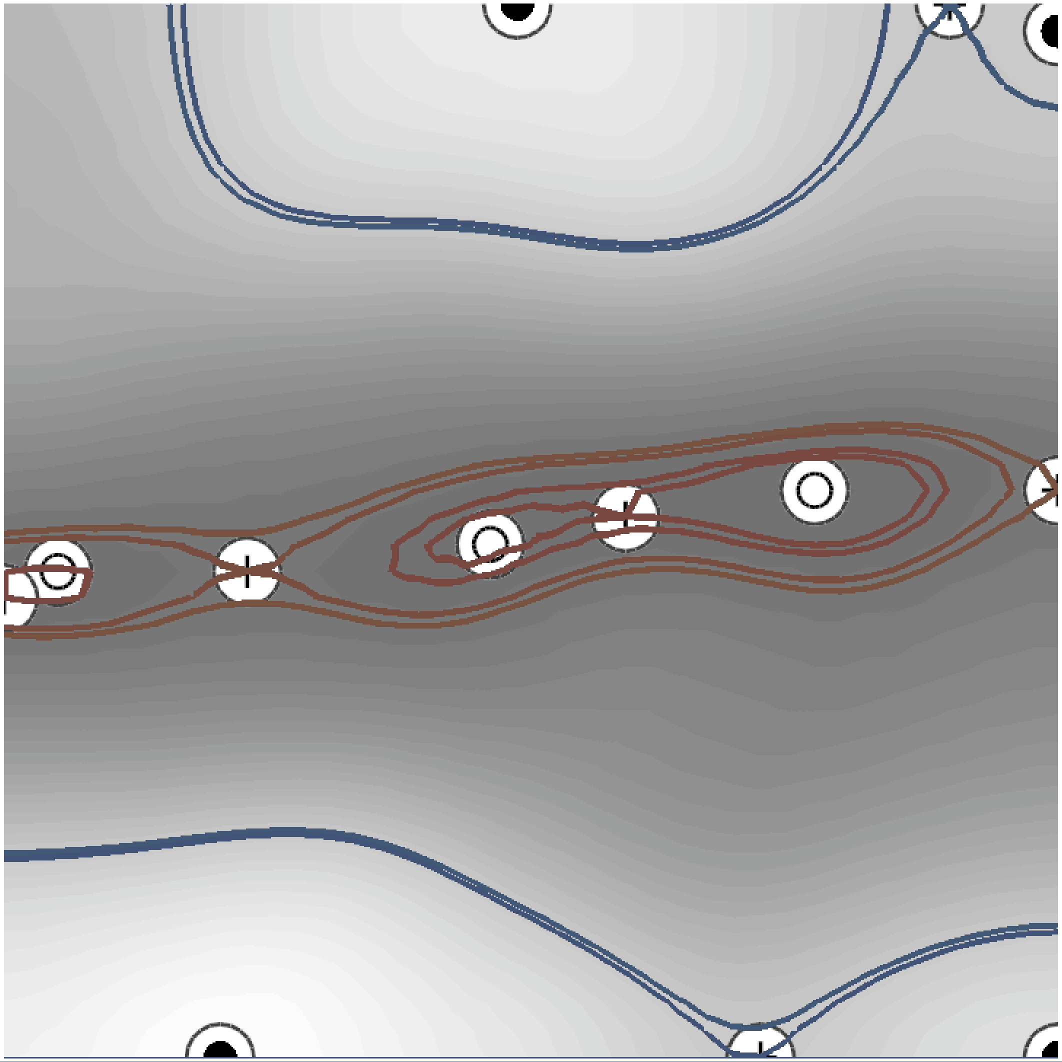
Puget Sound (8.4)



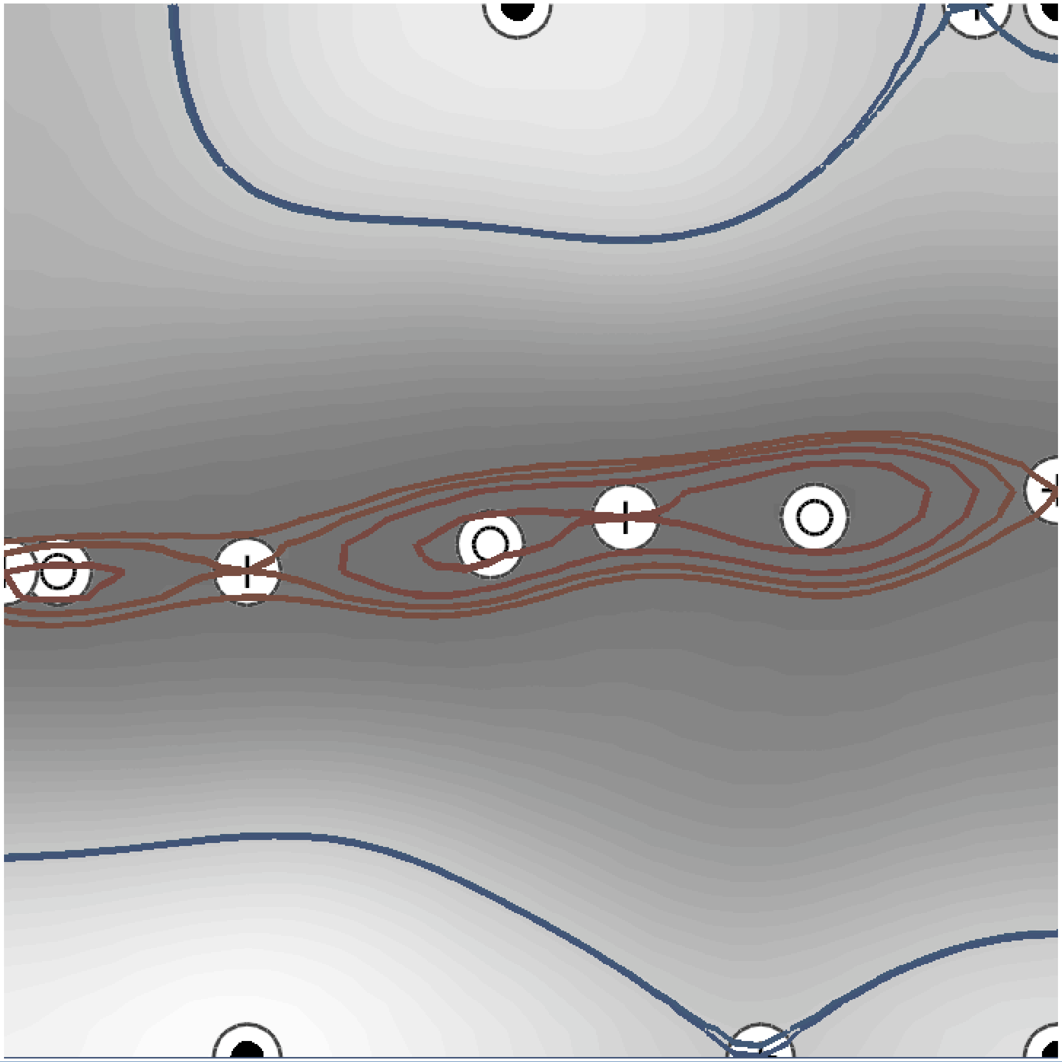
Puget Sound (13.3)



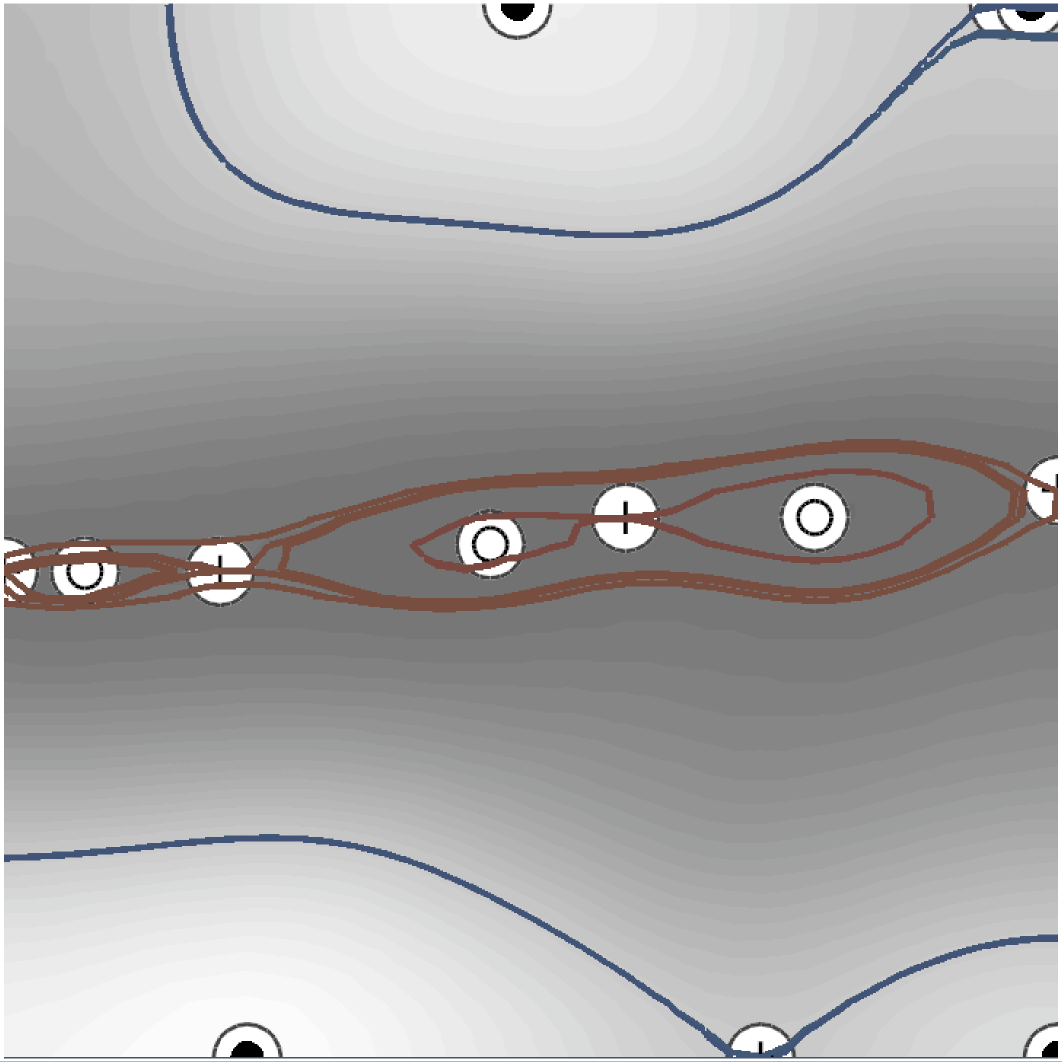
Puget Sound (18.2)



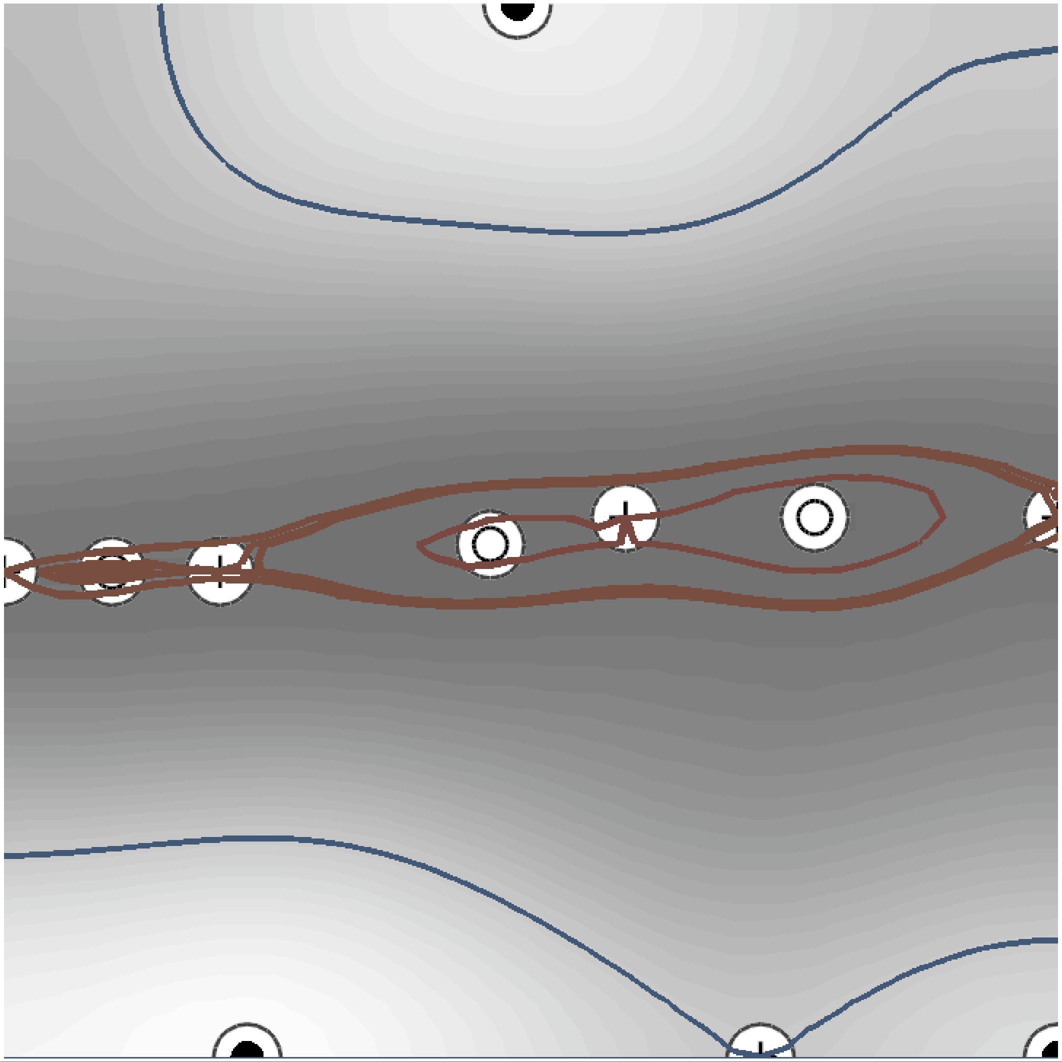
Puget Sound (23.1)



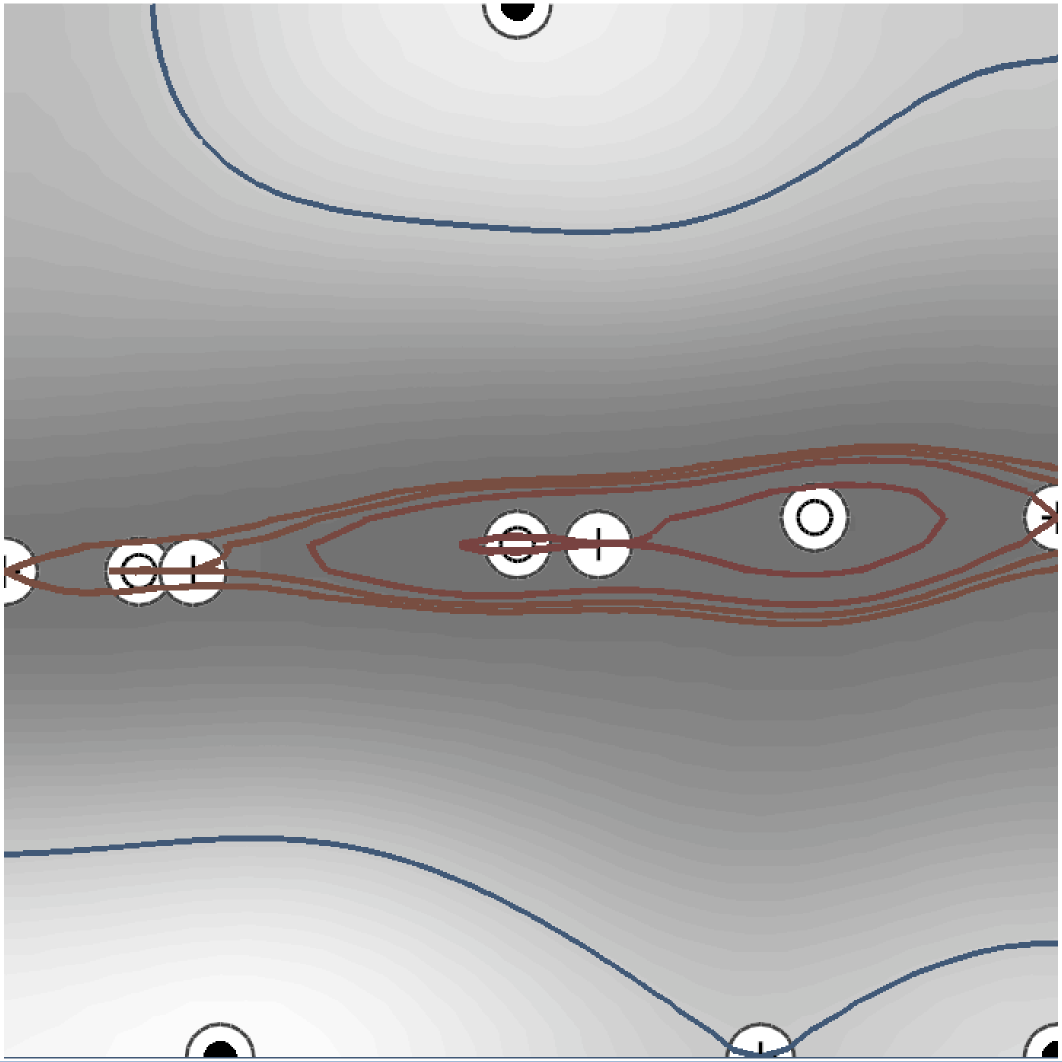
Puget Sound (28.0)



Puget Sound (32.9)

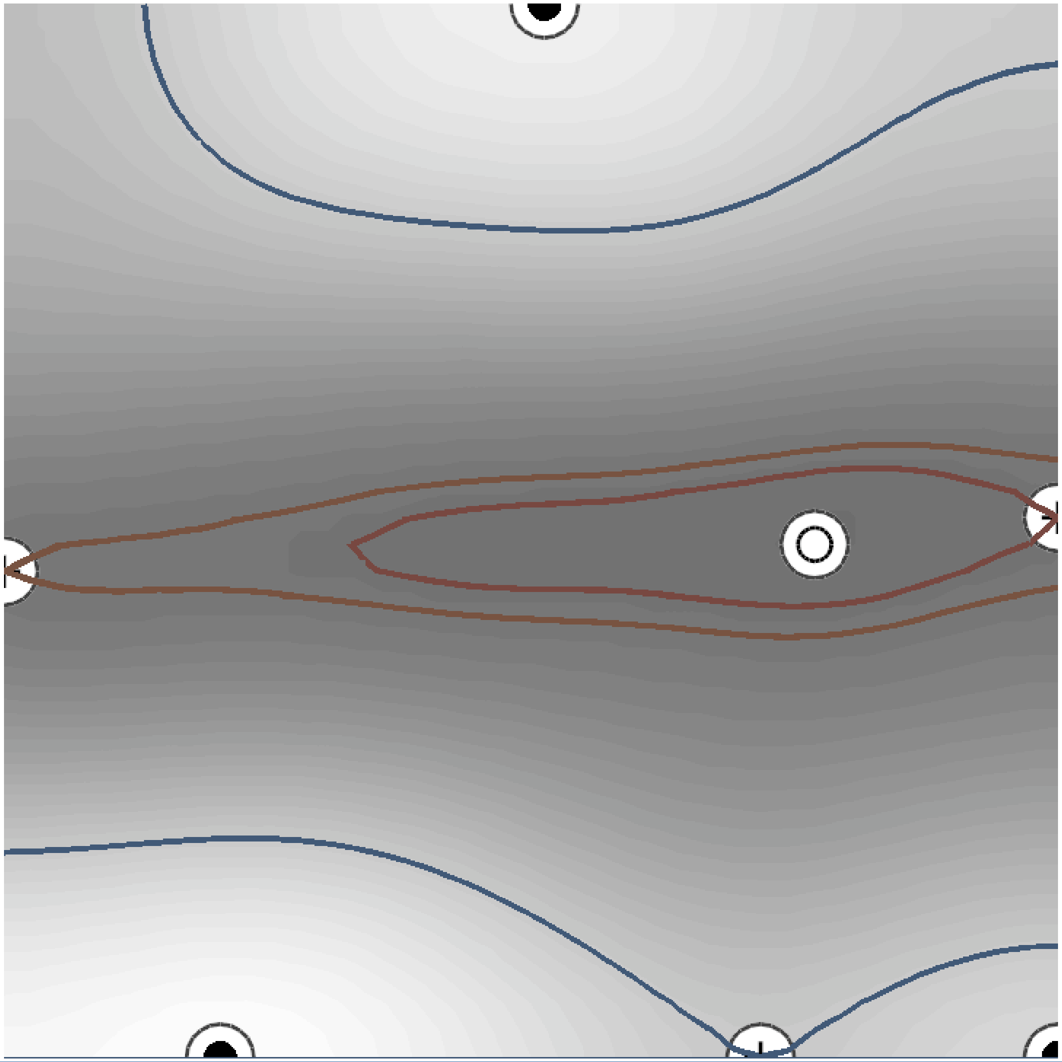


Puget Sound (37.8)

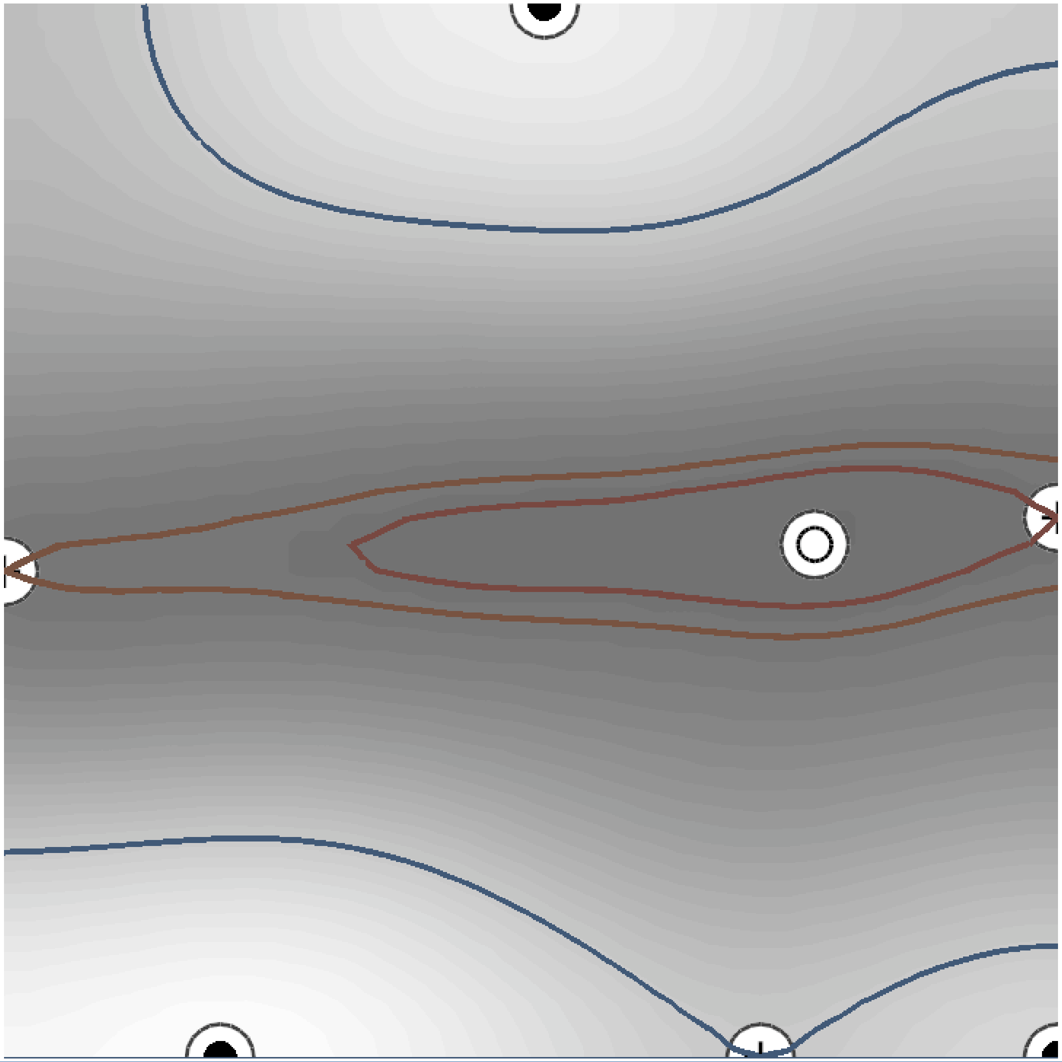


new york university mrl

Puget Sound (42.7)



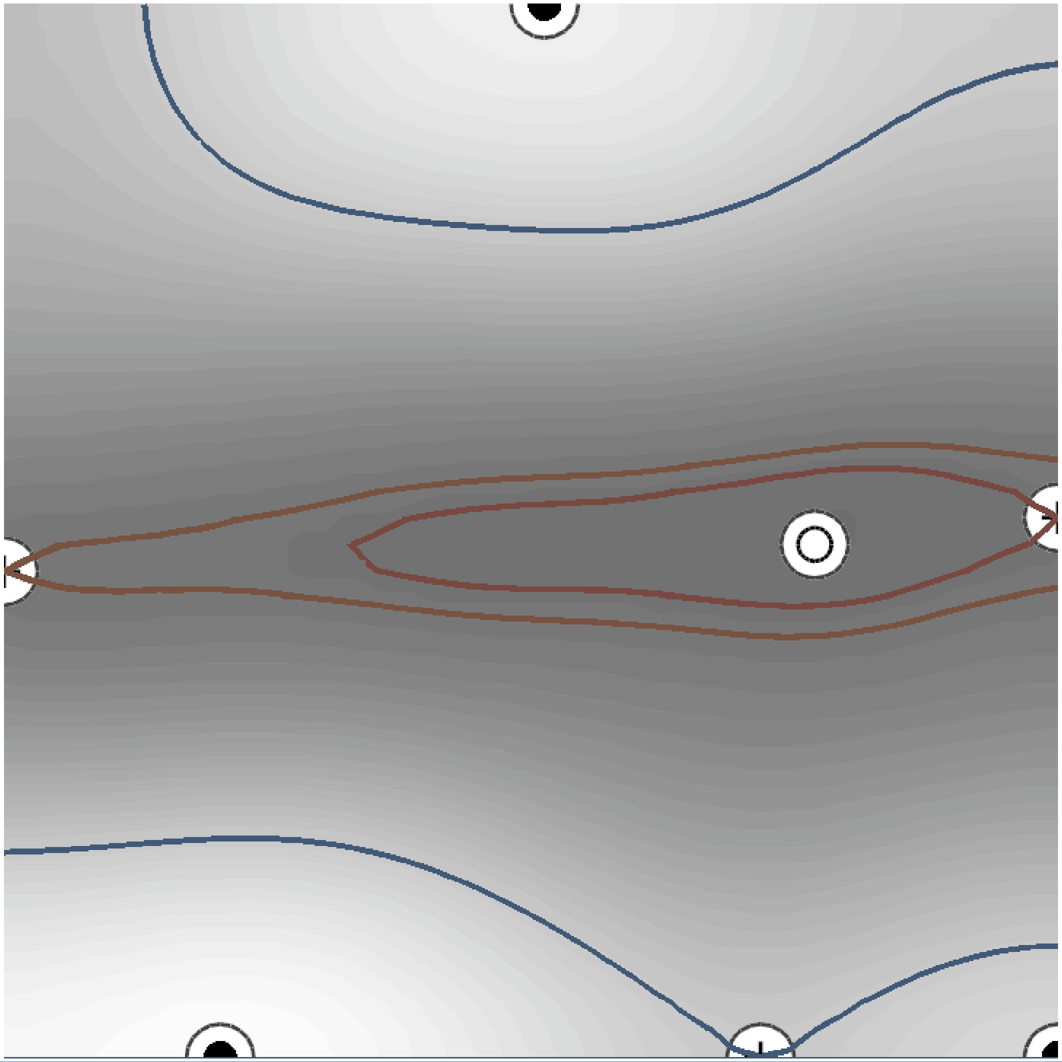
Puget Sound (42.7)



2/3

new york university mrl

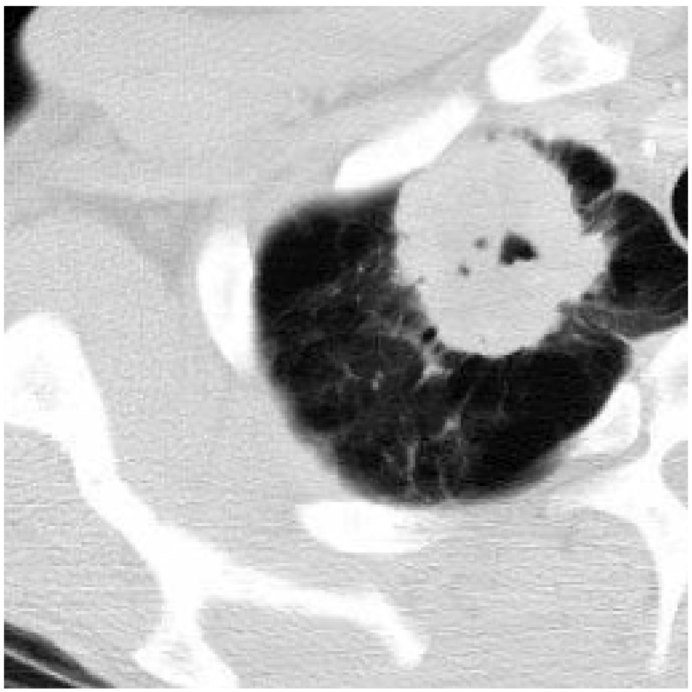
Puget Sound (42.7)

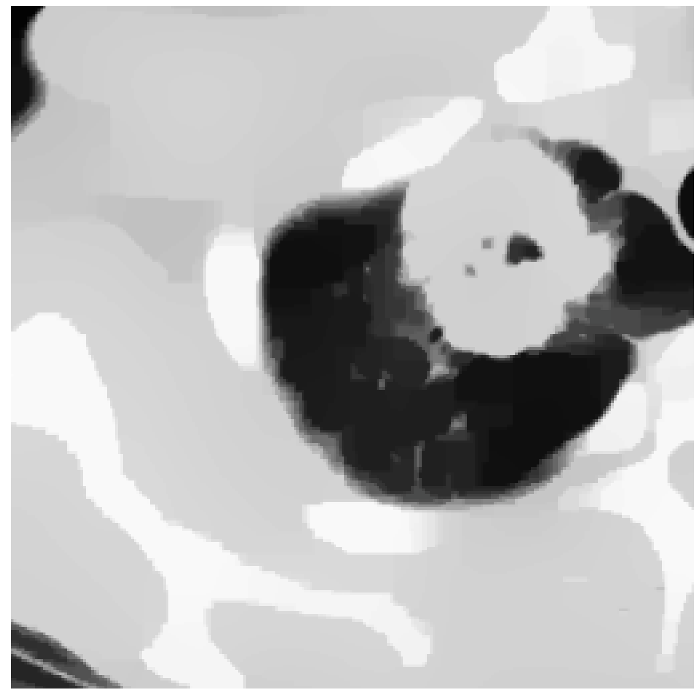


3/3

Anisotropic Smoothing

$$I_t = c(x,y,t)\Delta I + \nabla c \cdot \nabla I$$



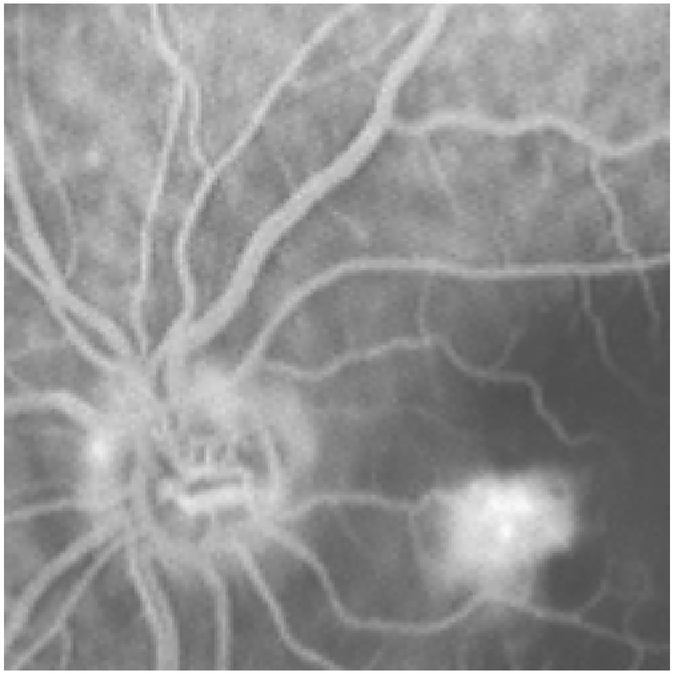


- Can also create features

new york university mrl

Sharpening

$$I_t = I_0 - smoothed(I)$$



- Don't want to create or destroy features

Algorithm

1. Obtain proposed values from the filter function
2. Identify edge flips and sort in time
3. Detect and prevent disallowed events
4. Goto step 1

1. Obtain proposed values

$$\bar{p}_{l+1} = F(\Delta t, p_l)$$

Proposed values at step l+1

Filter function

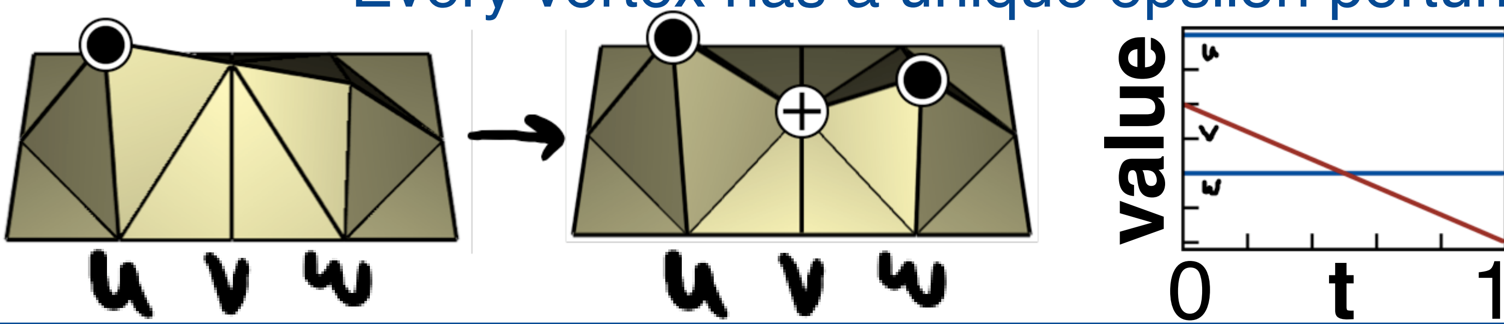
Step size

Current values at step l

2. Identify flips and sort

Function values change linearly from p_l to $\bar{p}_{l+1}$

- An edge between adjacent vertices flips at most once
- Must resolve flip time exactly
  - Otherwise vertex ordering becomes cyclic
  - Every vertex has a unique epsilon perturbation

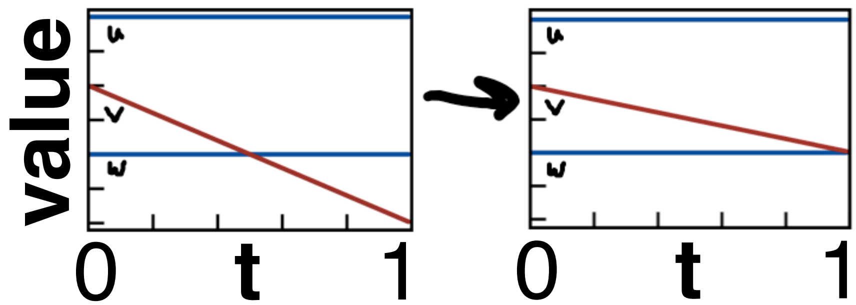


new york university mrl

3. Detect & prevent disallowed events

Process events in order

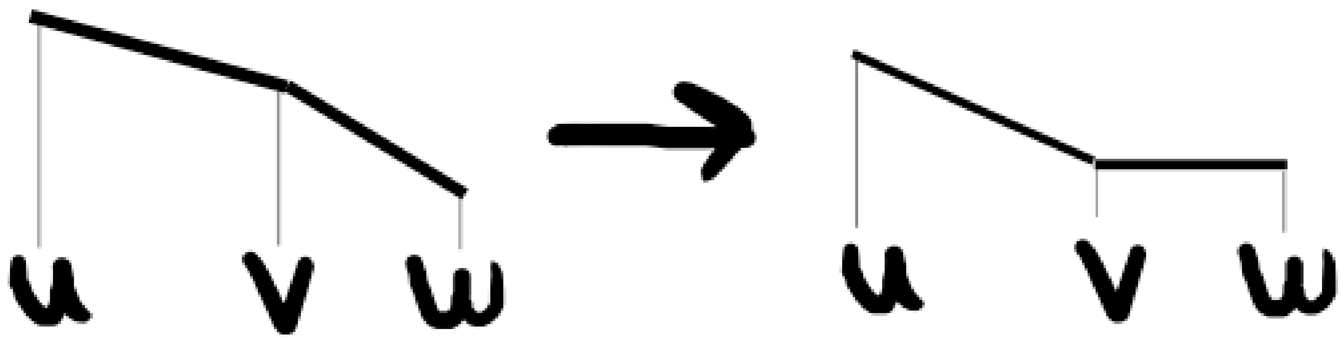
- If we reach a disallowed event between vertices (v, w), set $\bar{p}(v), \bar{p}(w)$ to values infinitesimally before the event



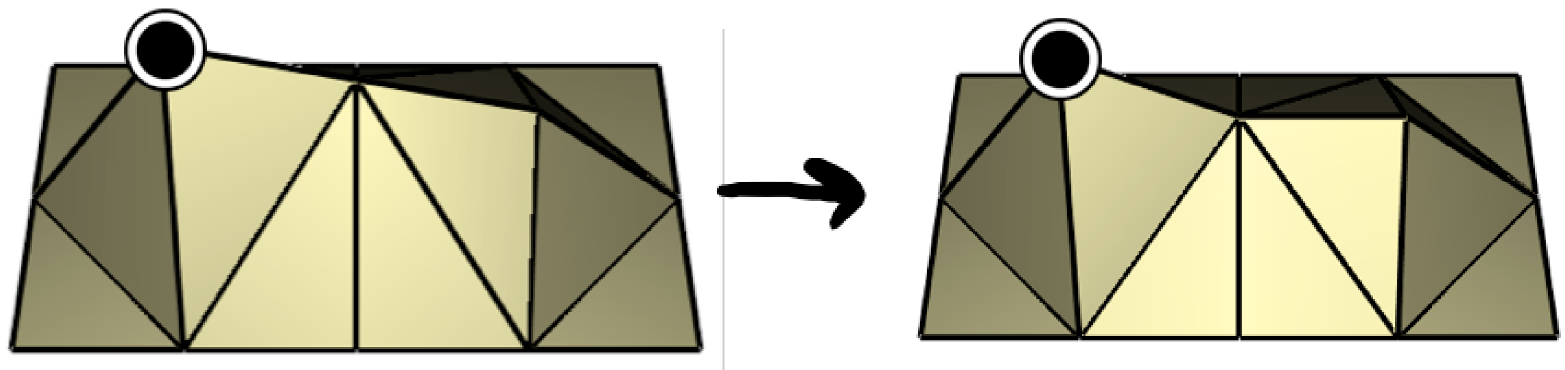
- Must re-identify events involving *v,w*
- May need to rewind the event queue

3. Detect & prevent disallowed events

1D:



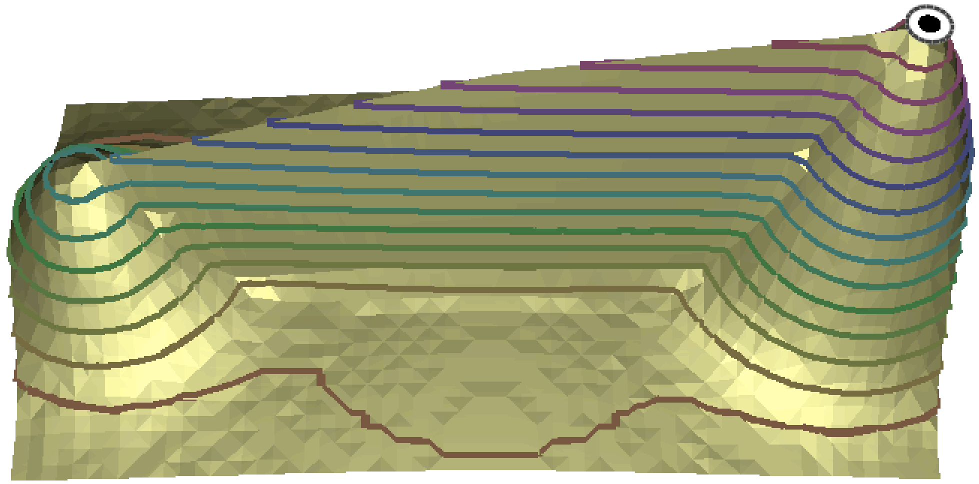
2D:

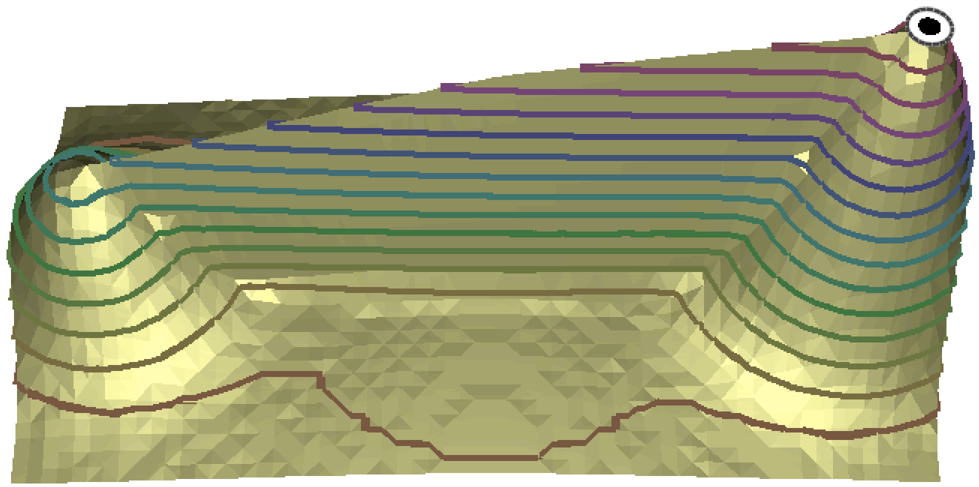


Algorithm

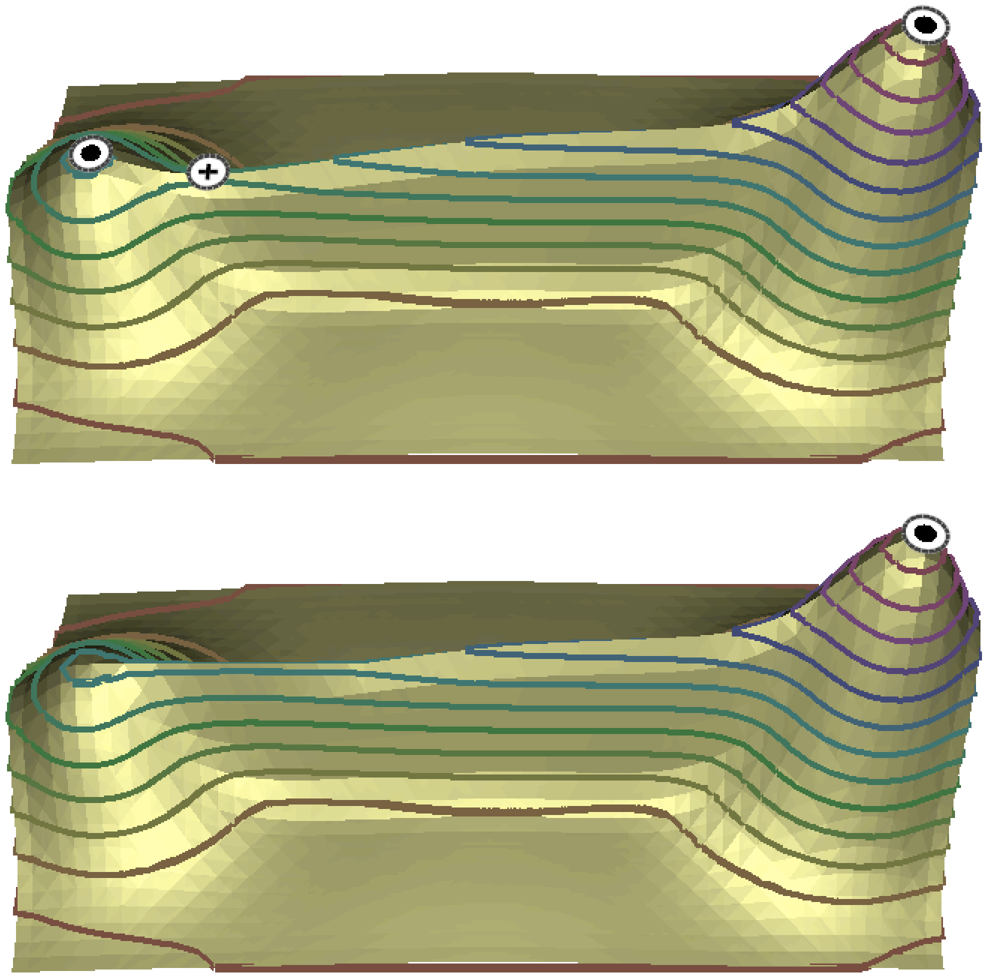
1. Obtain proposed values from the filter function
2. Identify edge flips and sort in time
3. Detect and prevent disallowed events
4. Goto step 1

Results: Ridge Bridge (0.0)



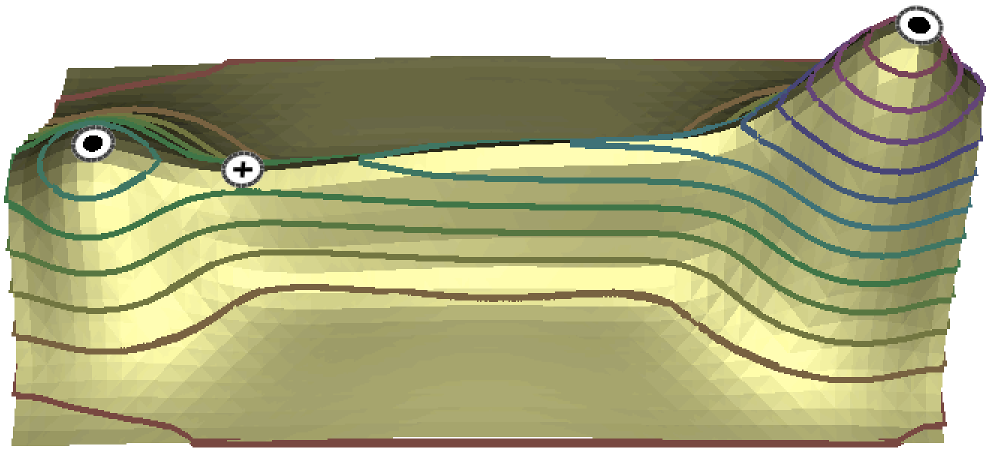


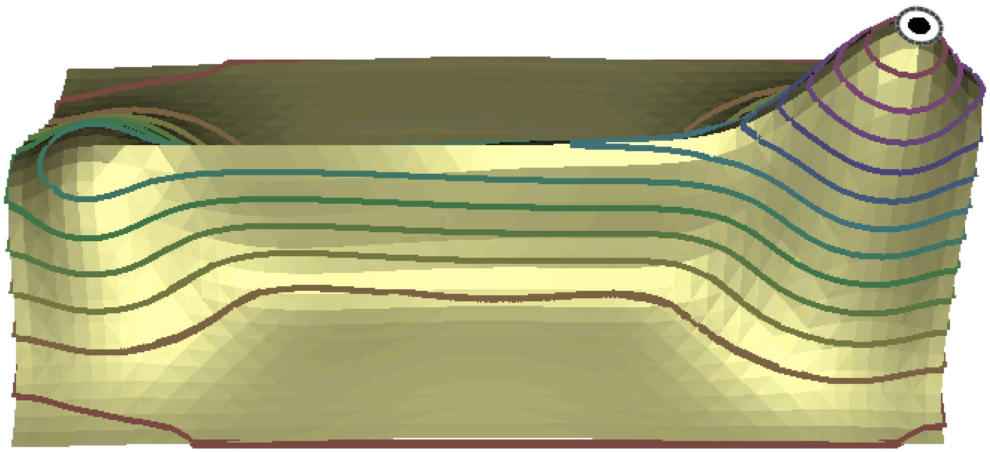
Results: Ridge Bridge (3.5)



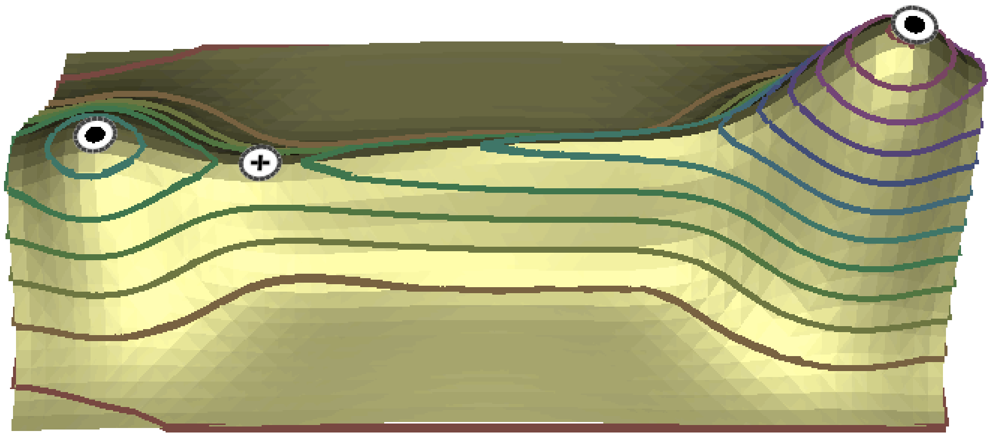
new york university mrl

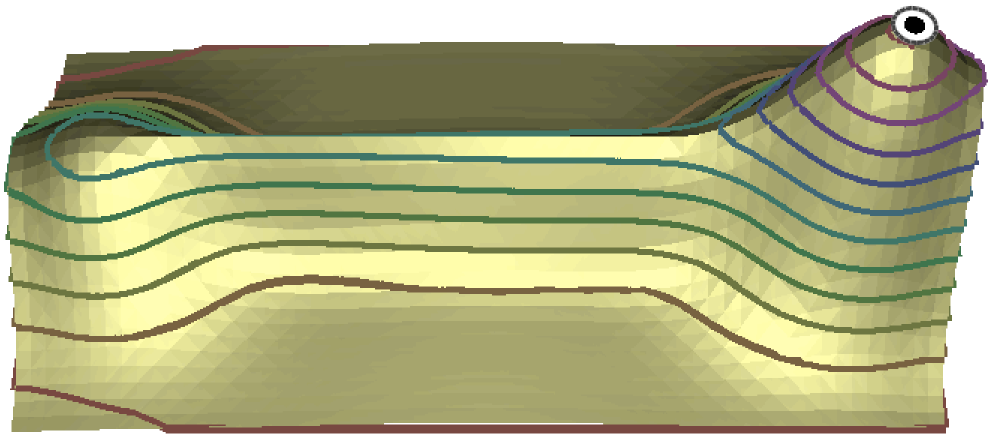
Results: Ridge Bridge (7.0)



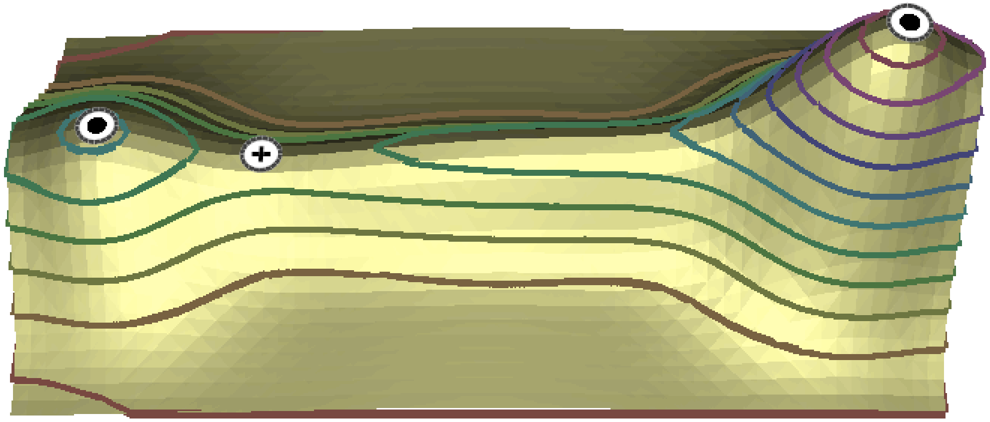


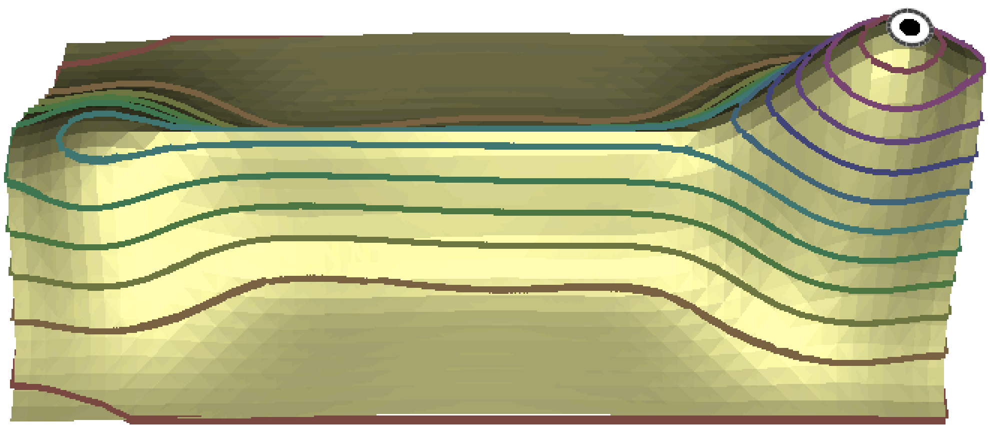
Results: Ridge Bridge (10.5)



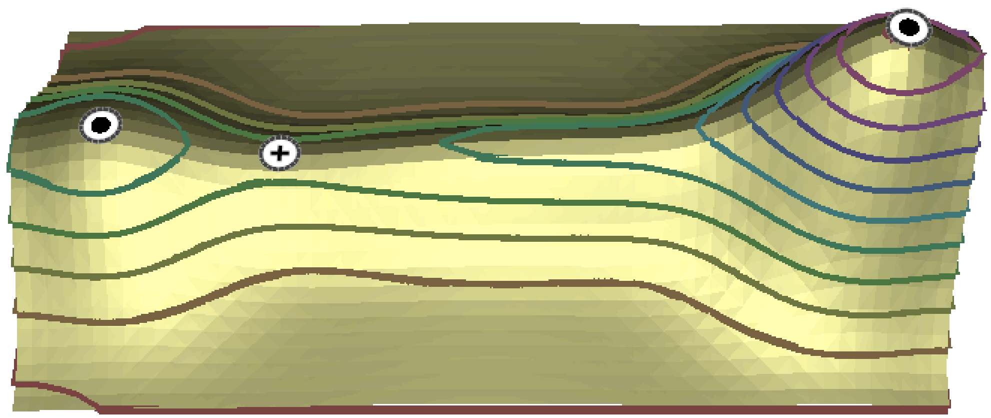


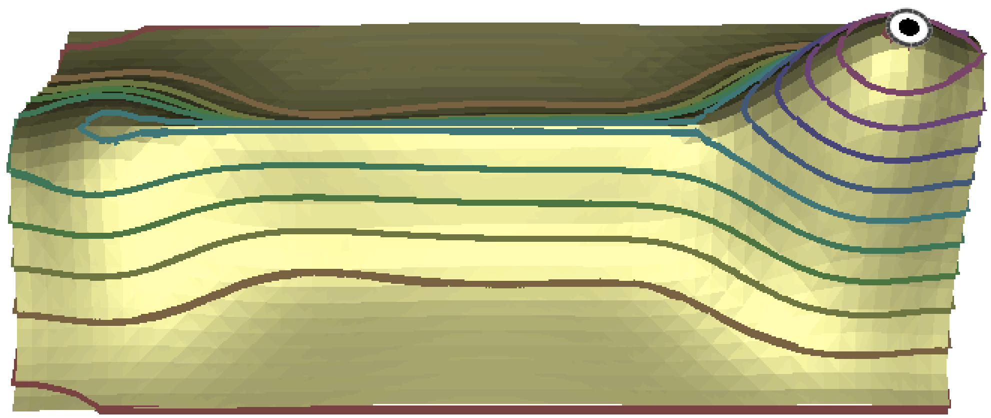
Results: Ridge Bridge (14.0)



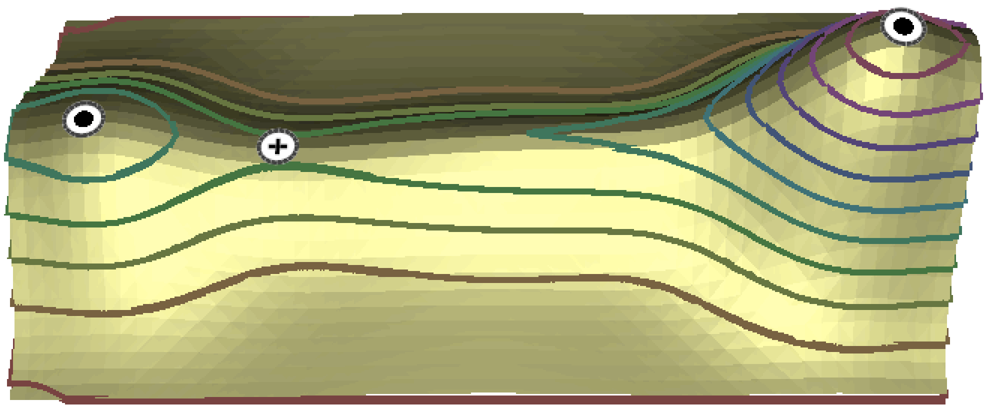


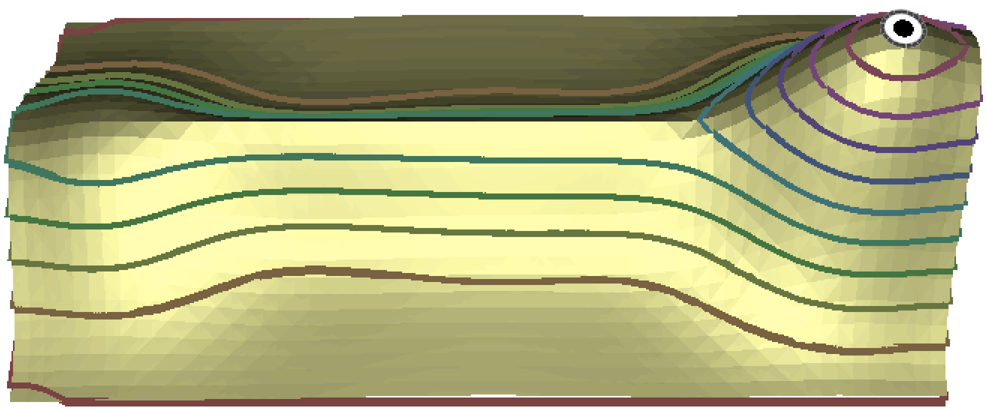
Results: Ridge Bridge (17.5)





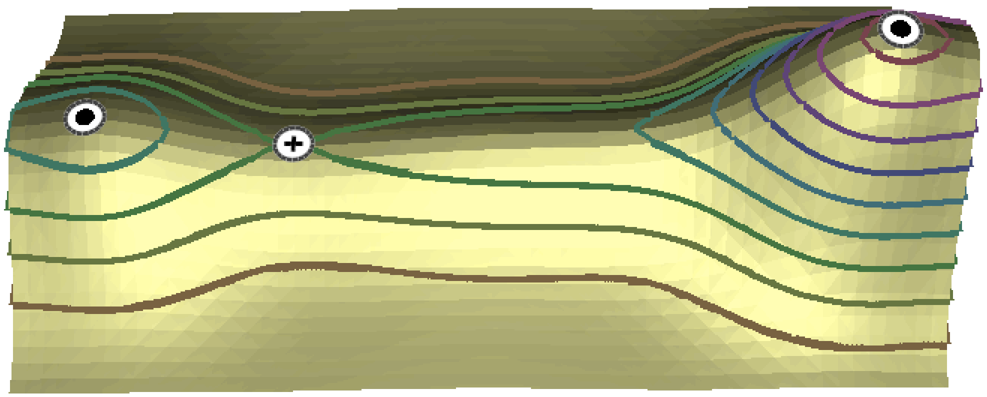
Results: Ridge Bridge (21.0)

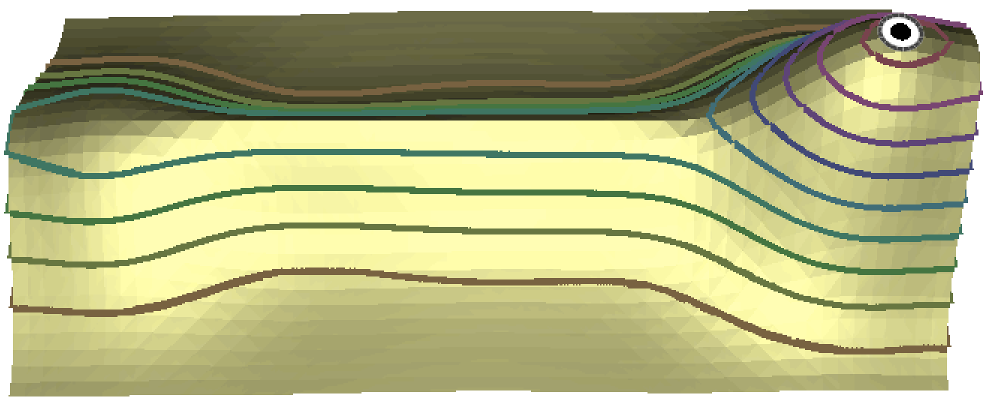




new york university mrl

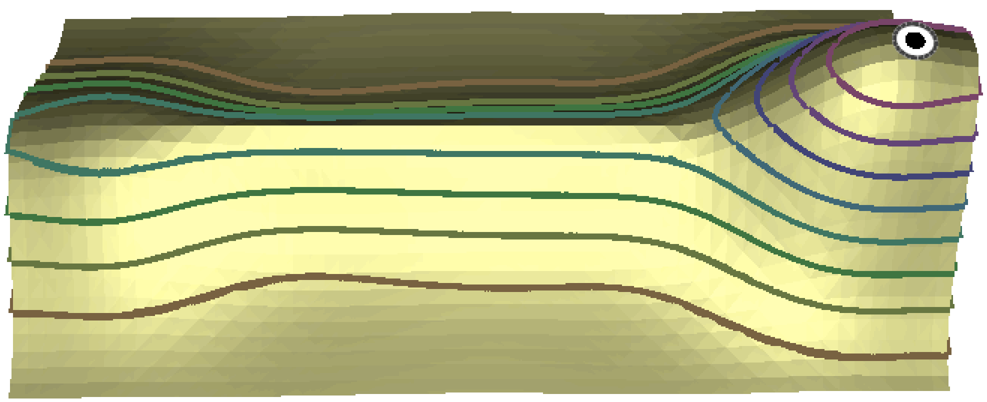
Results: Ridge Bridge (24.5)



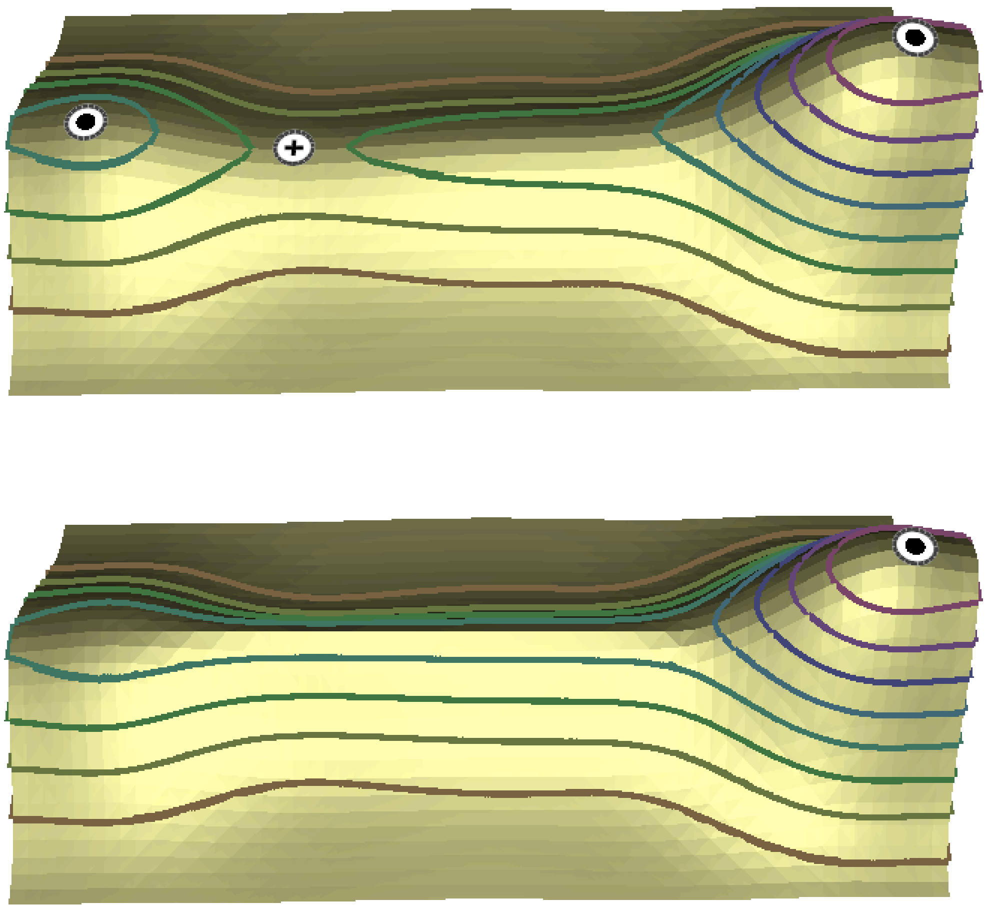


new york university mrl

Results: Ridge Bridge (28.0)

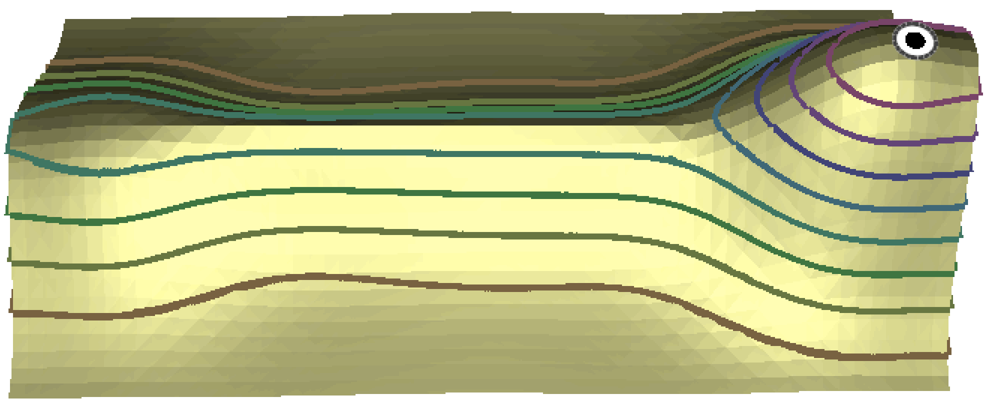


Results: Ridge Bridge (28.0)



2/3

Results: Ridge Bridge (28.0)

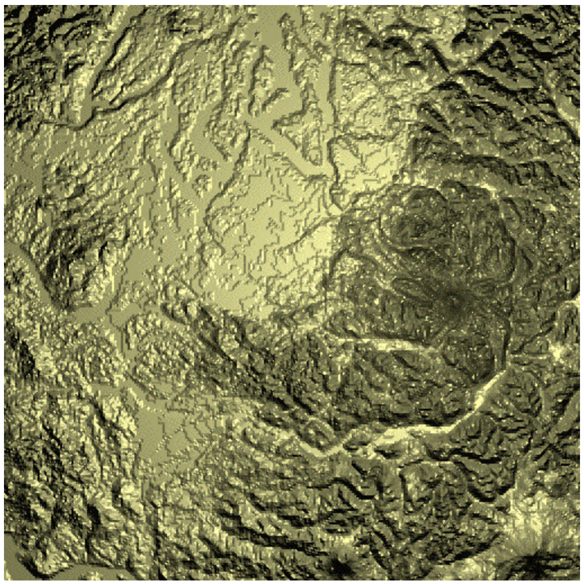


3/3

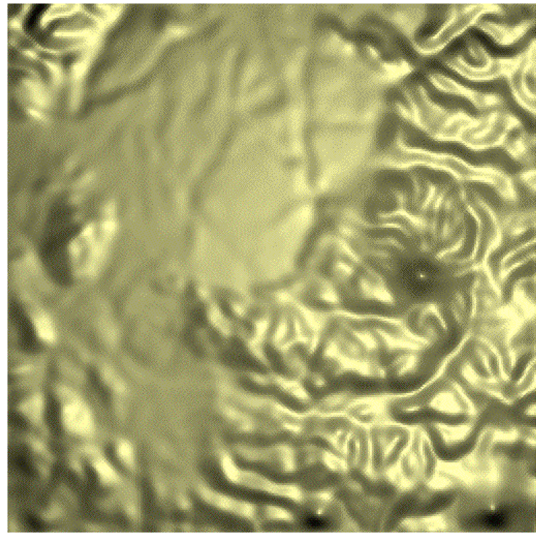
new york university mrl

Results: Laplacian Smoothing

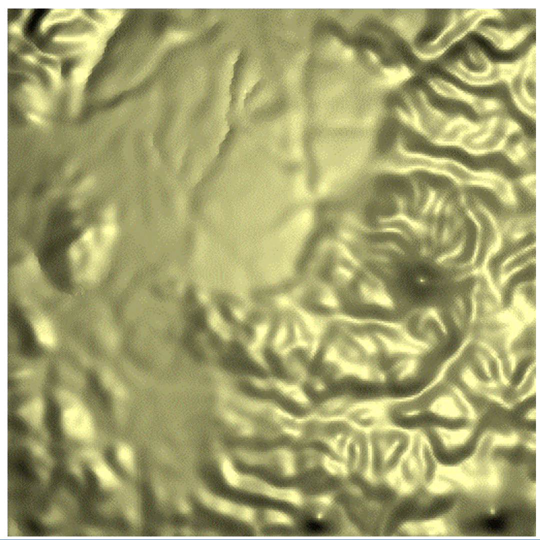
original



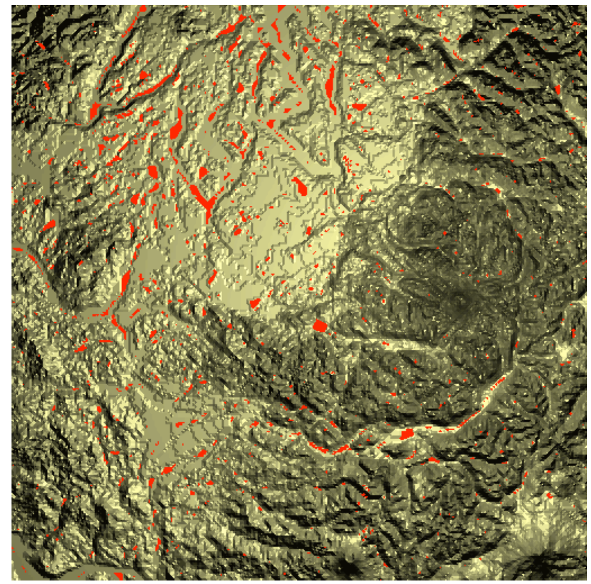
uncontrolled



controlled



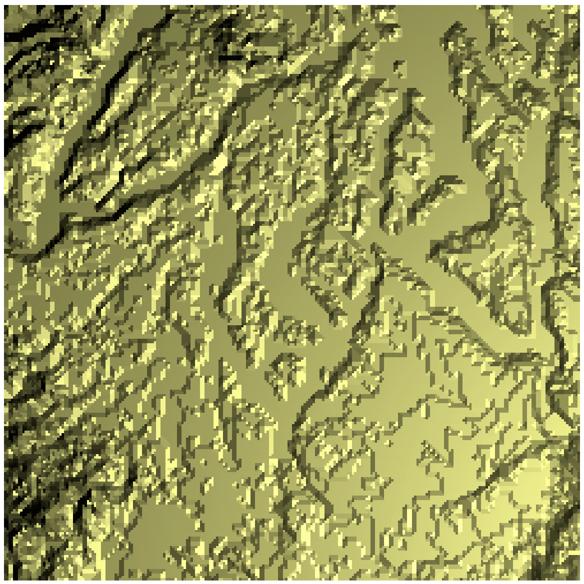
suppressed



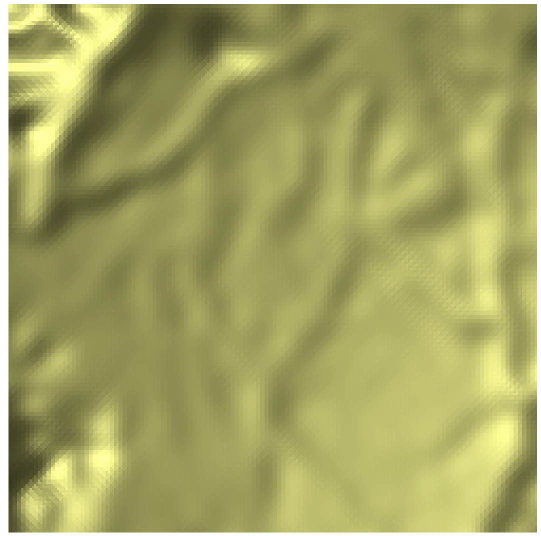
new york university mrl

Results: Laplacian Smoothing

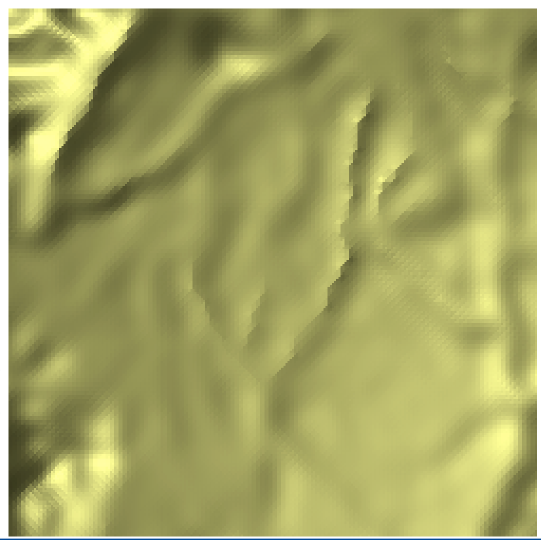
original



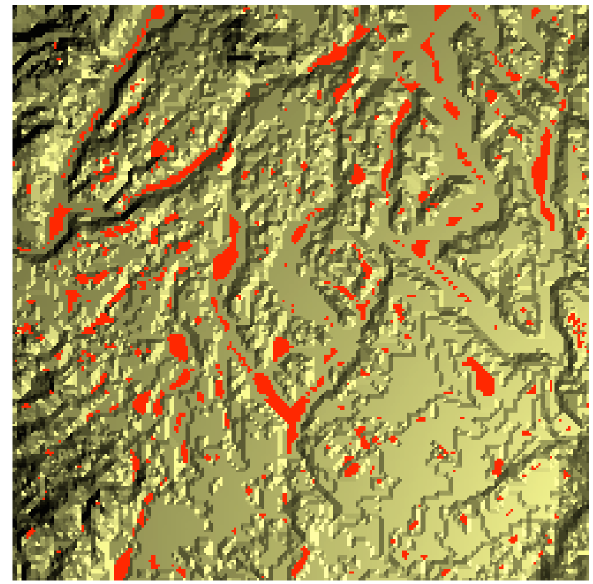
uncontrolled



controlled

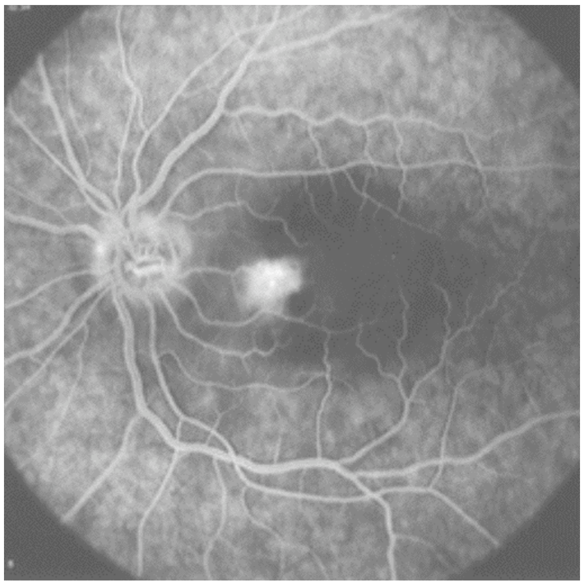


suppressed

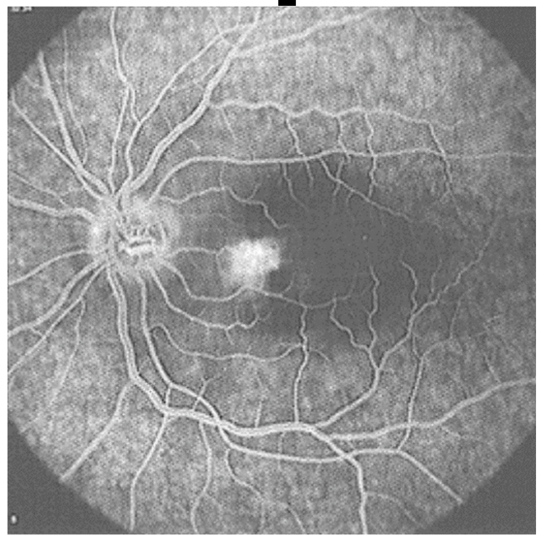


Results: Sharpening

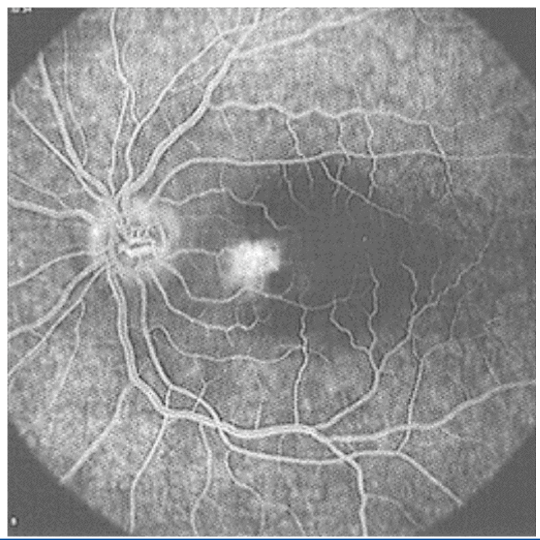
original



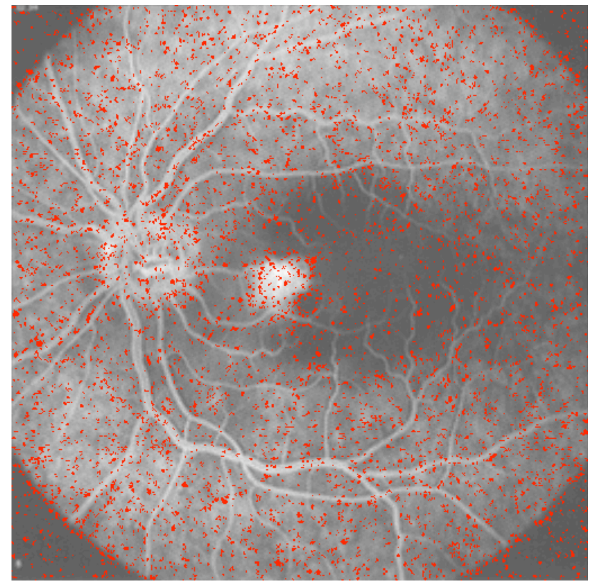
uncontrolled



controlled



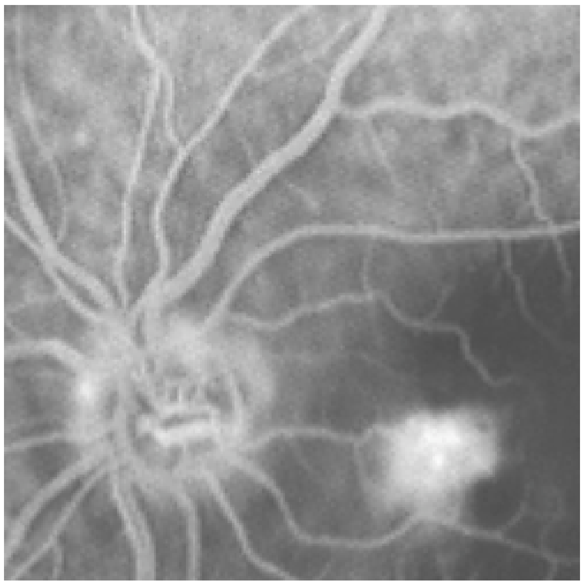
suppressed



new york university mrl

Results: Sharpening

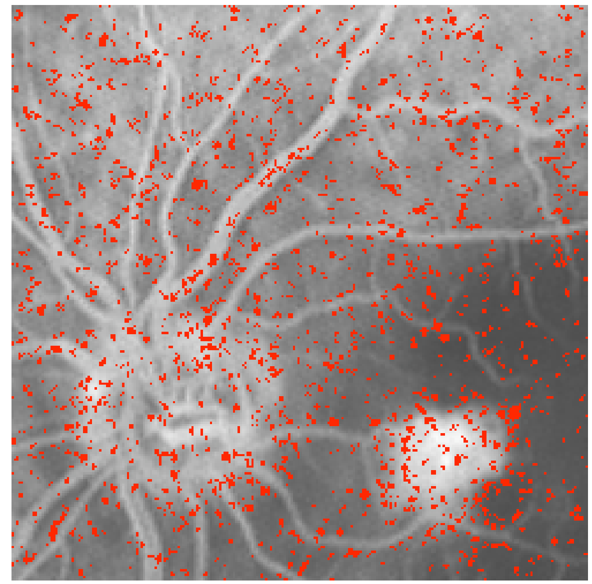
original



uncontrolled

controlled

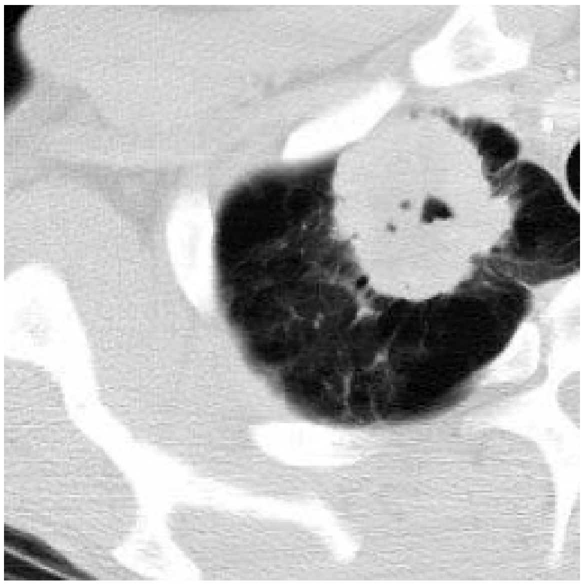
suppressed



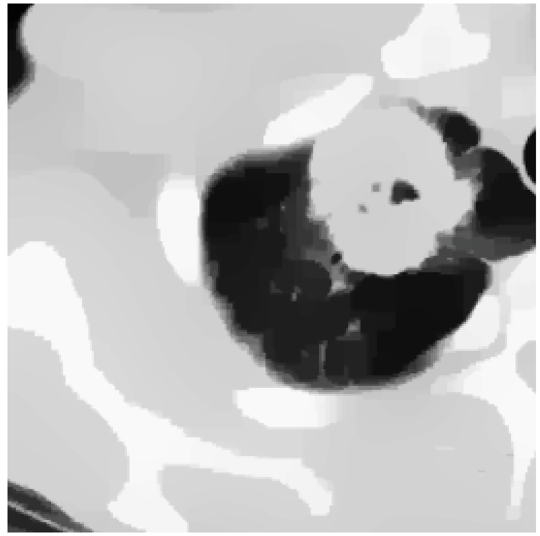
new york university mrl

Results: Anisotropic Smoothing

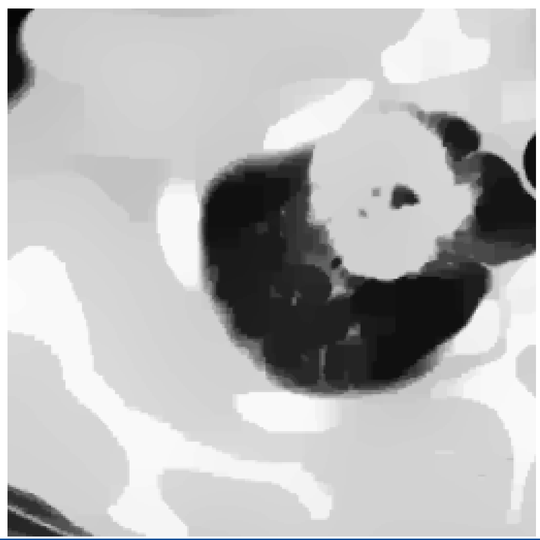
original



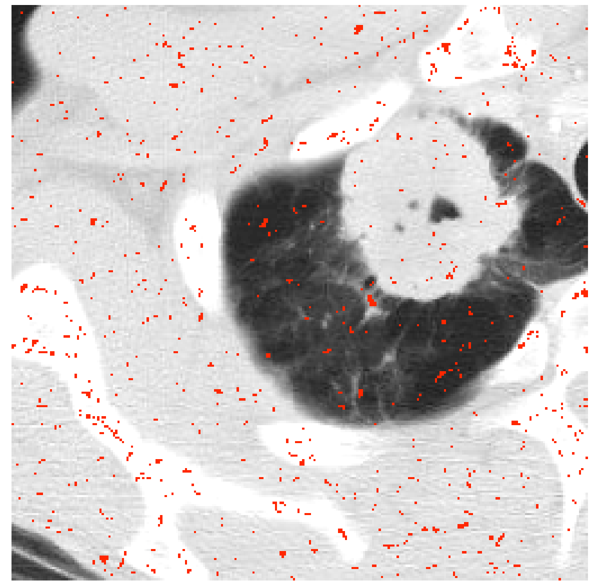
uncontrolled



controlled

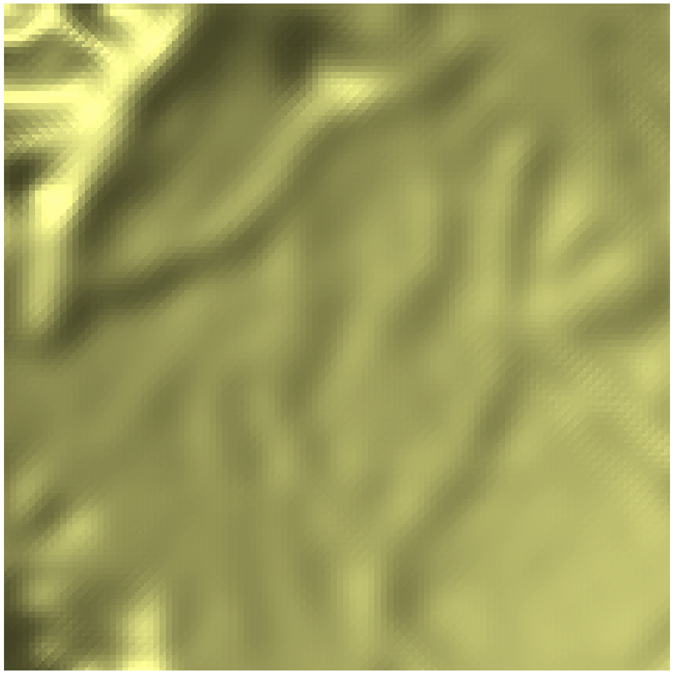


suppressed

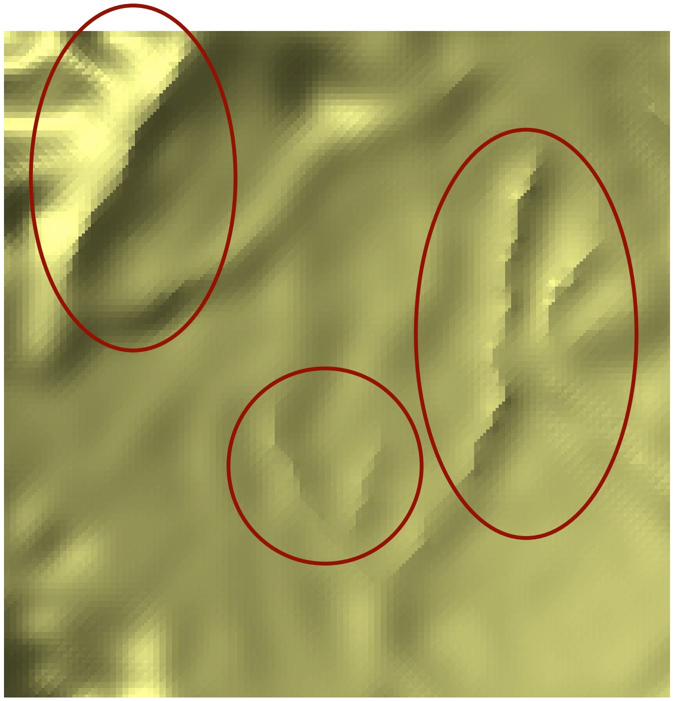


Local artifacts

Due to slowing time



uncontrolled



controlled

new york university mrl

No progress guarantee

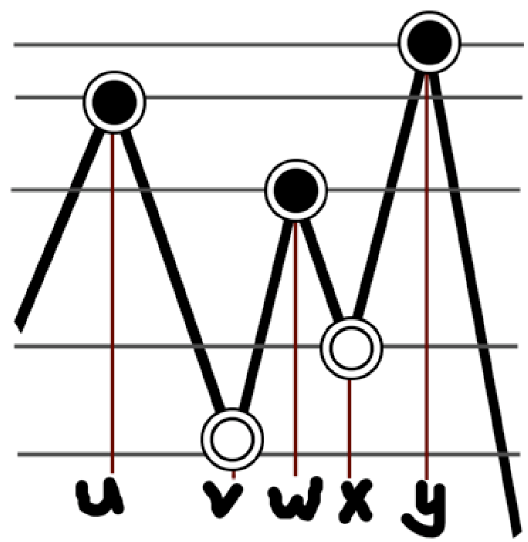
Terminate when proposed values same as current

Never saw topology control prevent progress everywhere

Results: Persistence

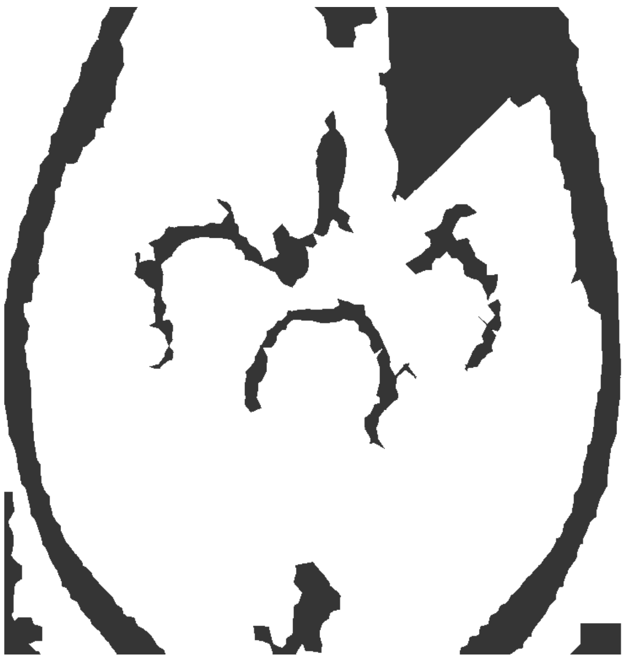
- Measures the importance of a feature
- Common measure is difference in value between an extremum and its paired saddle
- We can track the time it takes for an extremum (and its paired saddle) to be annihilated under smoothing

1D:



Results: Persistence

Difference
in value

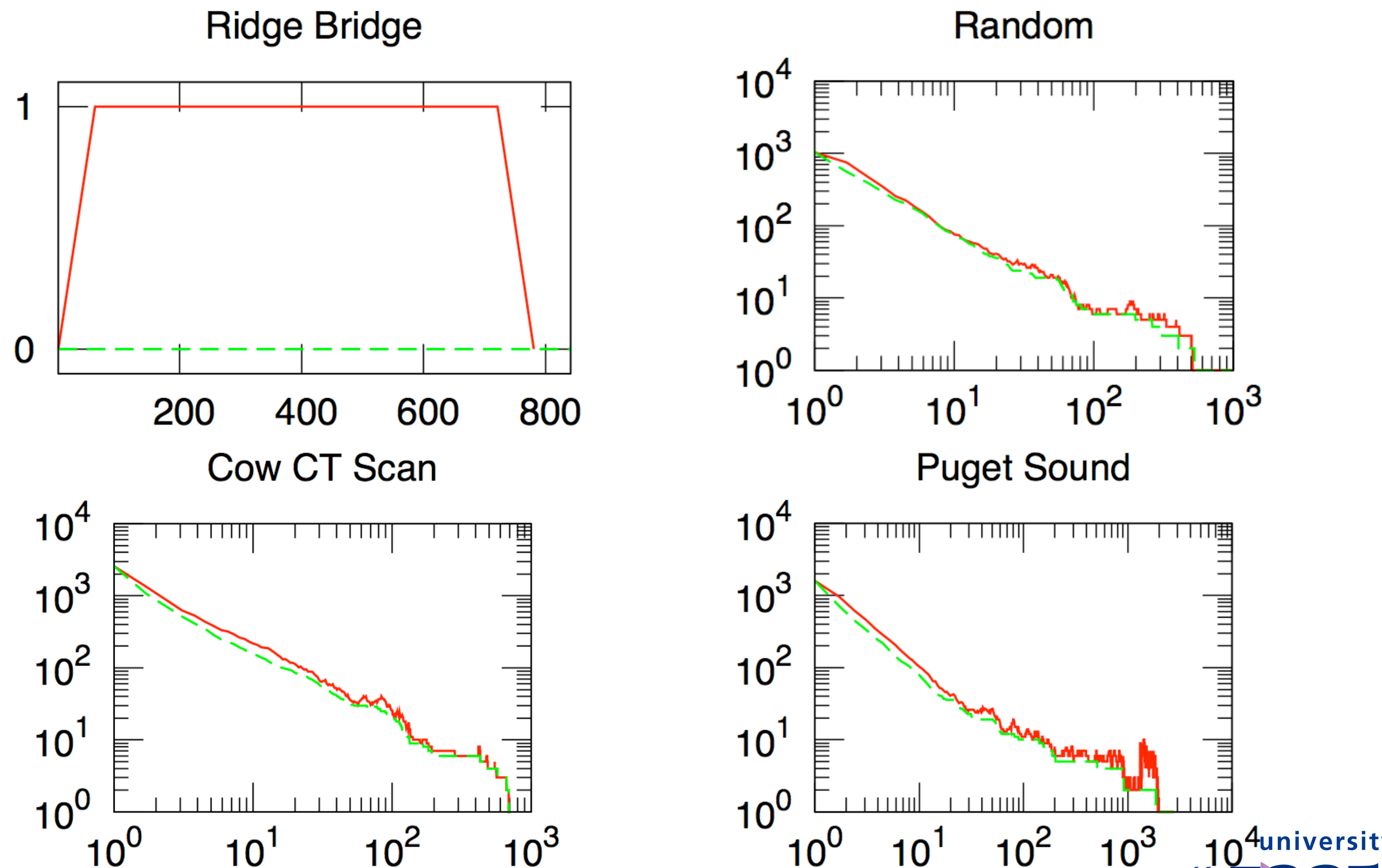


Cow CT Scan

Anisotropic
diffusion lifetime

Features shaded black

Critical Points Over Time



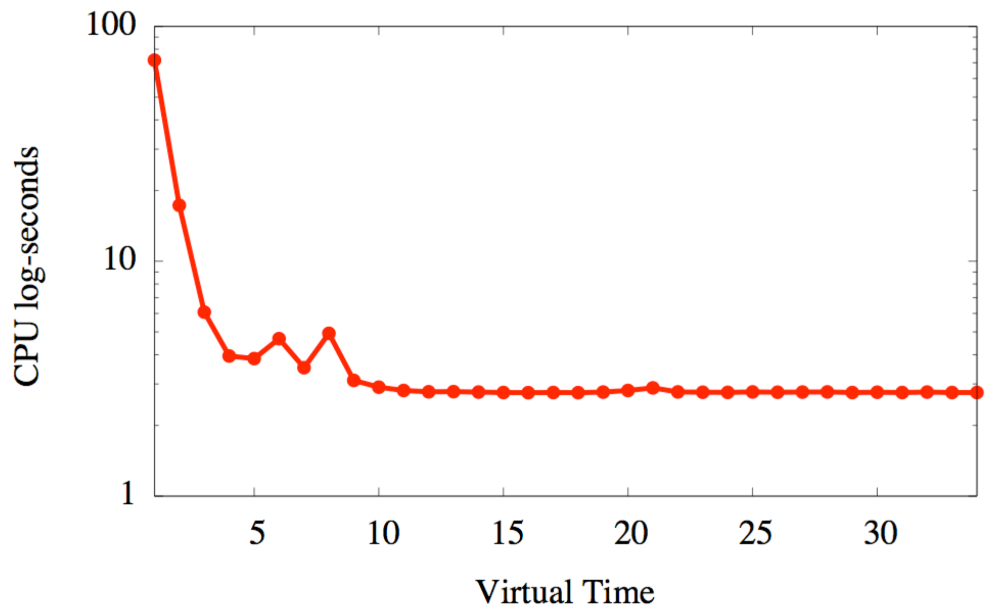
Performance

Performance related to number of edge flips and number of undesired topological events

Roughly corresponds to number of critical points

Performance

- Laplacian smoothing Puget Sound to one global maximum is 2.2x slower
- 44% of the time is spent in the first 3 steps



Future Work

Extend to 3D

Predict disallowed events to distribute undesired artifacts

Conclusion

A simple algorithm that controls topology changes when filtering

Contact: Yotam Gingold <gingold@mrl.nyu.edu>
Denis Zorin <dzorin@mrl.nyu.edu>

Thanks to: Chris Wu, Chee Yap, Adrian Secord,
NYU CS Colleagues, and the reviewers

new york university mrl

Fin

Filters

Discrete Laplacian smoothing (diffusion)

$$p_L^{l+1}(v) = p^l(v) + \Delta t \sum_{edges(v,w)} (p^l(w) - p^l(v))$$

Sharpening

$$p_S^{l+1}(v) = p^l(v) + \Delta t (p_L^n(v) - p^0(v))$$

Discrete Anisotropic smoothing ([Perona and Malik 1990])

$$p_{AD}^{l+1}(v) = p^l(v) + \Delta t \sum_{edges(v,w)} \frac{p^l(w) - p^l(v)}{1 + \|p^l(w) - p^l(v)\|^2 / k^2}$$

Feature

- Sweep a plane down. Notice how at a maximum a new contour is born! Notice how it merges with another contour at a saddle!
- Same with minimums and sweeping upwards!

Feature

Hills and valleys until it gets complicated -- max and mins (pics from original slide)

Feature

Sweep a plane down data set

- Maximum creates a contour
- Minimum creates a contour
- Saddle merges two contours
- Talk to denis to get terminology consistent

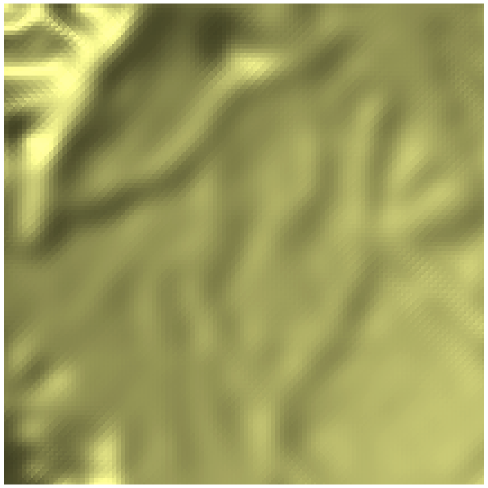
Feature

The set of contours from an extremum until the saddle which merges the contours with another extremum's contours, as we sweep a plane upwards/downwards

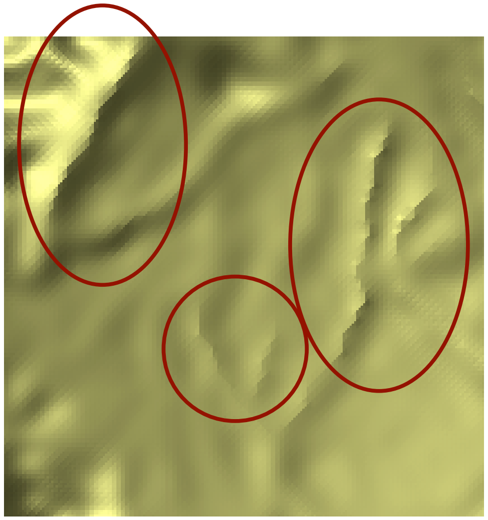
Single contour component from appearance at max/min until merger with another component at saddle as isovalue decreases/increases

No guaranteed progress

Terminate when proposed values same as current
Can topology control prevent progress everywhere?
Locally:

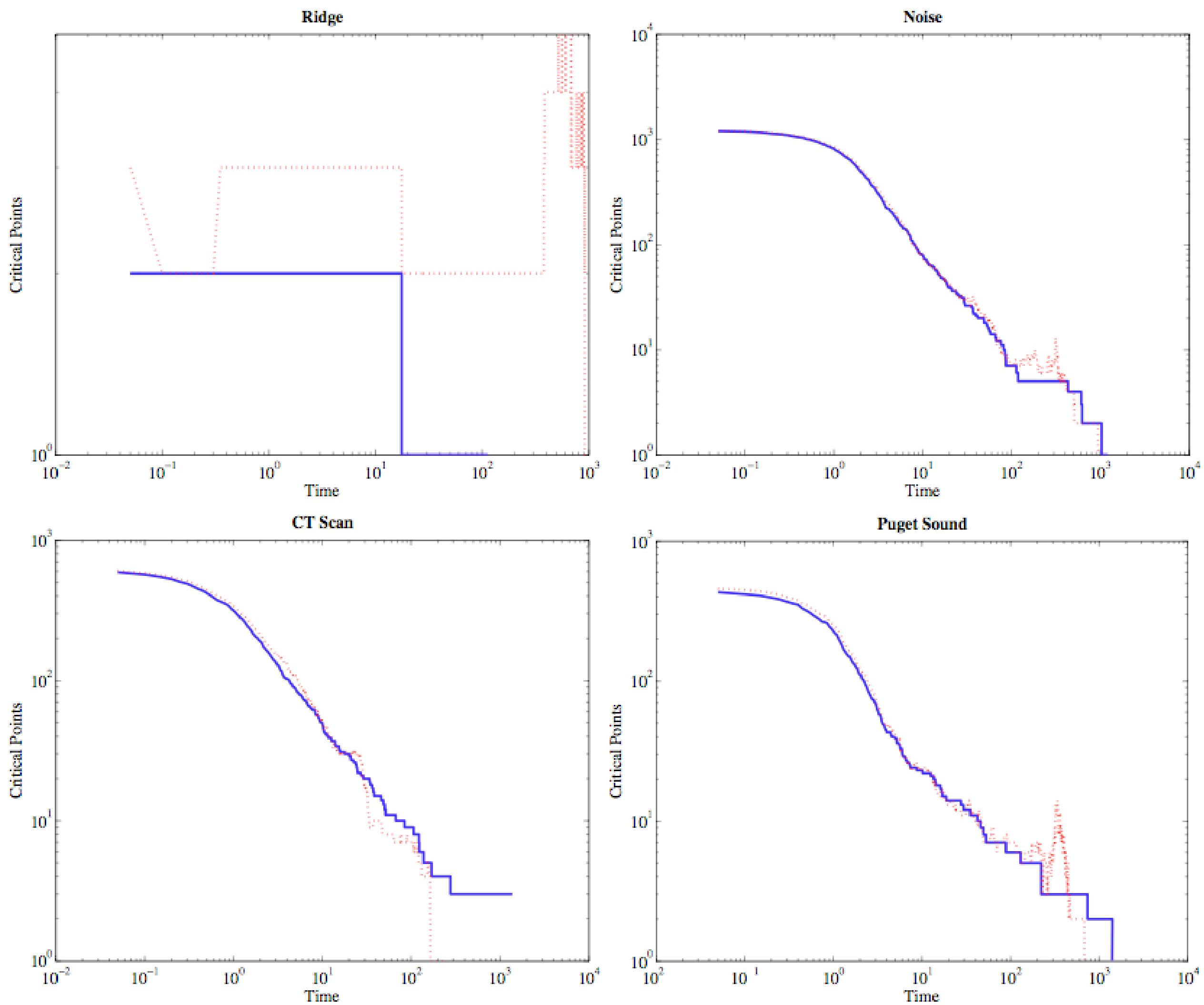


uncontrolled



controlled

Critical Points Over Time



new york university mrl