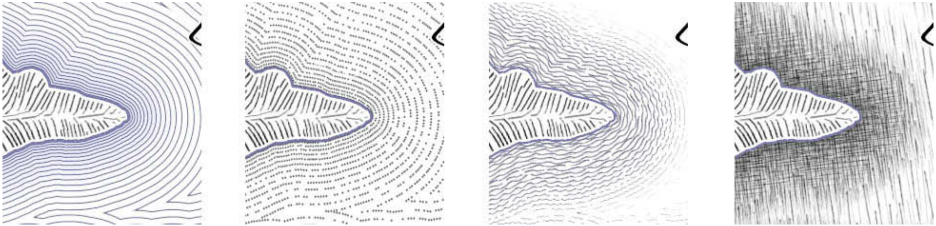
Real-Time Rendering of Water Surfaces with Cartography-Oriented Design

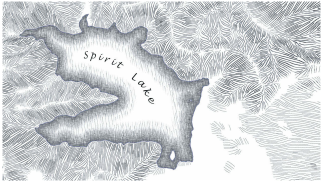
9th International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging

Amir Semmo [1] Jan Eric Kyprianidis [2] Matthias Trapp [1] Jürgen Döllner [1]



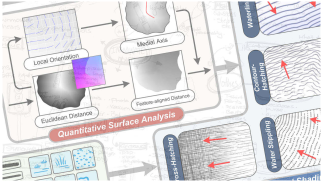
[1] Hasso-Plattner-Institut, Germany
[2] TU Berlin, Germany

Overview

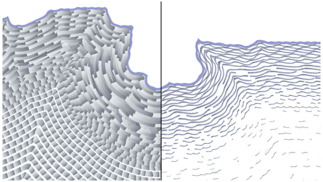


1) Motivation

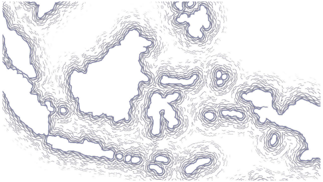
2) Design Principles



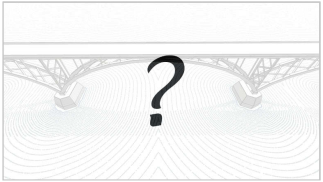
3) System Overview



4) Method

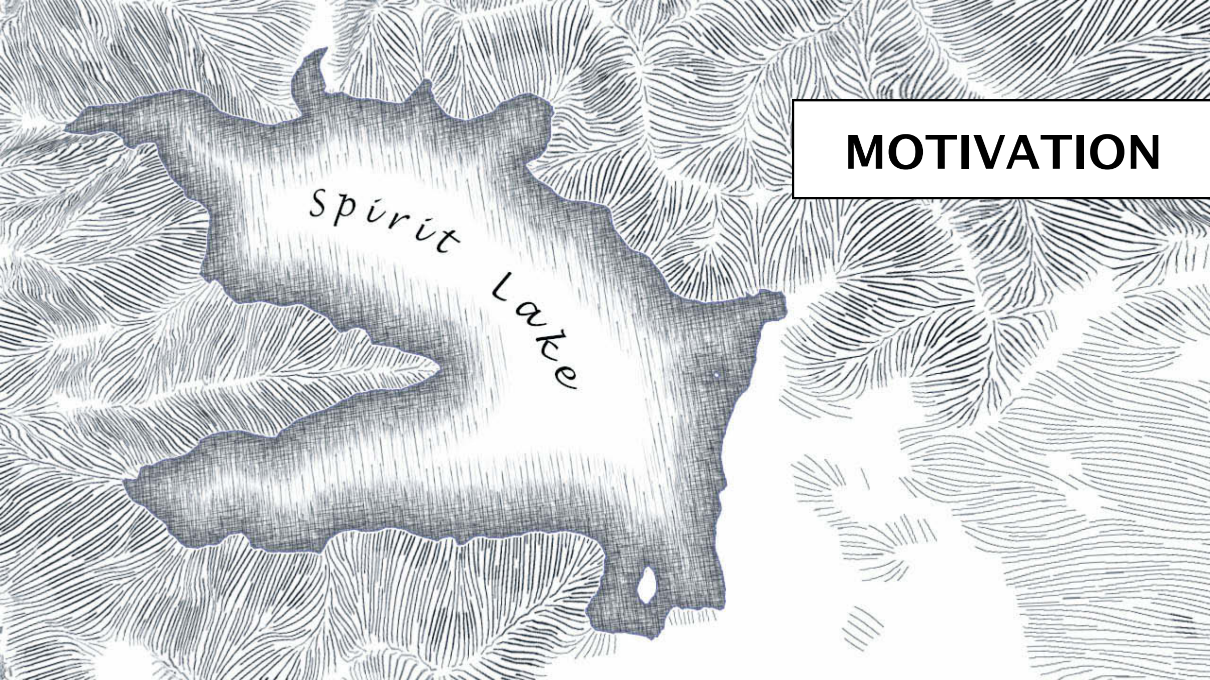


5) Results



6) Questions

MOTIVATION



State-of-the-Art Water Rendering

- computer-generated illustrations of water surfaces mainly based on photorealistic rendering
- but have neglected the challenges water surfaces exhibit for map design
- how to ease orientation, navigation, and analysis tasks within 3D geovirtual environments?
- challenges: emphasize land-water interface, consider figure-ground relations, express motion

CryEngine 3

Google Earth

Cartographers developed design principles that address these challenges

State-of-the-Art Water Rendering

- computer-generated illustrations of water surfaces mainly based on photorealistic rendering
- but have neglected the challenges water surfaces exhibit for map design
- how to ease orientation, navigation, and analysis tasks within 3D geovirtual environments?
- challenges: emphasize land-water interface, consider figure-ground relations, express motion

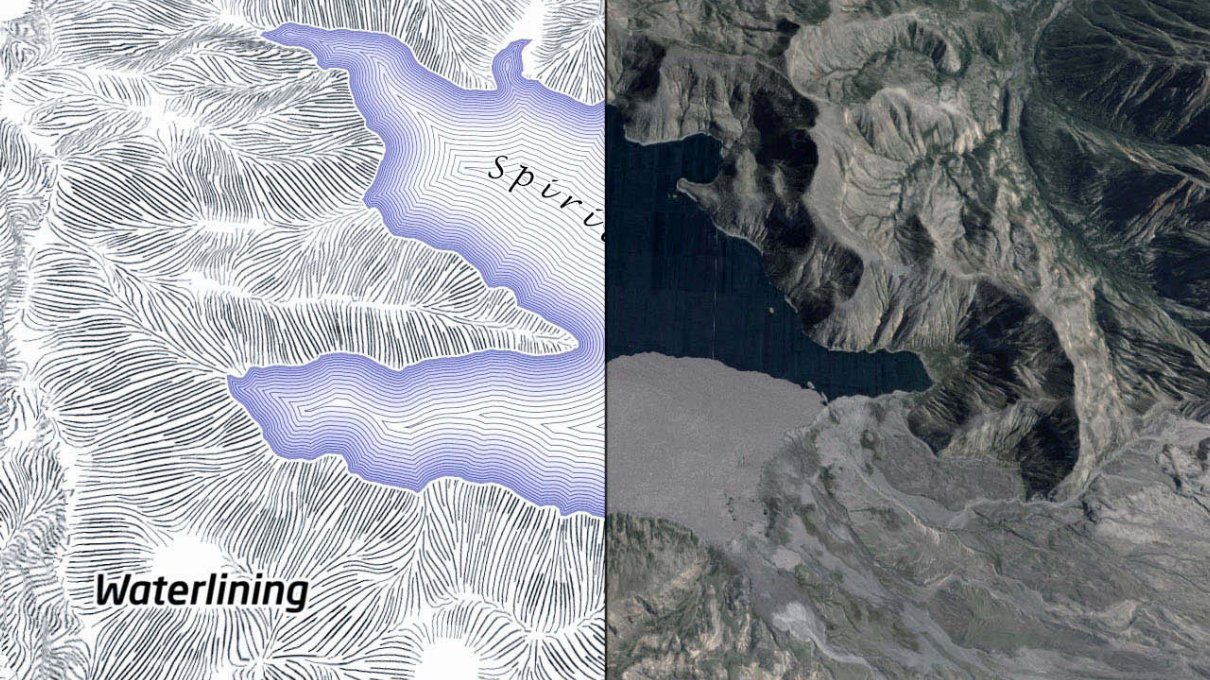
CryEngine 3

Google Earth

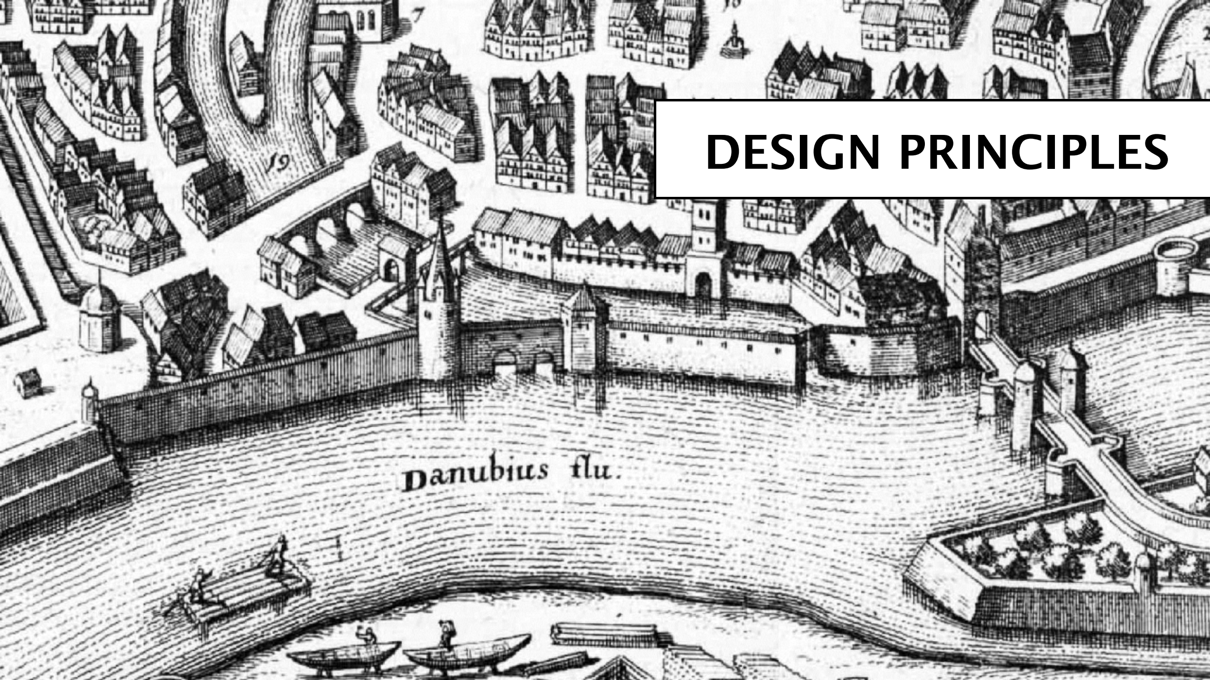
Cartographers developed design principles that address these challenges

Cartographic Design

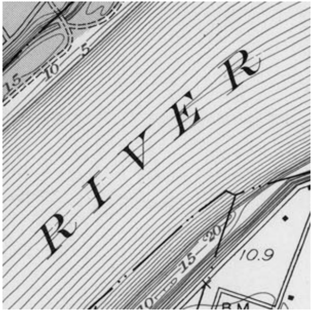
The visualization of geospatial features using a map-like representation that considers its purpose and the target audience.

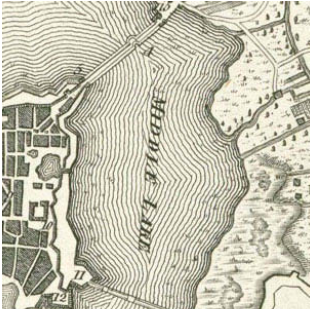
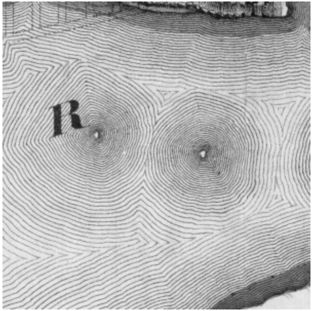

Spiri
Waterlining

DESIGN PRINCIPLES



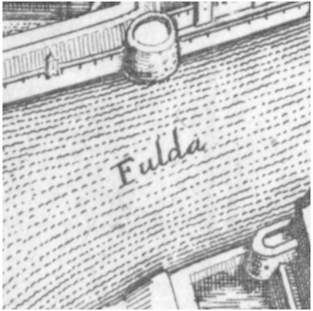
Design Principles – Waterlining



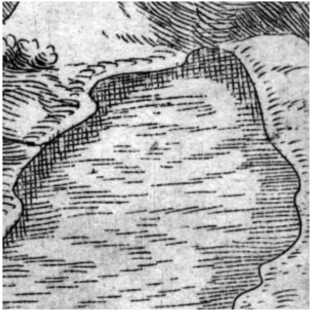
- became popular in the first half of the 20th century for lithographed maps
- fine solid lines are drawn parallel to shorelines with gradually increasing spaces
- waterlining provides dynamism and communicates distance information

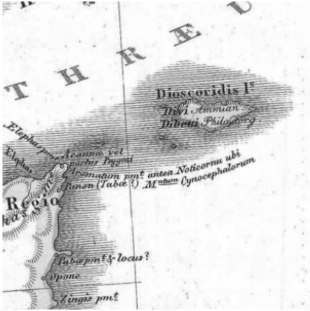
Design Principles – Water Stippling



- small dots aligned to shorelines with non-linear distance
- contrary to traditional artwork, stipples have a varying density and irregularly overlap
- varying density to depict flow velocity or at occluded areas to enhance depth cues

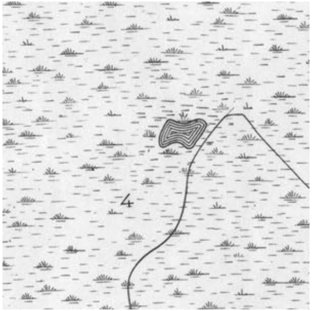
Design Principles – Hatching & Vignetting

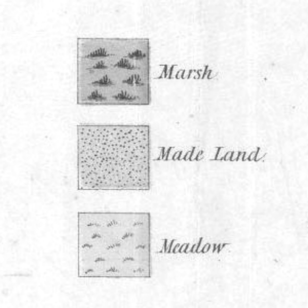




- individual strokes placed with high density near shorelines complemented by loose lines
- drawn excessively wavy with increasing irregularity towards middle stream to express motion
- non-feature-aligned cross-hatches for land-water-distinction, coastal vignettes

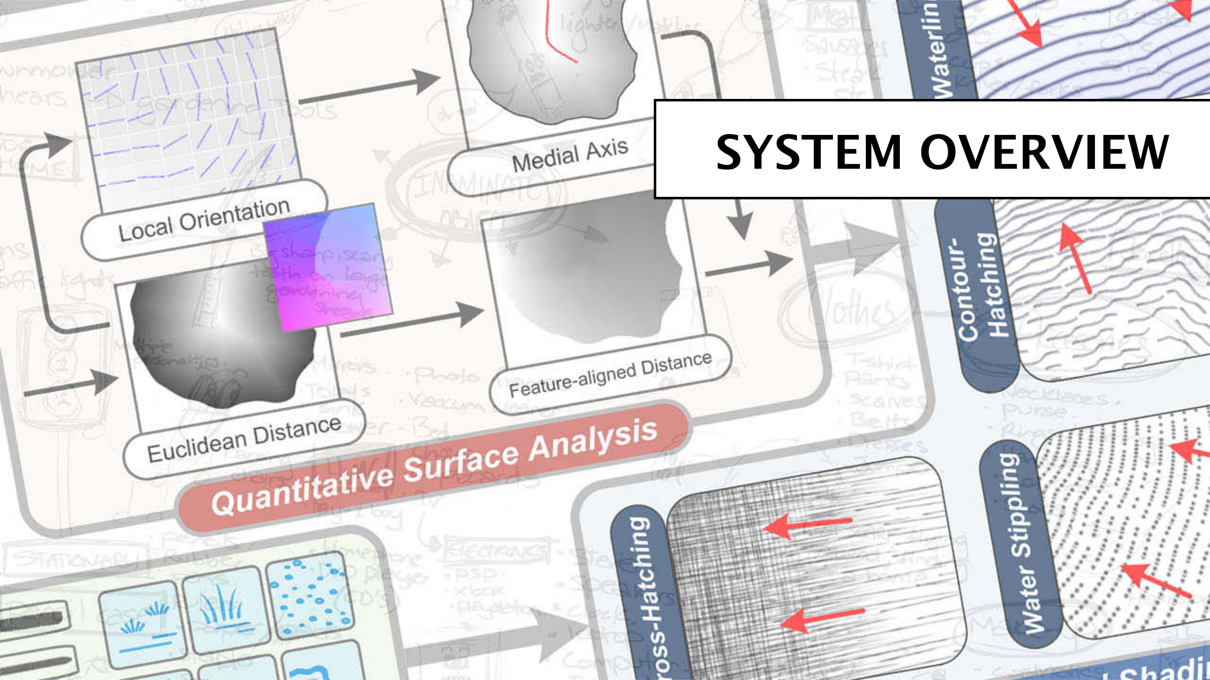
Design Principles – Thematic Visualization: Annotation / Symbolization

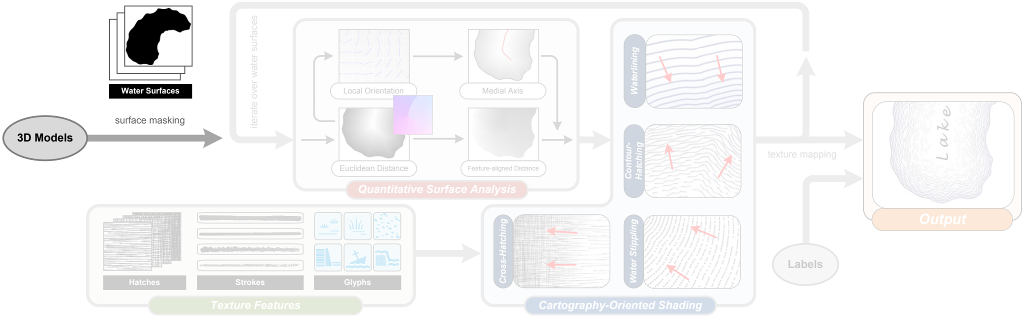




- names depicted with italic (slanted) letters, following principal curvature directions
- irregular placement of signatures with area-wide coverage to communicate water features
- placement of glyphs (e.g., arrows) along streamlines to symbolize flow direction of rivers

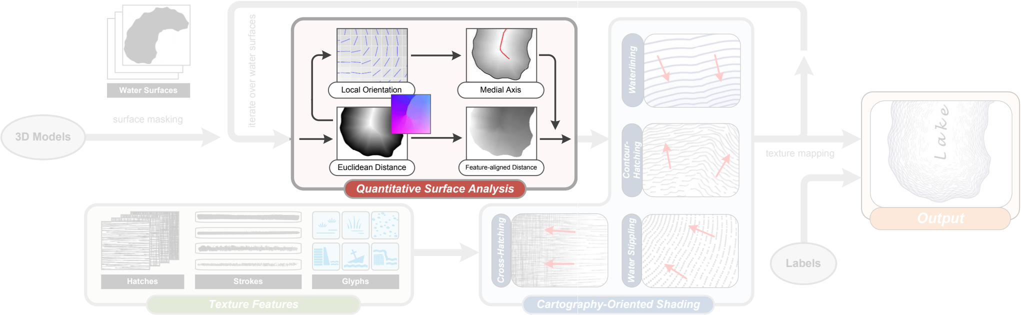
SYSTEM OVERVIEW





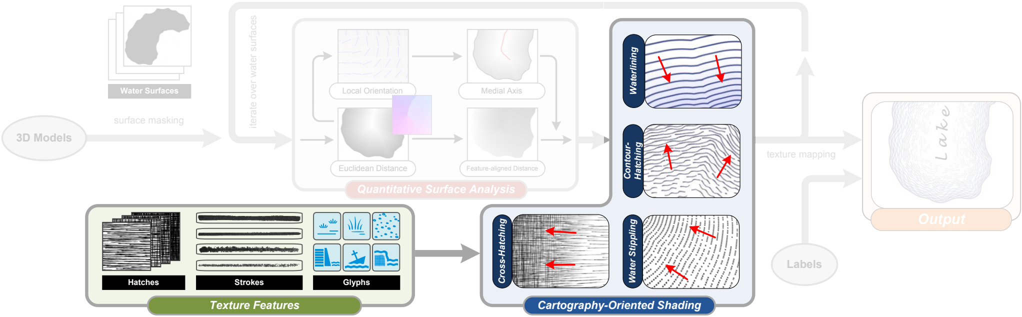
Model Input & Masking

System Overview



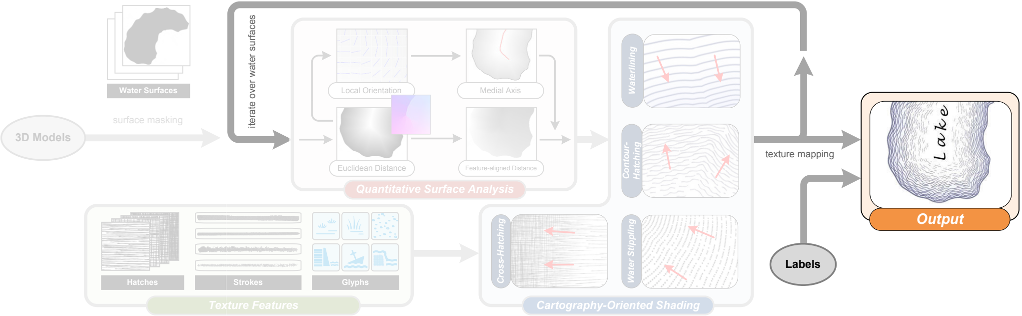
Quantitative Surface Analysis

System Overview

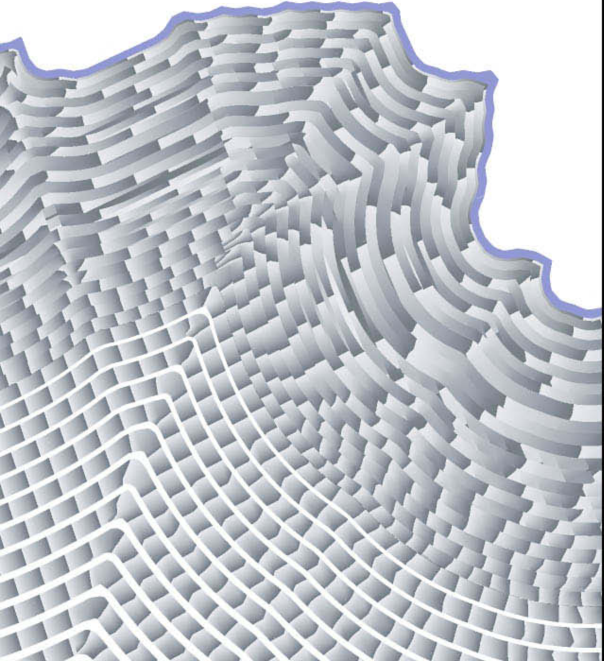


Cartography-Oriented Shading & Texture Features

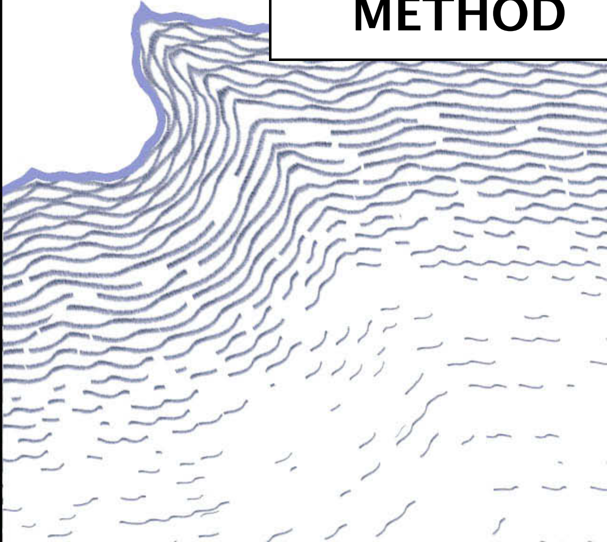
System Overview



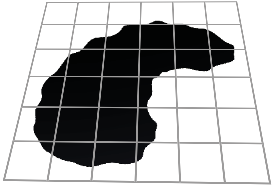
Process Iteration & Texture Mapping



METHOD



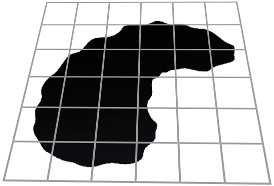
- input: 2D or 3D water surfaces defined as polygons or triangular irregular networks



Input

Method – Data Input & Euclidean Distance Transform

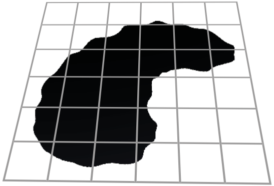
- input: 2D or 3D water surfaces defined as polygons or triangular irregular networks
- using orthographic projections, the models' shape are captured in 2D binary masks



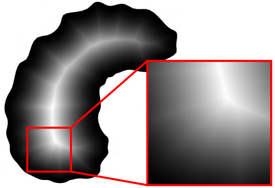
Input

Method – Data Input & Euclidean Distance Transform

- input: 2D or 3D water surfaces defined as polygons or triangular irregular networks
- using orthographic projections, the models' shape are captured in 2D binary masks
- distance transform to obtain minimum Euclidean distance of each pixel to a shoreline



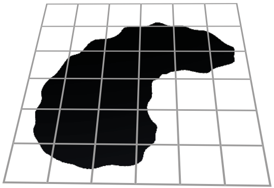
Input



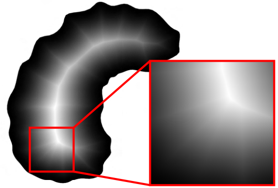
Shoreline Distance

Method – Data Input & Euclidean Distance Transform

- input: 2D or 3D water surfaces defined as polygons or triangular irregular networks
- using orthographic projections, the models' shape are captured in 2D binary masks
- distance transform to obtain minimum Euclidean distance of each pixel to a shoreline
- "Parallel Banding Algorithm" [Cao et al., I3D 2010] obtains *distance map* in real-time [CUDA]



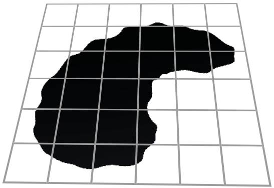
Input



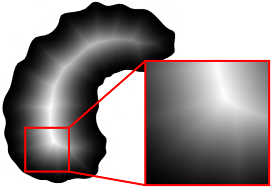
Shoreline Distance

Method – Data Input & Euclidean Distance Transform

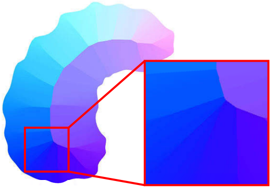
- input: 2D or 3D water surfaces defined as polygons or triangular irregular networks
- using orthographic projections, the models' shape are captured in 2D binary masks
- distance transform to obtain minimum Euclidean distance of each pixel to a shoreline
- "Parallel Banding Algorithm" [Cao et al., I3D 2010] obtains *distance map* in real-time [CUDA]



Input



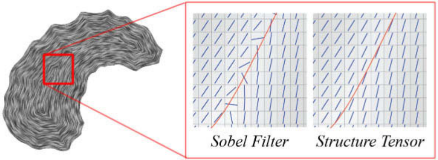
Shoreline Distance



Shoreline Direction

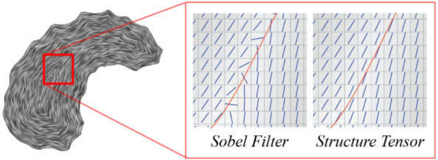
Method – Local Orientation Estimation & Medial Axes Computation

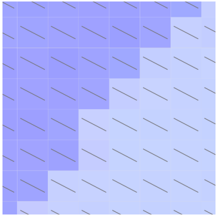
- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream



Method – Local Orientation Estimation & Medial Axes Computation

- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream
- compare and threshold the unsigned gradient orientation for each point

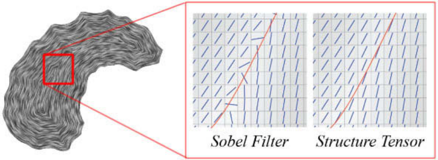


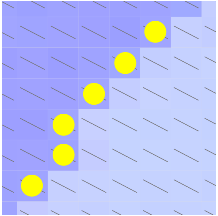


Thresholding
Shoreline Directions

Method – Local Orientation Estimation & Medial Axes Computation

- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream
- compare and threshold the unsigned gradient orientation for each point

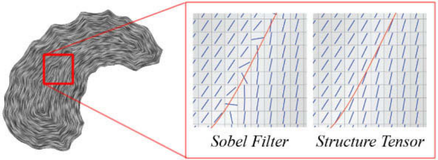


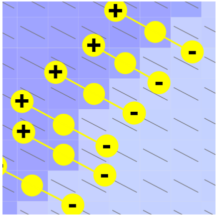


Thresholding
Shoreline Directions

Method – Local Orientation Estimation & Medial Axes Computation

- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream
- compare and threshold the unsigned gradient orientation for each point

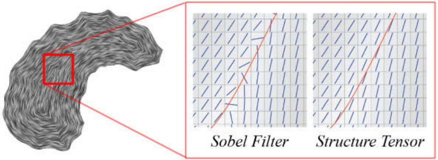




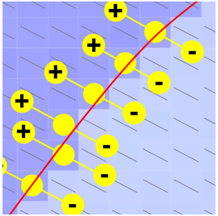
Thresholding
Shoreline Directions

Method – Local Orientation Estimation & Medial Axes Computation

- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream



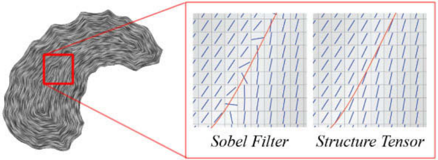
- compare and threshold the unsigned gradient orientation for each point

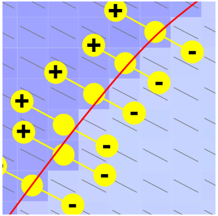


Thresholding
Shoreline Directions

Method – Local Orientation Estimation & Medial Axes Computation

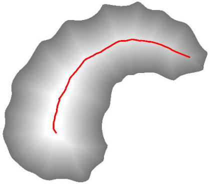
- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream
- compare and threshold the unsigned gradient orientation for each point





Thresholding
Shoreline Directions

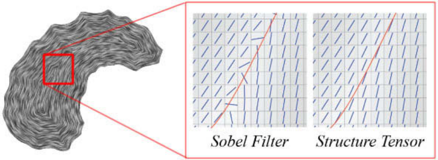
→

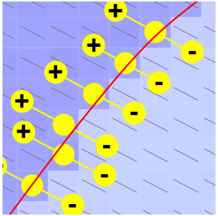


Medial Axis Result

Method – Local Orientation Estimation & Medial Axes Computation

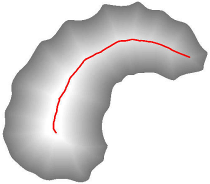
- smoothed structure tensor with eigenanalysis to obtain stable estimates of local orientation
  [Brox et al., 2006]
- used to derive medial axes for aligning design elements (e.g., labels) along middle stream
- compare and threshold the unsigned gradient orientation for each point



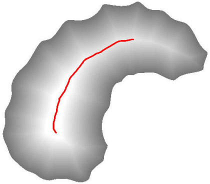


Thresholding Shoreline Directions

→



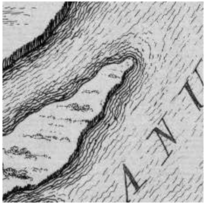
Medial Axis Result

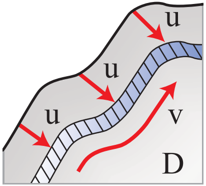


+ Thresholded Shoreline Distance

Contour-Hatching

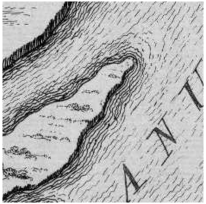
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)

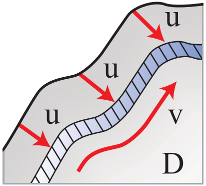




Contour-Hatching

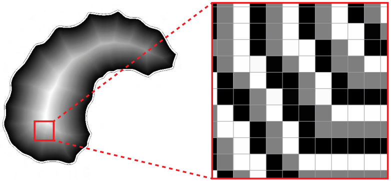
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)





Our Approach

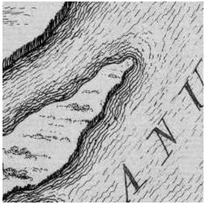
- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]

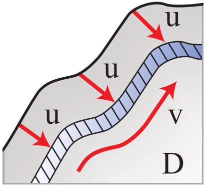


Method – Feature-aligned Distance Transform

Contour-Hatching

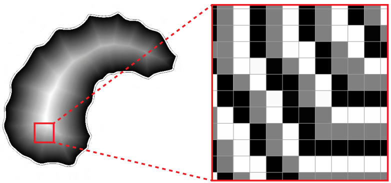
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)

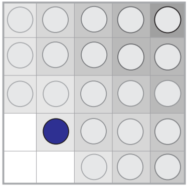




Our Approach

- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]

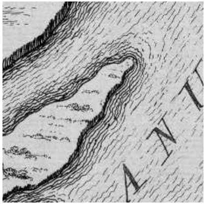


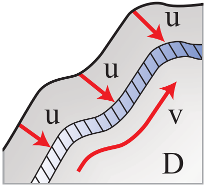


Method – Feature-aligned Distance Transform

Contour-Hatching

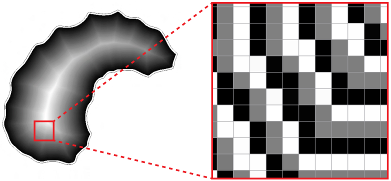
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)

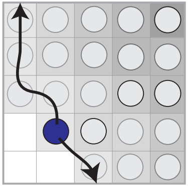




Our Approach

- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]

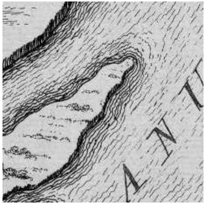


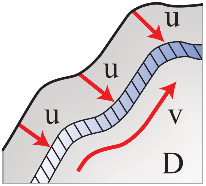


Method – Feature-aligned Distance Transform

Contour-Hatching

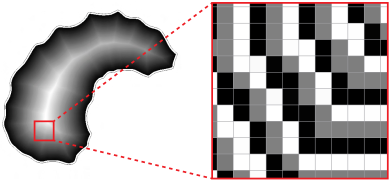
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)

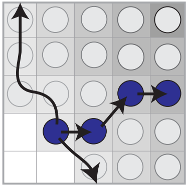




Our Approach

- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]

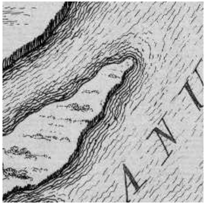


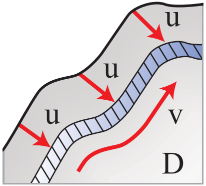


Method – Feature-aligned Distance Transform

Contour-Hatching

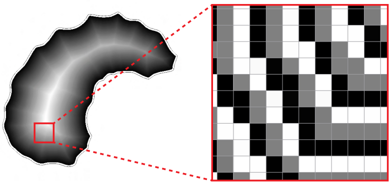
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)

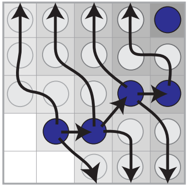




Our Approach

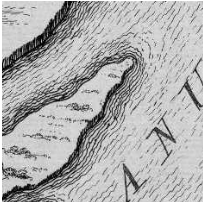
- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]

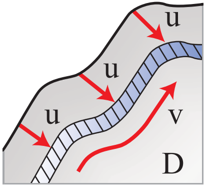




Contour-Hatching

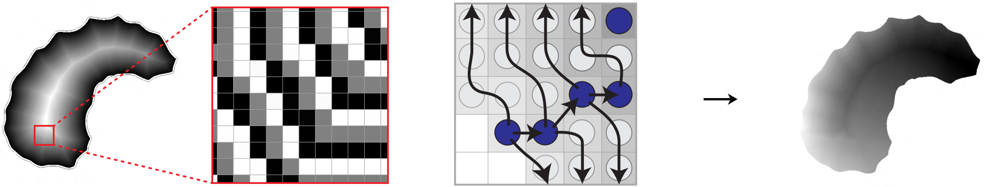
- requires design elements (e.g., strokes) to be exactly aligned with the shorelines
- requires fine control over their placement per rendering pass (water movements!)



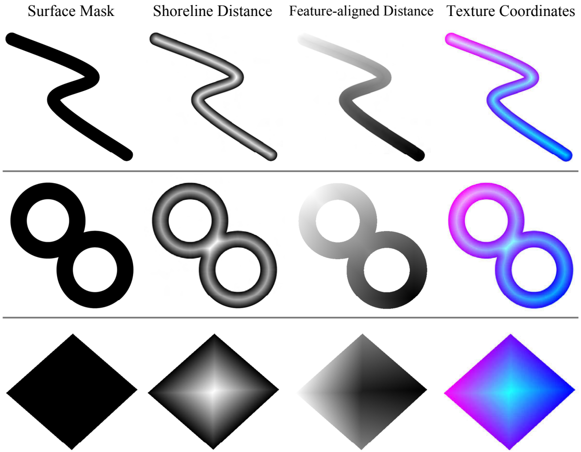


Our Approach

- parameterize level-set curves of distance map to obtain distances along its tangential field
- level sets correspond to integral part of shoreline distances for non-normalized distance map
- similar to vector propagation, distances are iteratively propaged in parallel [CUDA]



Method – Feature-aligned Distance Transform



- **Texture Coordinates**

  are directly derived by combining shoreline and feature-aligned distances

- **Texture-based Shading**

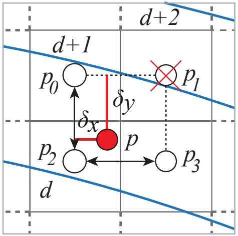
  that is capable of feature-aligned hatching at real-time frame rates

- **Artistic Control**

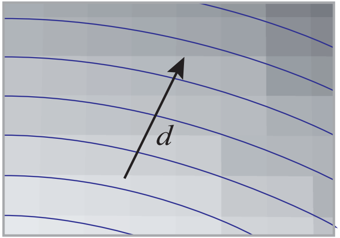
  with dynamic parameterization at run-time to express water movements

Method – Feature-aligned Distance Transform

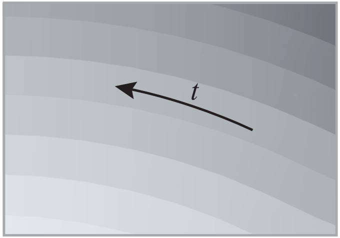
- contrary to energy minimization, our approach does not provide continuity across level sets
- individual texture features are aligned with level-set curves → no such constraint is required
- piecewise-linear approximation of shoreline / feature-aligned distances by bilinear sampling
- choose distance map resolution that balances rendering quality and performance



Bilinear Sampling



Discrete Input + Distance Lines



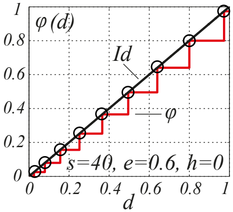
Continuous Output

Method – Waterlining

- waterlines correspond to shaded areas with equal shoreline distance
- non-linear step function to compute target distance values with non-equidistant interspaces
- waterlines are padded by fade-in and fade-out intervals for antialiasing and parameterized by the view distance for a continuous level-of-abstraction

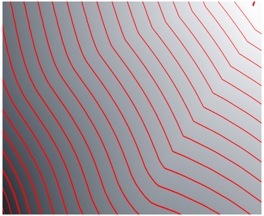
Shoreline Distance

+



Non-Linear Step

→



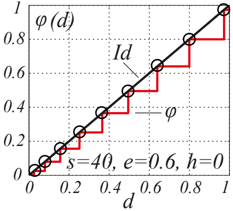
Waterlines

Method – Waterlining

- waterlines correspond to shaded areas with equal shoreline distance
- non-linear step function to compute target distance values with non-equidistant interspaces
- waterlines are padded by fade-in and fade-out intervals for antialiasing and parameterized by the view distance for a continuous level-of-abstraction

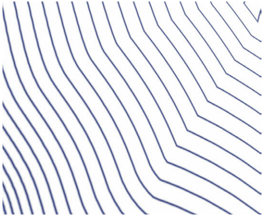
Shoreline Distance

+

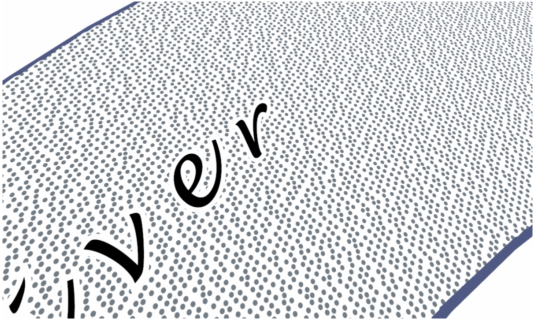


Non-Linear Step

→



Waterlines



Texture Bombing

- ... from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

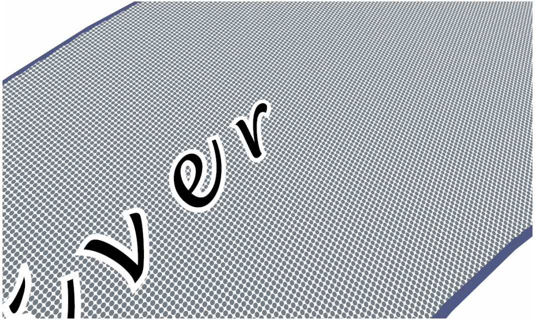
Stipples aligned with shorelines

- ... by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- ... using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]

Method – Water Stippling



Texture Bombing

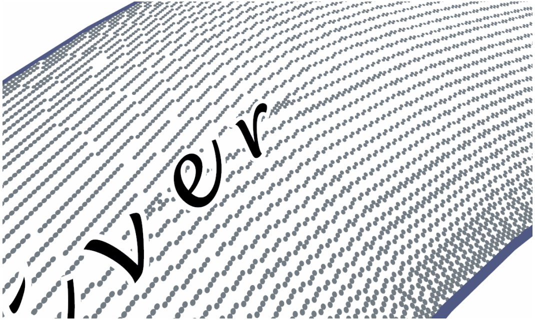
- ... from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

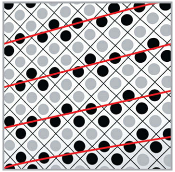
Stipples aligned with shorelines

- ... by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- ... using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]





Selection

Texture Bombing

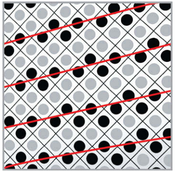
- … from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

Stipples aligned with shorelines

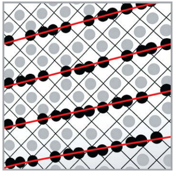
- … by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- … using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]



Selection



Displacement

Texture Bombing

- ... from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

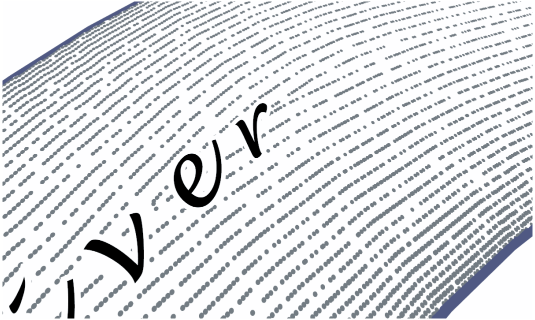
Stipples aligned with shorelines

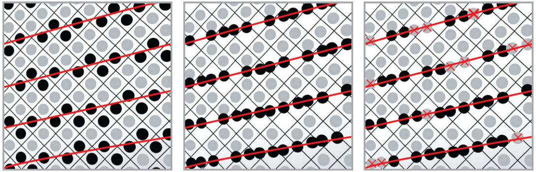
- ... by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- ... using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]

Method – Water Stippling





Selection Displacement Filtering

Texture Bombing

- ... from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

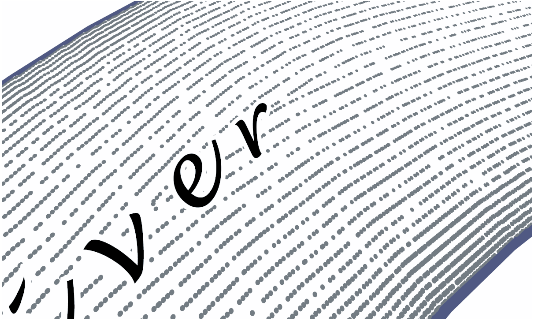
Stipples aligned with shorelines

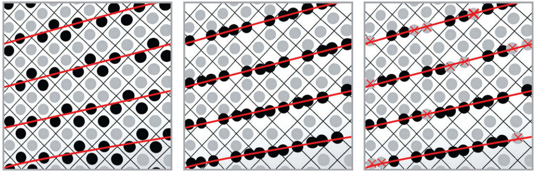
- ... by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- ... using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]

Method – Water Stippling





Selection Displacement Filtering

Texture Bombing

- ... from Glanville [2004] enhanced to place water stipples with feature-aligned distribution and irregular density

Stipples aligned with shorelines

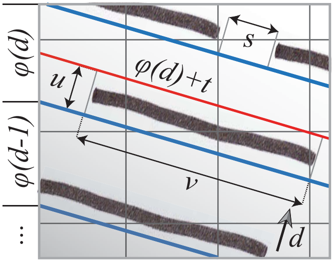
- ... by stipple selection, stipple displacement, and stipple filtering

Texture-based parameterization

- ... using tonal art maps to locally vary stipple density and tone, layered / iterative approach to regularize / vary stipple density [GLSL]

Method – Contour-Hatching

- parameterize shoreline / feature-aligned distances to irregularly place stroke maps [GLSL]
- control over stroke length, thickness, spacing, randomness according to shoreline distance
- enables time-varying parameterization for water movements and frame-to-frame coherence



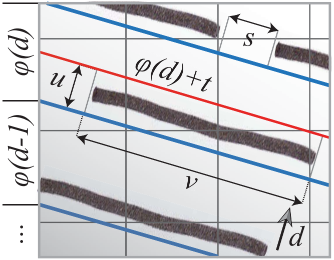
Stroke Placement

Texture Coordinates

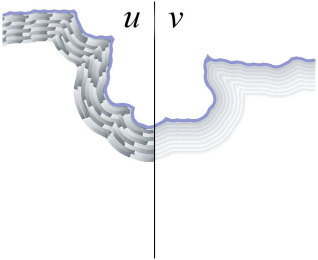
Rendering Result

Method – Contour-Hatching

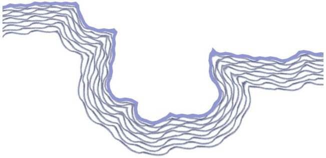
- parameterize shoreline / feature-aligned distances to irregularly place stroke maps [GLSL]
- control over stroke length, thickness, spacing, randomness according to shoreline distance
- enables time-varying parameterization for water movements and frame-to-frame coherence



Stroke Placement



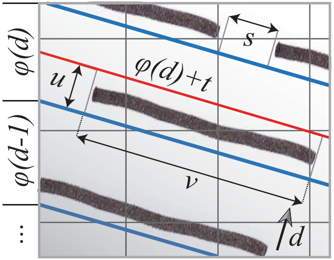
Texture Coordinates



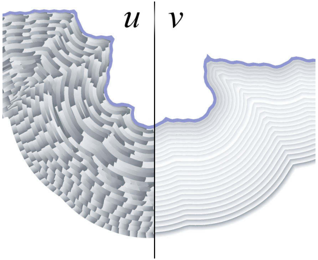
Rendering Result

Method – Contour-Hatching

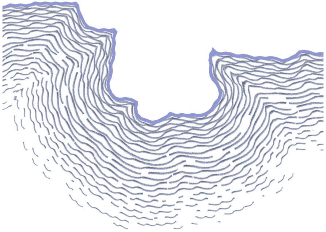
- parameterize shoreline / feature-aligned distances to irregularly place stroke maps [GLSL]
- control over stroke length, thickness, spacing, randomness according to shoreline distance
- enables time-varying parameterization for water movements and frame-to-frame coherence



Stroke Placement

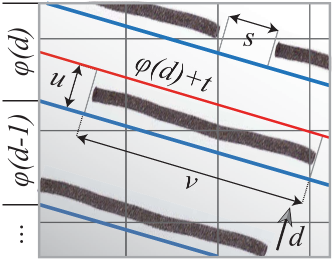


Texture Coordinates

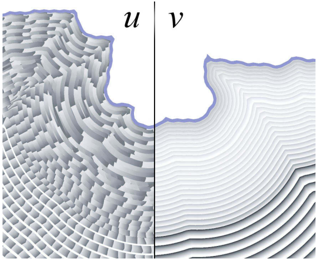

Rendering Result

Method – Contour-Hatching

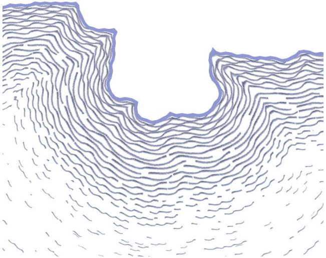
- parameterize shoreline / feature-aligned distances to irregularly place stroke maps [GLSL]
- control over stroke length, thickness, spacing, randomness according to shoreline distance
- enables time-varying parameterization for water movements and frame-to-frame coherence



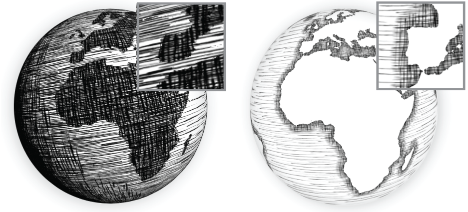
Stroke Placement



Texture Coordinates


Rendering Result

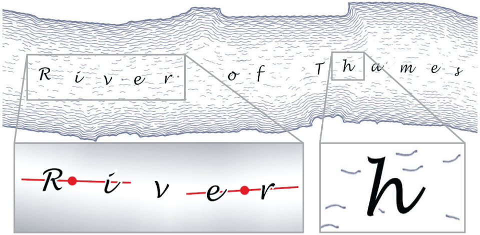
Method – Cross-Hatching & Labeling



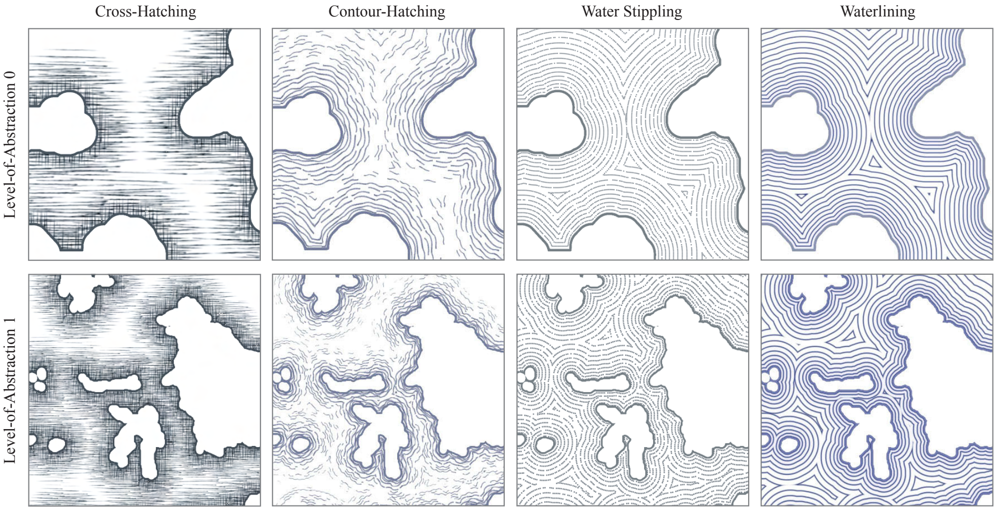
[Webb et al., 2002]   Our approach

- cross-hatching at shorelines by a tonal art map with varying stroke sizes / densities [Praun et al., 2001]
- our approach does not affect shading of landmass, i.e., terrain can be stylized separately

- downsample medial axes and use tangent information of structure tensor for arc-length parameterization of piecewise cubic Bezier curves
- OpenGL extension `NV_path_rendering` to render / transform instance-based text in a single pass

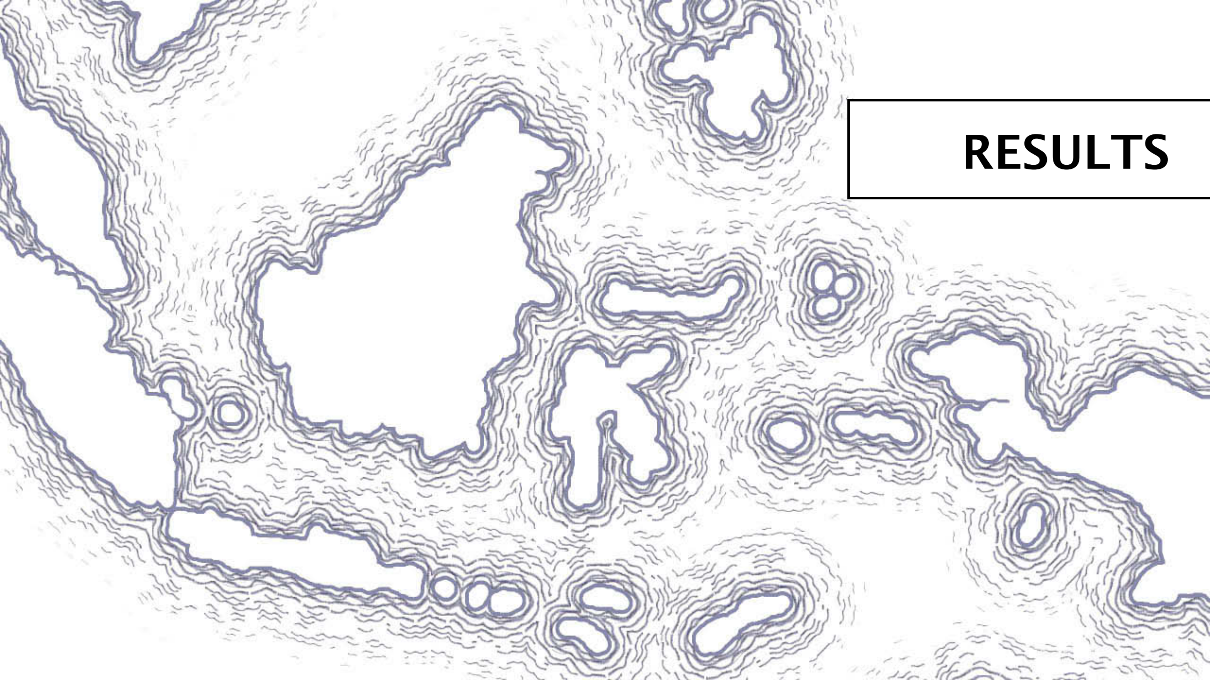


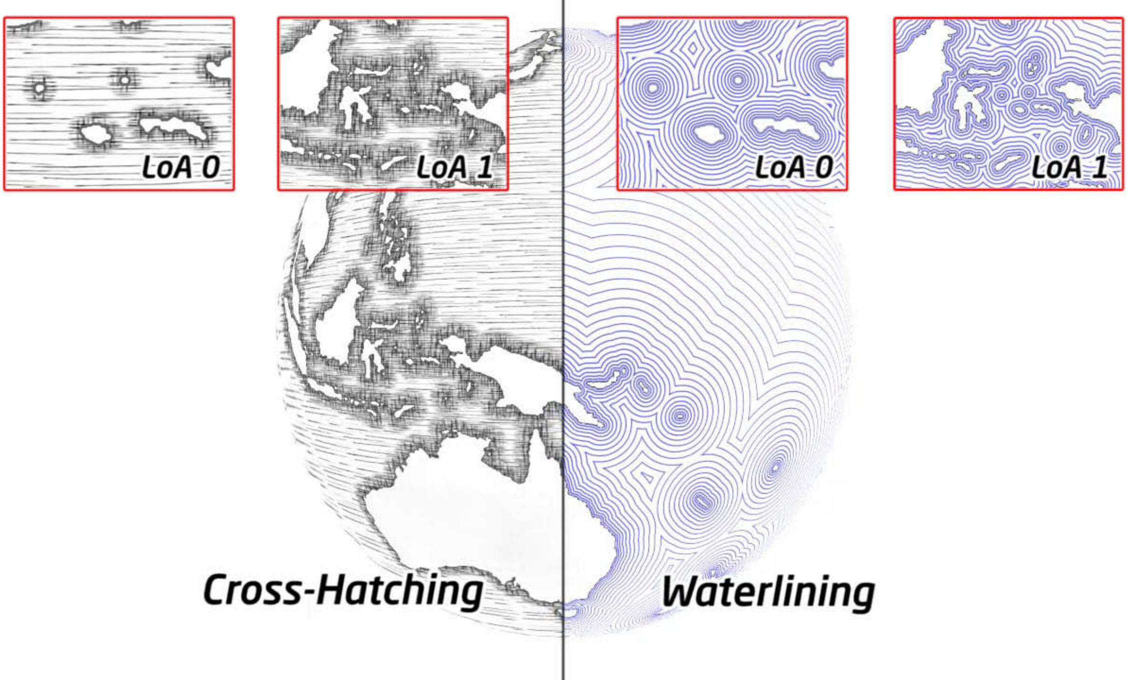
Method – View-Dependent Level-of-Abstraction



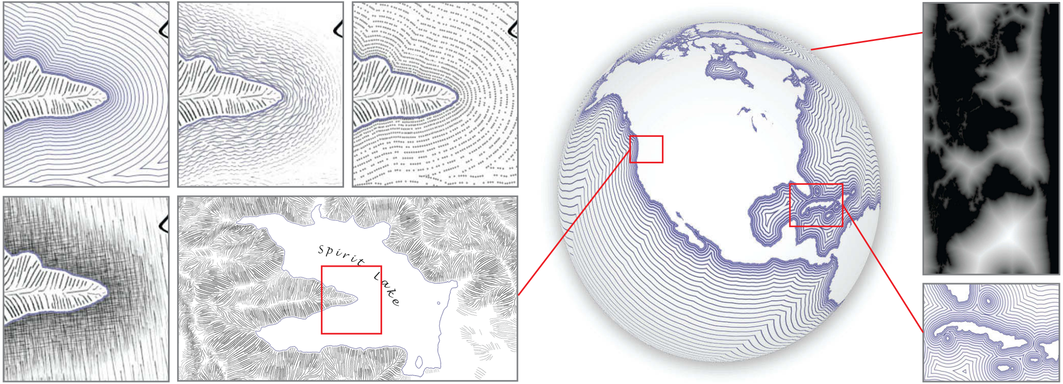
Semmo et al.

RESULTS

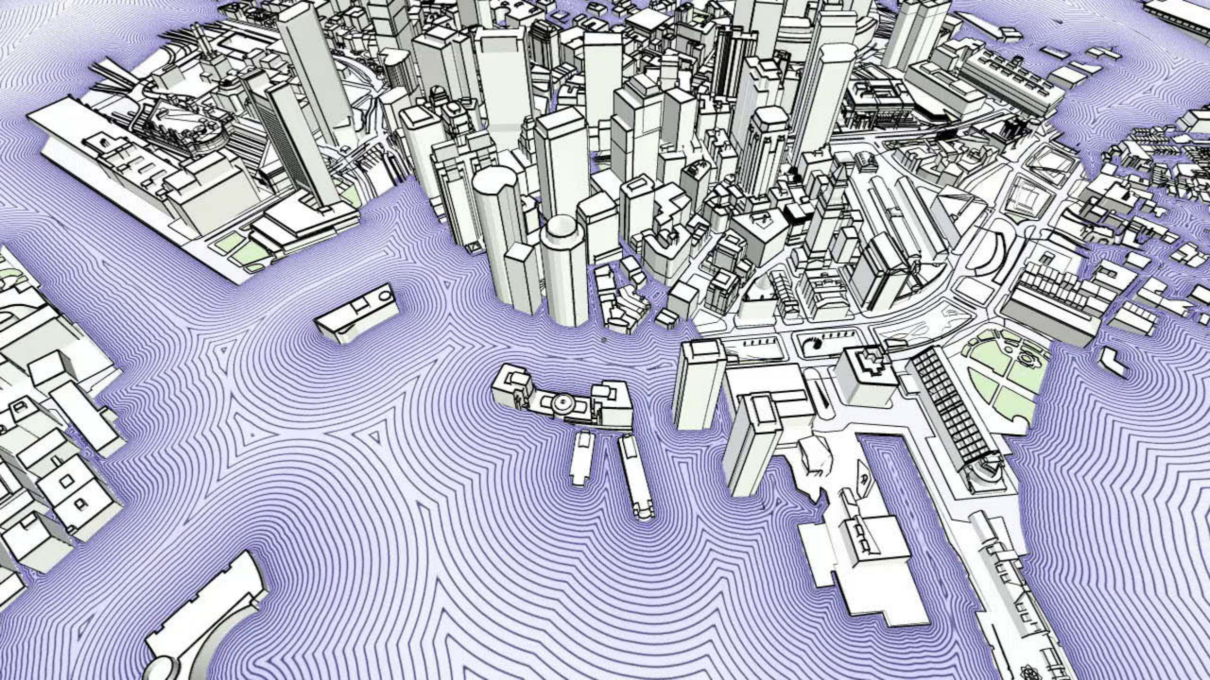



LoA 0
LoA 1
LoA 0
LoA 1
Cross-Hatching
Waterlining

Results – 3D Mapping

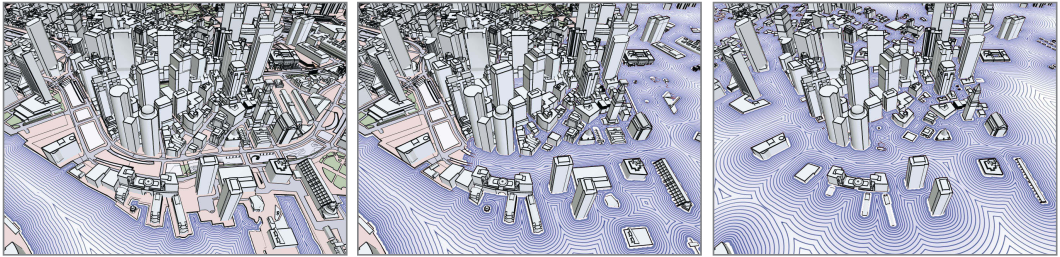


3D mapping within the environment of Mount St. Helens / waterlining applied to a globe

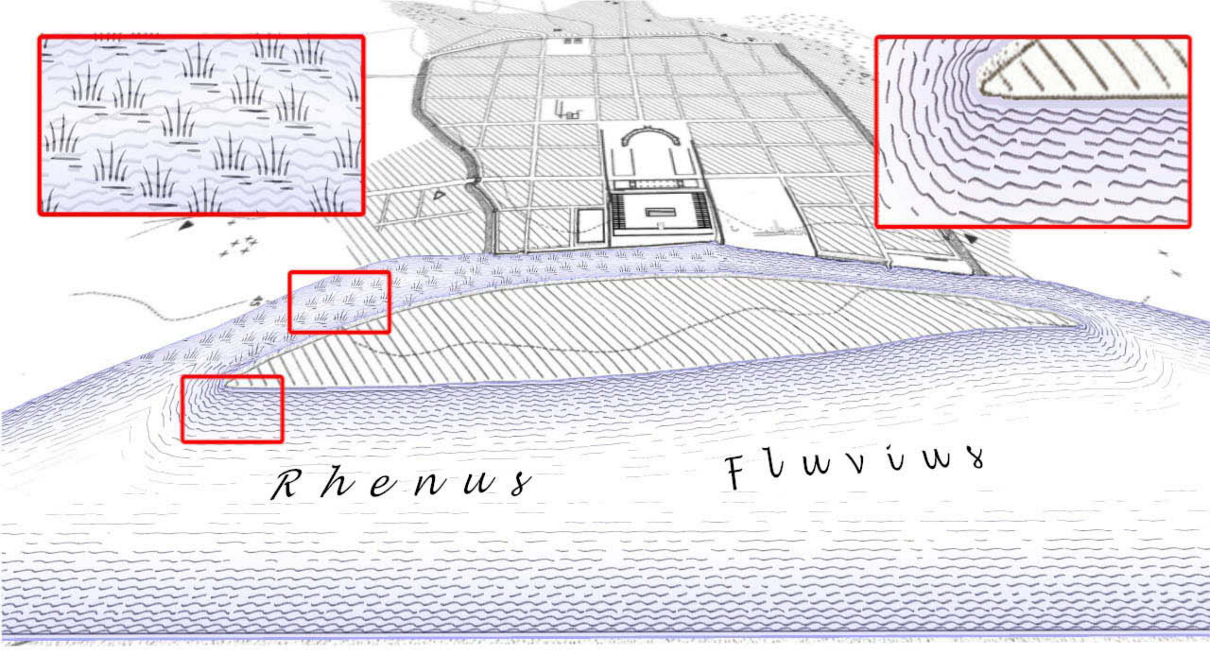


Results – Flooding Simulation

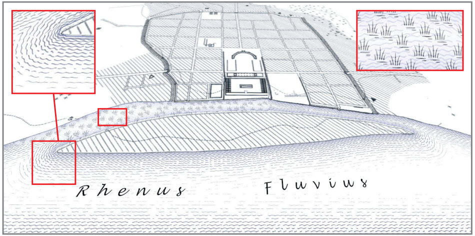
- waterlining expresses uncertainty, conveys motion, and enhances the depiction of land cover
- flooding simulation: assess distances to nearest safety zones for evacuation planning
- a plane is used as clipping mask and temporally shifted upwards to represent the change of the mean sea level, uses ortographic projection to obtain flooded areas

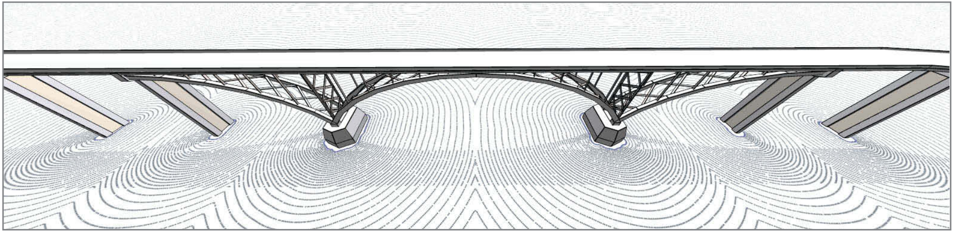


Flooding simulation for the city of Boston


Rhenus
Fluvius

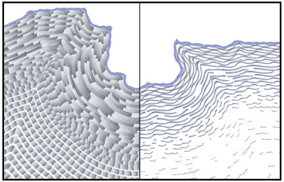
Results – Urban Planning & Landscaping

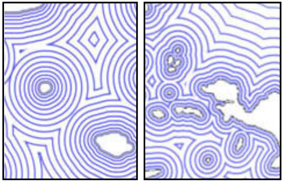




Contour-hatching and water stippling to express uncertainty

Conclusions





- **design principles** from cartography can be used to improve **figure-ground perception**, which requires further **validation**
- shoreline and feature-aligned **distance maps** are suitable for **texture-based** waterlining, water stippling, contour-hatching
- **context-aware parameterization** of shading effects remains an important issue for **view-dependent level-of-abstraction**

Thank You For Your Attention!

- Amir Semmo

  amir.semmo@hpi.uni-potsdam.de

- Computer Graphics Systems Group

  Prof. Dr. Jürgen Döllner

  hpi3d.de youtube.com/hpicgs @hpi3d

HPI · TU berlin · Bundesministerium für Bildung und Forschung · INNOPROFILE UNTERNEHMEN REGION Die BMBF-Innovationsinitiative Neue Länder · 4DnD·Vis

Performance Analysis

Performance evaluation (in *ms*): distance / feature-aligned distance transform (D/T), orientation and medial axis computation.

Image Resolution	D	T	Orient.	M. Axes	Total
128×128	1.6	26.6	0.1	0.7	29.0
256×256	2.2	96.3	0.2	0.8	99.5
512×512	4.5	371.6	0.6	0.9	377.6

Performance evaluation for rendering techniques using different screen resolutions (in frames-per-second).

Screen Resolution	waterlining	stippling	contour-hatching
800×600	534	162	159
1280×720	523	88	84
1600×900	521	59	55
1920×1080	514	42	41