12

Hot Topics in Visualization

Hot Topic 1: Illustrative visualization

Illustrative visualization: computer supported interactive and expressive visualizations through abstractions as in traditional illustrations.

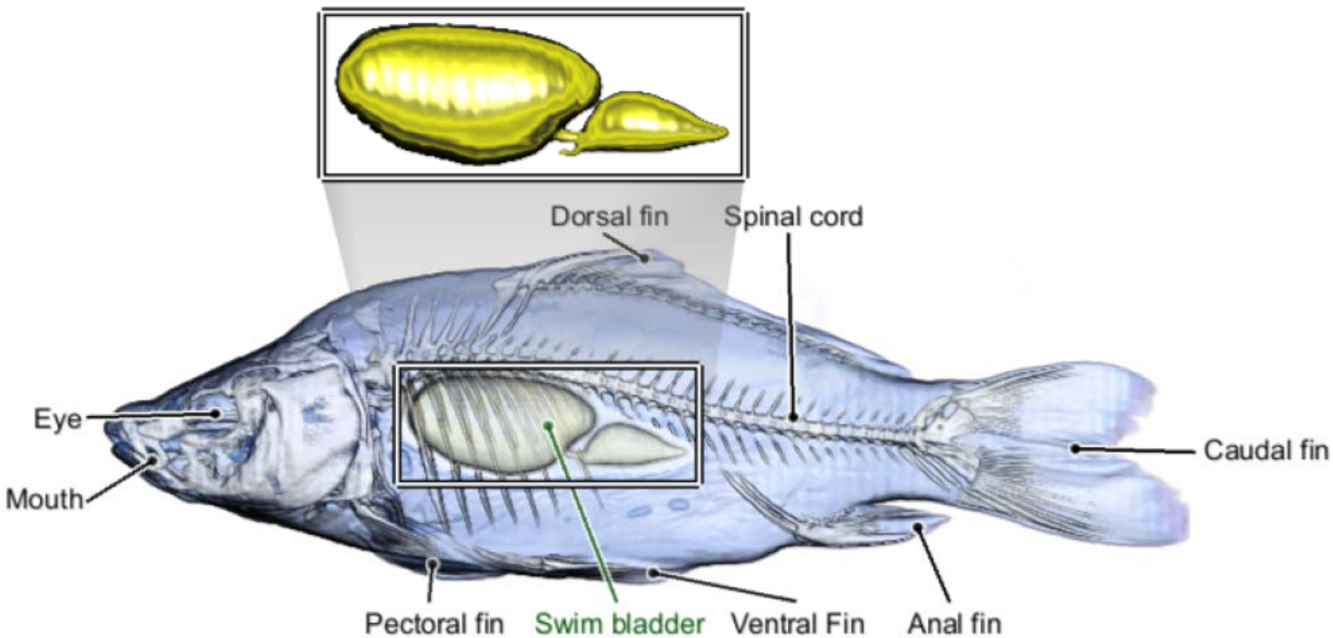


Image credit: S. Bruckner

Illustrative visualization uses several non-photorealistic rendering (NPR) techniques:

- smart visibility
- silhouettes
- hatching
- tone shading
- focus+context techniques
  - context-preserving volume rendering

Smart visibility

Abstraction techniques:

- cut-aways (a)
- ghosted views (b)
- section views (c)
- exploded views (d)

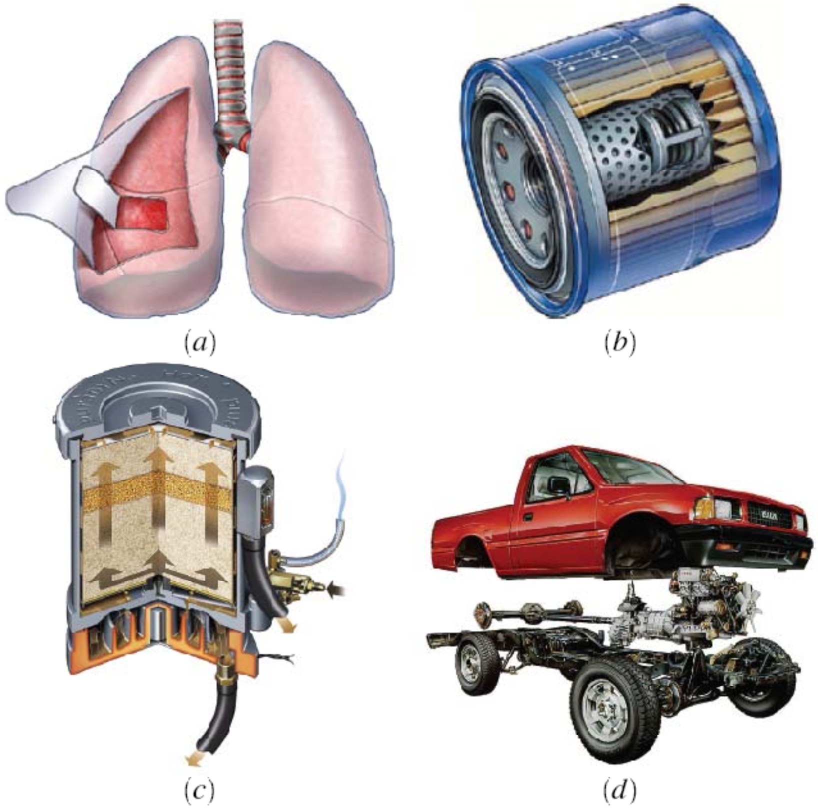
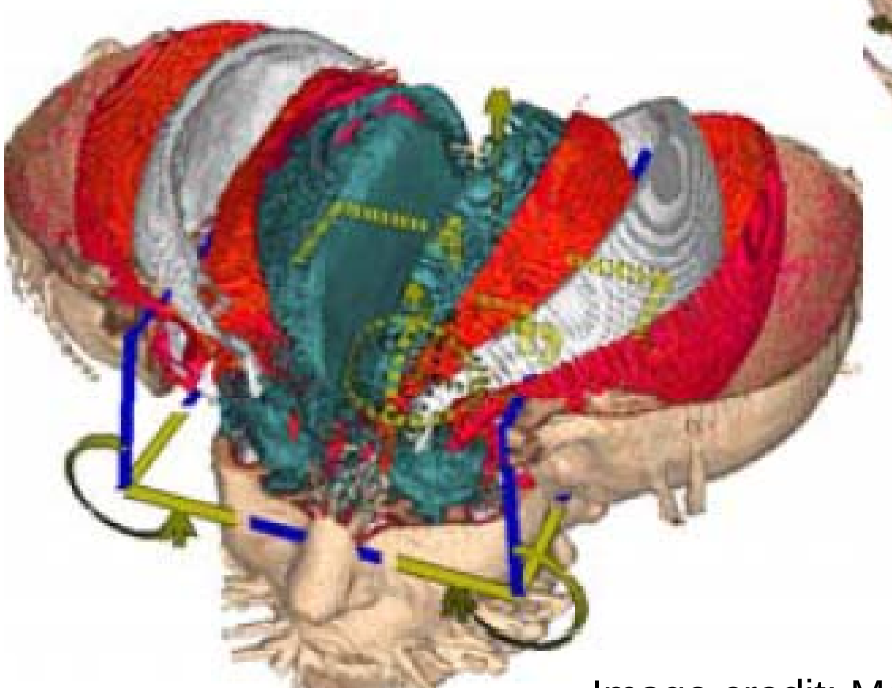


Image credit: K. Hulsey Illustration Inc.

Browsing deformations:

Peeler

Leafer



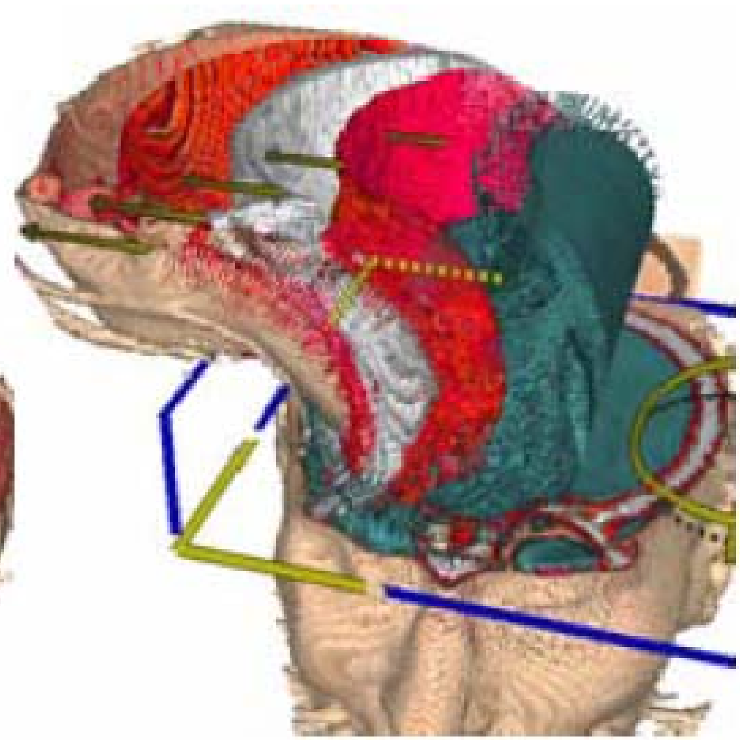


Image credit: McGuffin et al.

Silhouette algorithms

The silhouette of a surface consists of those points where view vector **V** and surface normal **N** are orthogonal.

Silhouettes can be either outlines or internal silhouettes.

In contrast to other important feature lines such as curvature ridges/valleys and texture boundaries, silhouettes are view-dependent.

Object space algorithms exist for:

- polygonal surfaces. Principle:

  for each polygon

  - set front-facing flag to all edges if $\mathbf{N} \cdot \mathbf{V} \geq 0$
  - set back-facing flag to all edges if $\mathbf{N} \cdot \mathbf{V} < 0$

  for each edge

  - draw if both flags are set

  (assumes triangles or planar quads)

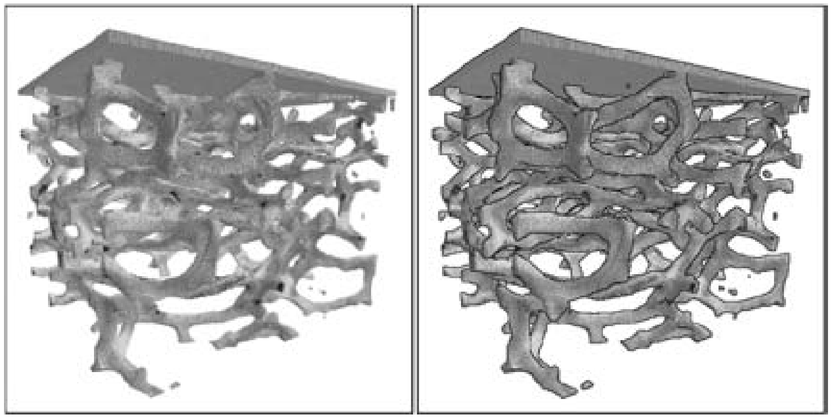
- implicit surfaces
- NURBS surfaces

Image space algorithms:

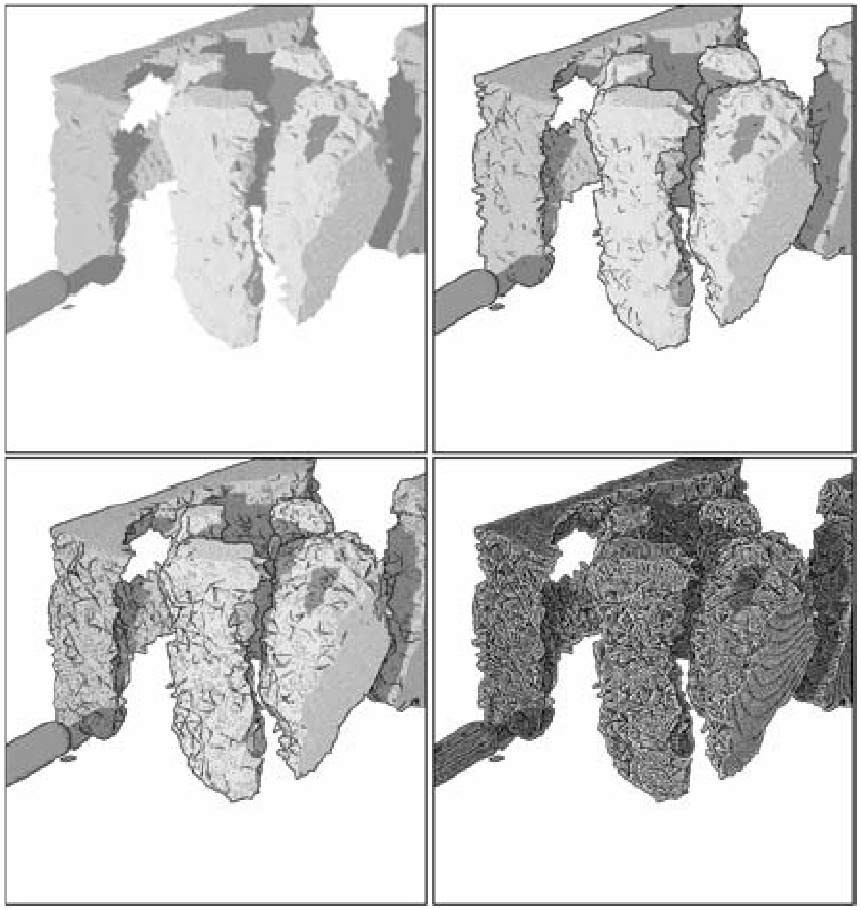
- for polygonal surfaces
  - render polygons with depth buffer enabled
  - look for discontinuities in depth buffer:
    - compute depth difference between two adjacent pixels, or the Laplacian on a 3x3 stencil
    - if larger than threshold, draw a silhouette pixel
- for volume data (Ebert and Rheingans).
  - idea: "silhouette points" are where the gradient is orthogonal to the view vector
  - use opacity transfer function depending on $\left|\nabla s \cdot \mathbf{V}\right|$

Example: case study (Bigler)

without silhouettes | with silhouettes



different thresholds



Example: Silhouettes in volumes

(DVR without lighting!)

- Skin is transparent in non-silhouette regions to avoid visual obstruction

- Bones are darkened along silhouettes to emphasize structure

Focus on ankle joints

Image credit: N. Svakhine and D. Ebert

Hatching

Surface rendering with
hatching techniques:

- shading and shadows (Winkenbach/Salesin)

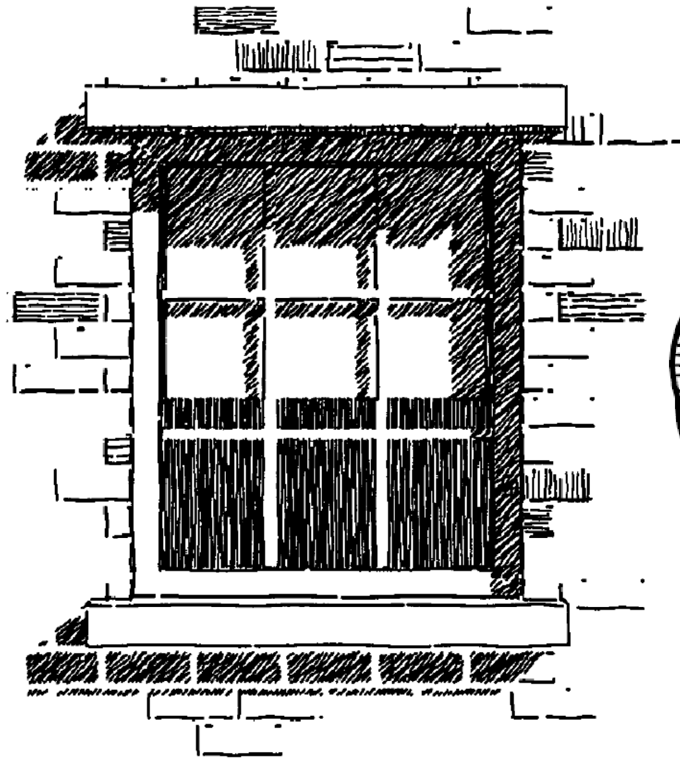


Image credit: G. Winkenbach

- smooth surfaces (Hertzmann/Zorin)

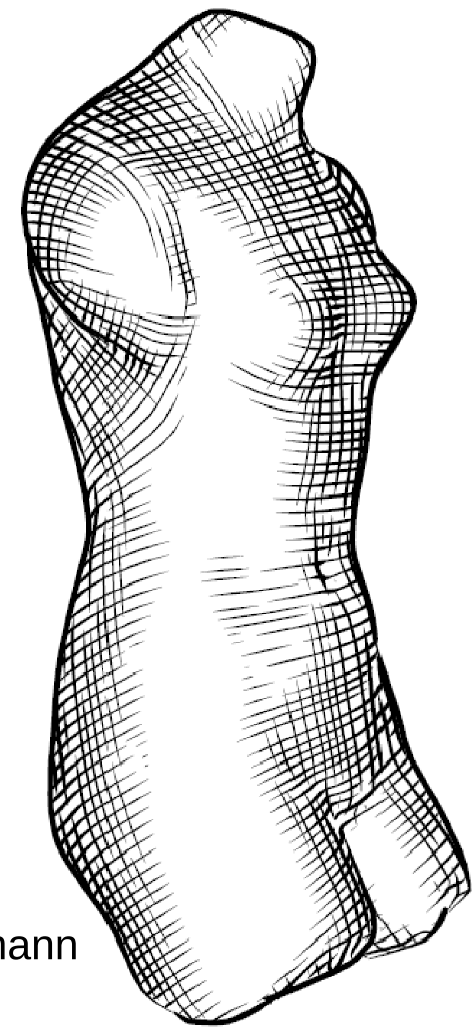
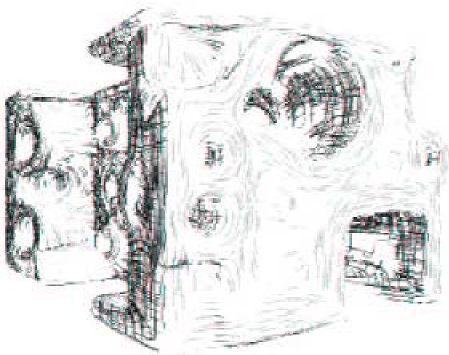
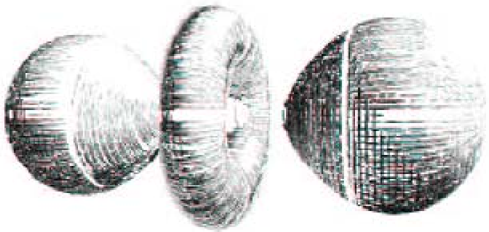


Image credit: A. Hertzmann

Volume illustration with hatching (Nagy):

- compute an isosurface
- compute curvature fields (1st and 2nd principal curvature directions on the isosurface), fast algorithm by Monga et al.
- compute hatching as streamlines of both curvature fields, using streamline placement techniques





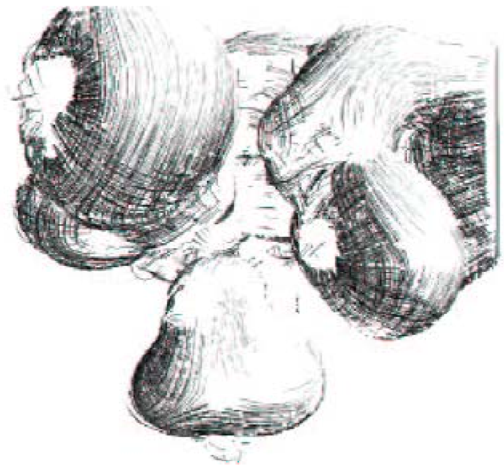
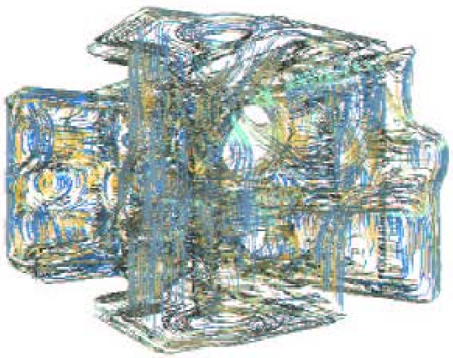
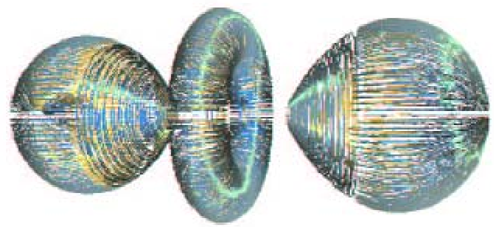


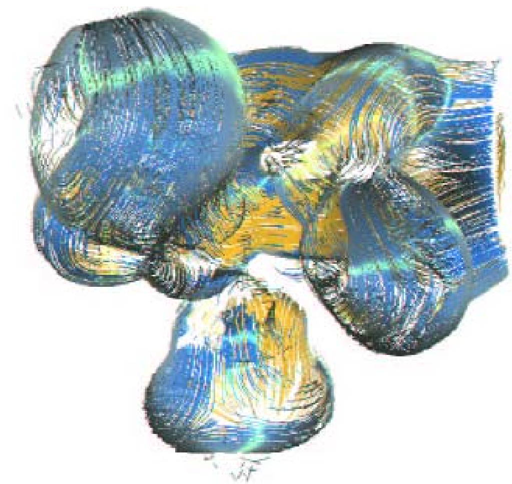
Image credit: Z. Nagy

Hatching

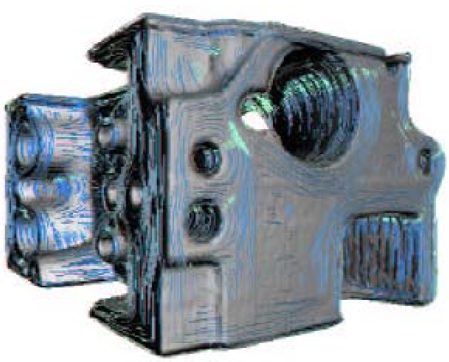
- render streamlines as illuminated lines

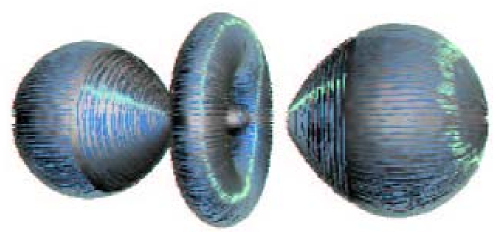






- overlay with volume rendering





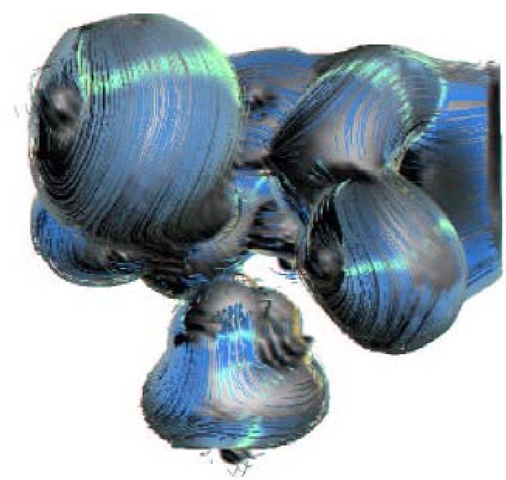


Image credit: Z. Nagy

Tone shading

Tone shading or "toon shading" (cartoons) uses tones instead of luminance for shading.

Examples: Warm to cool hue shift

Gray model, tone shaded

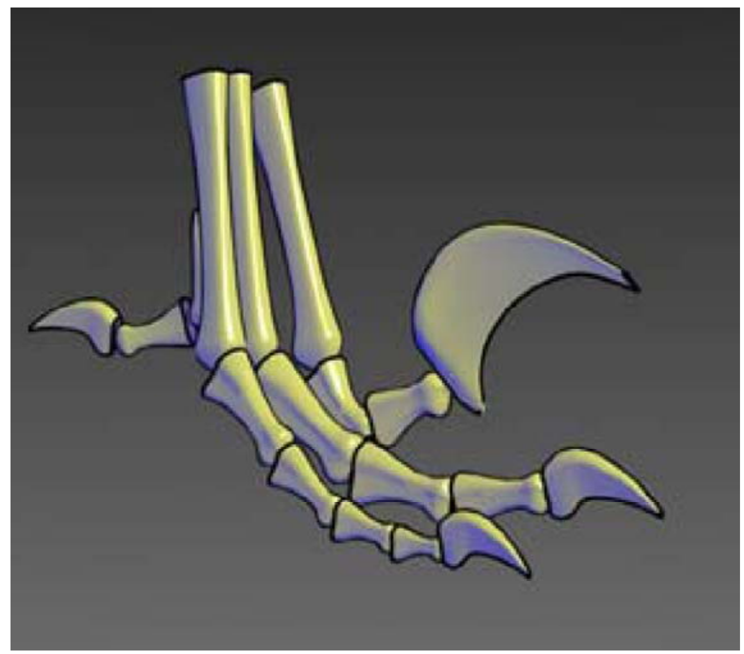


Image credit: A. Gooch

Depth cue: warm colors advance while cool colors recede.

Tone shading

Phong shading

vs.

tone shading

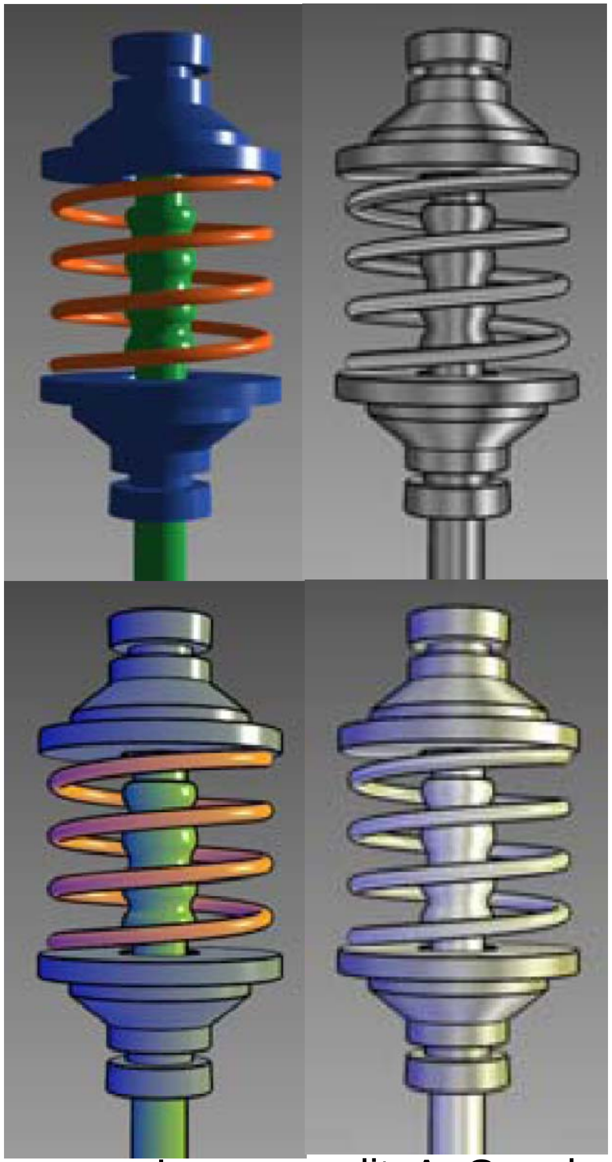
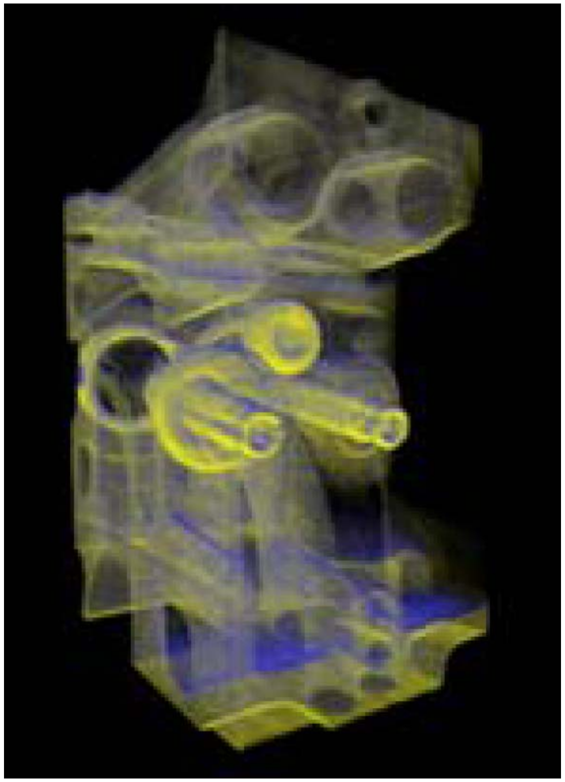


Image credit: A. Gooch

Tone shaded volume rendering



Context-preserving volume rendering

Ghosted view: surface transparency depends on the grazing angle (angle between view ray and surface).

More transparent for large, more opaque for small grazing angle.

Example:

Image credit: K. Hulsey Illustration Inc.

Context-preserving volume rendering (Bruckner):

Use of ghosted views in volume rendering:

Overview of context-preserving volume rendering model:

gradient magnitude, normalized

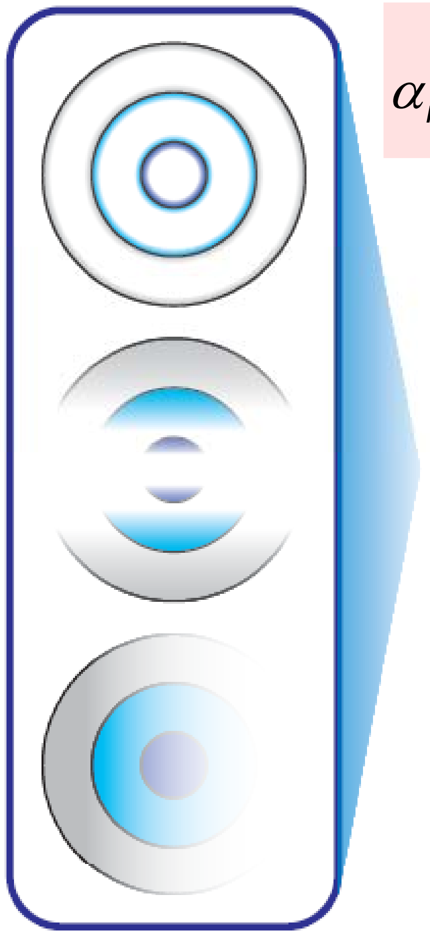
$$\|\nabla s(\mathbf{x})\|_{[0..1]}$$

shading intensity

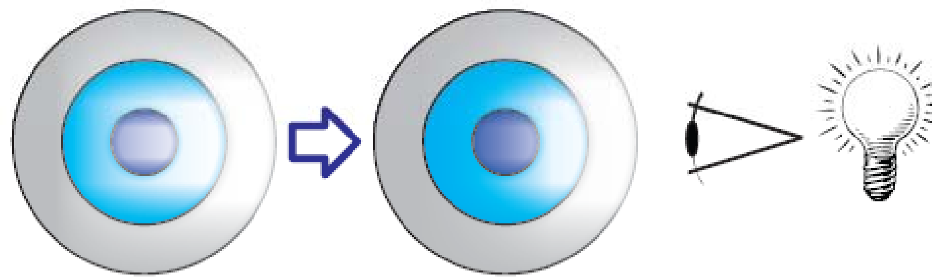
$$\sigma(\mathbf{x})$$

eye distance, normalized

$$\|\mathbf{x}-\mathbf{x}_{eye}\|_{[0..1]}$$



$$\alpha_i = \alpha(s(\mathbf{x}))\|\nabla s(\mathbf{x})\|_{[0..1]}^{\left(\kappa_t\sigma(\mathbf{x})\left(1-\|\mathbf{x}-\mathbf{x}_{eye}\|_{[0..1]}\right)(1-\alpha_{i-1})\right)^{\kappa_s}}$$



previously accumulated opacity α_{i-1}

Image credit: S. Bruckner

$$\alpha_i = \alpha\left(s\left(\mathbf{x}\right)\right)\left\|\nabla s\left(\mathbf{x}\right)\right\|_{[0..1]}^{\left(\kappa_t \sigma\left(\mathbf{x}\right)\left(1-\left\|\mathbf{x}-\mathbf{x}_{eye}\right\|_{[0..1]}\right)\left(1-\alpha_{i-1}\right)\right)^{\kappa_S}}$$

High shading intensity (of local Phong lighting model with light source at eye point) means: large grazing angle. It results in higher transparency.

Parameters

- κ_t corresponds roughly to the depth of a clipping plane
- κ_s controls the sharpness of the transition between visible and clipped

Context-preserving volume rendering

context-preserving VR

vs.

medical illustration

Image credit: S. Bruckner

Image credit: Nucleus Medical Art, Inc.

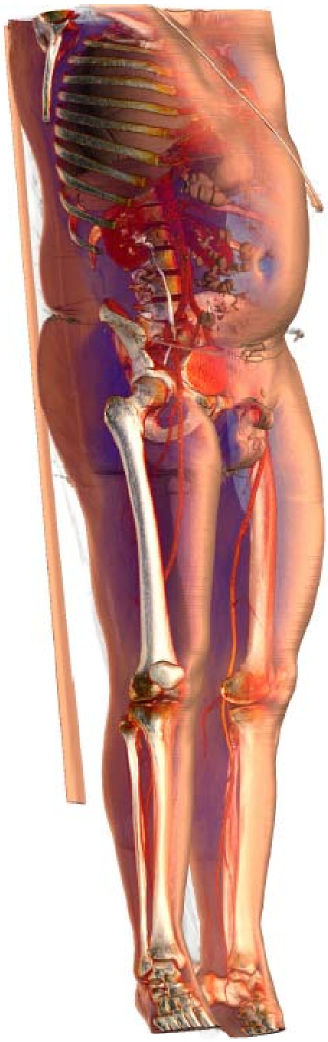


Image credit: S. Bruckner

Hot Topic 2: Lagrangian Coherent Structures

Motivation: Vector field topology does not well describe the topology of a "strongly" time-dependent vector field.

- Separatrices are defined in terms of streamlines, not pathlines, i.e. by integrating the instantaneous vector field.
- Critical points of saddle type are not the places where flow separation happens.

Example: "Double gyre" [S. Shadden]

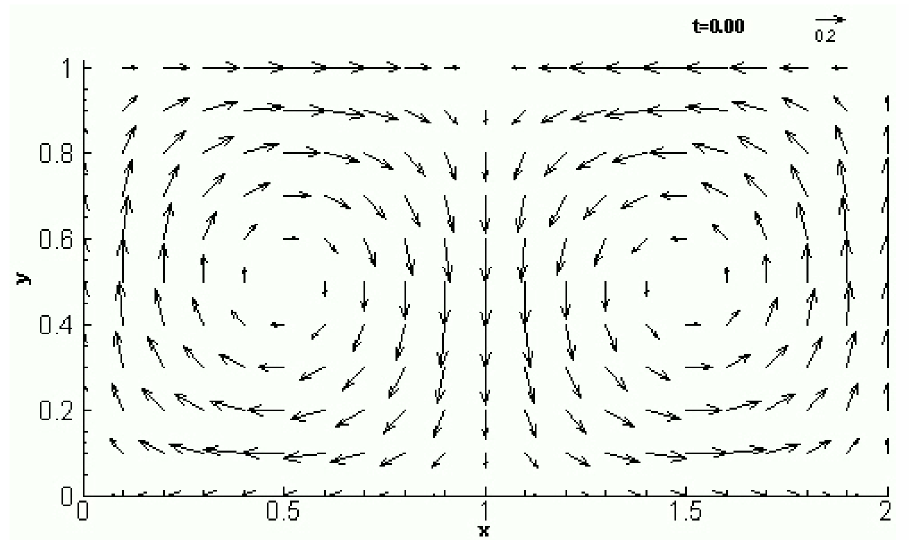
$$u(x,y,t) = -\pi A \sin(\pi f(x,t)) \cos(\pi y)$$

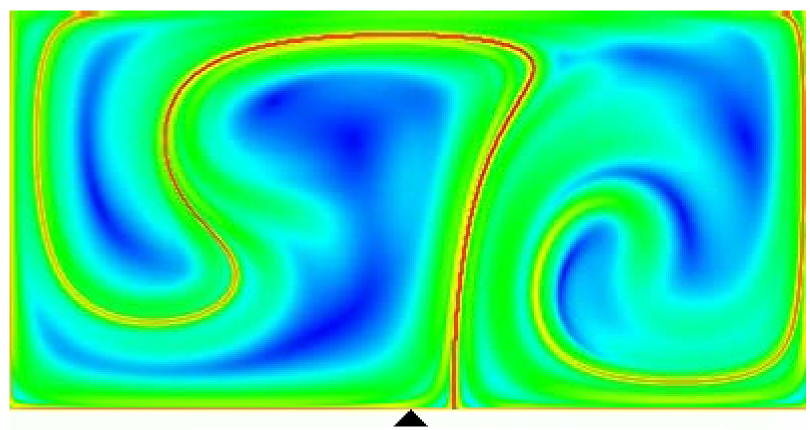
$$v(x,y,t) = \pi A \cos(\pi f(x,t)) \sin(\pi y) \frac{df(x,t)}{dx}$$

$$f(x,t) = \varepsilon \sin(\omega t) x^2 + (1 - 2\varepsilon \sin(\omega t)) x$$

The vector field (with parameters $A = 0.1$, $\omega = 2\pi/10$, $\varepsilon = 0.25$)

Lagrangian coherent structures (the red pixels approximate a material line).





topological saddle point

An LCS in nature.

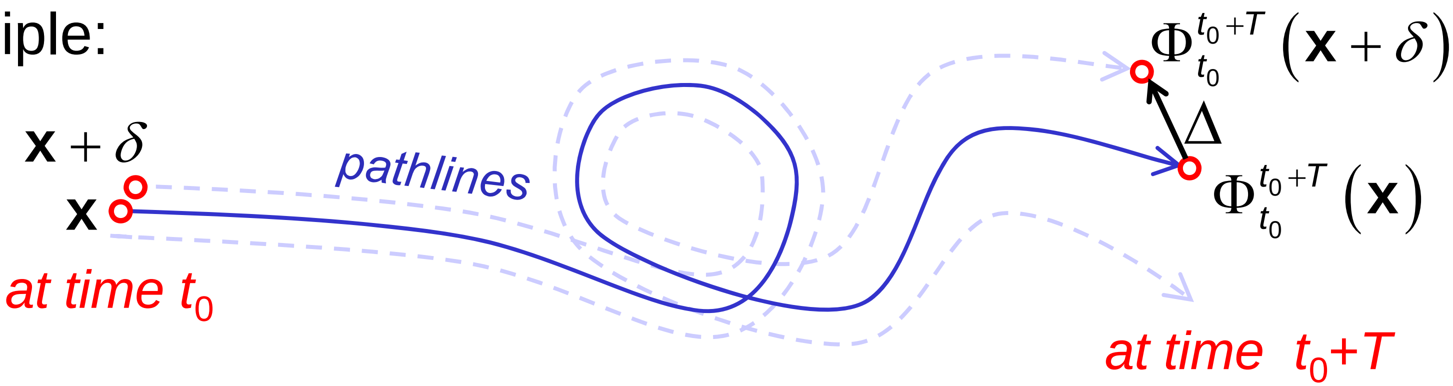
How to find the separating line (or surface)?

Idea: Integrate backward and detect large amount of separation.

The finite-time Lyapunov exponent

The FTLE describes the amount of separation (stretching) after a finite advection time *T*.

Principle:



Definition:

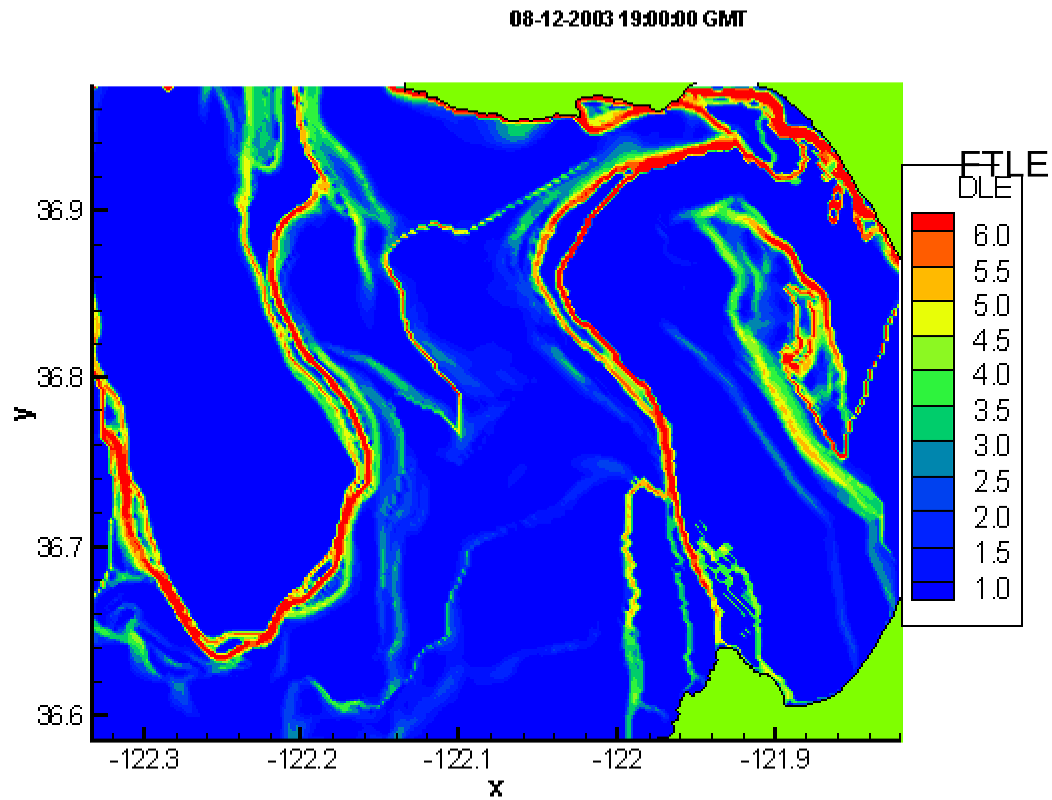
$$\mathrm{FTLE}(\mathbf{x},t_0,T) = \lim_{\delta\to 0} \max_{\text{direction of } \delta} \frac{1}{|T|} \ln \frac{\|\Delta\|}{\|\delta\|}$$

$$= \frac{1}{|T|} \ln\left(\left\|\nabla \Phi_{t_0}^{t_0+T}(\mathbf{x})\right\|_2\right) \qquad \left(\|A\|_2 = \sqrt{\lambda_{\max}\left(A^T A\right)}\right)$$

LCS as FTLE ridges

Definition (G. Haller): LCS are (height) ridges of the FTLE field.

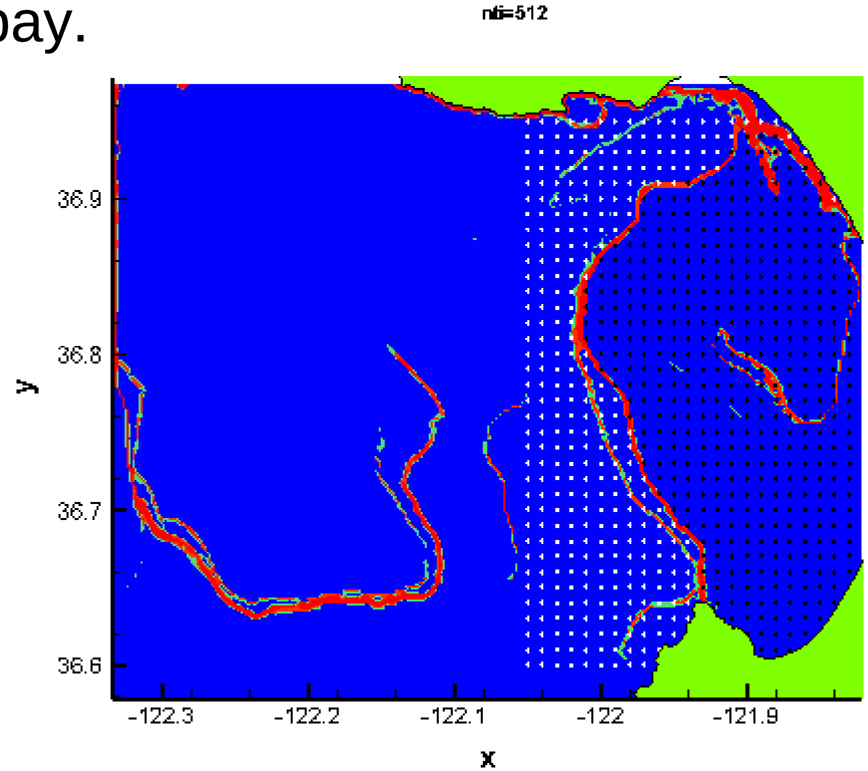
Example: Ocean currents in Monterey Bay.



(Video: S. Shadden)

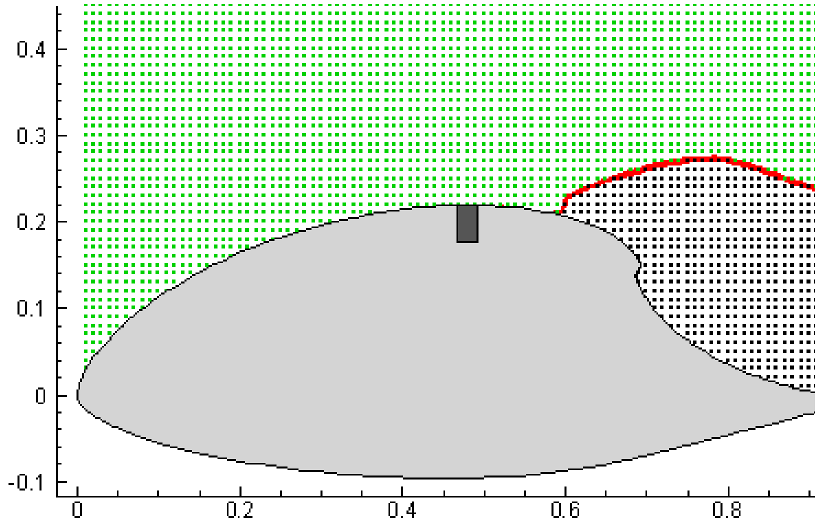
LCS are material lines (or material surfaces).

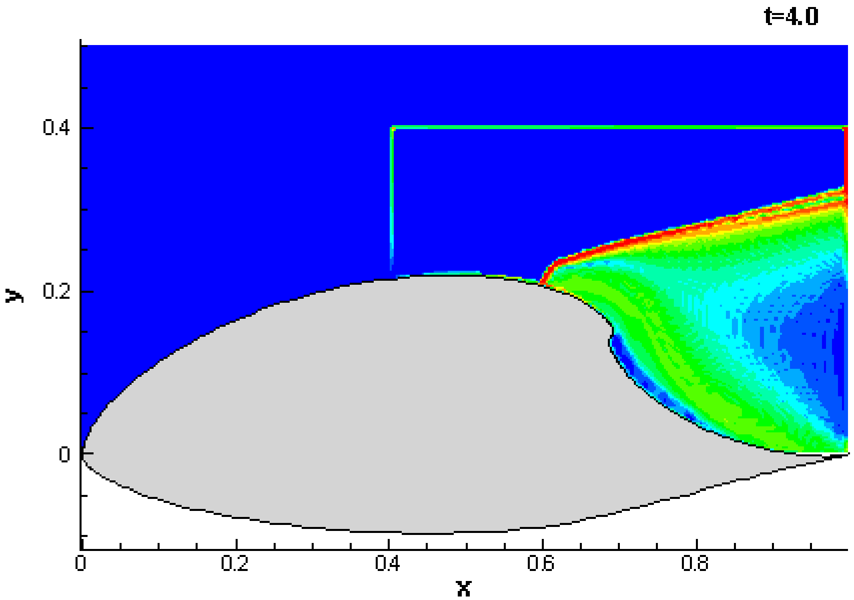
Example: The LCS separates recirulating flow from flow which leaves the bay.



(Video: S. Shadden)

Example: Flow over an airfoil with active flow control.



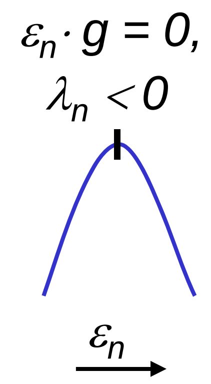


(Videos: S. Shadden)

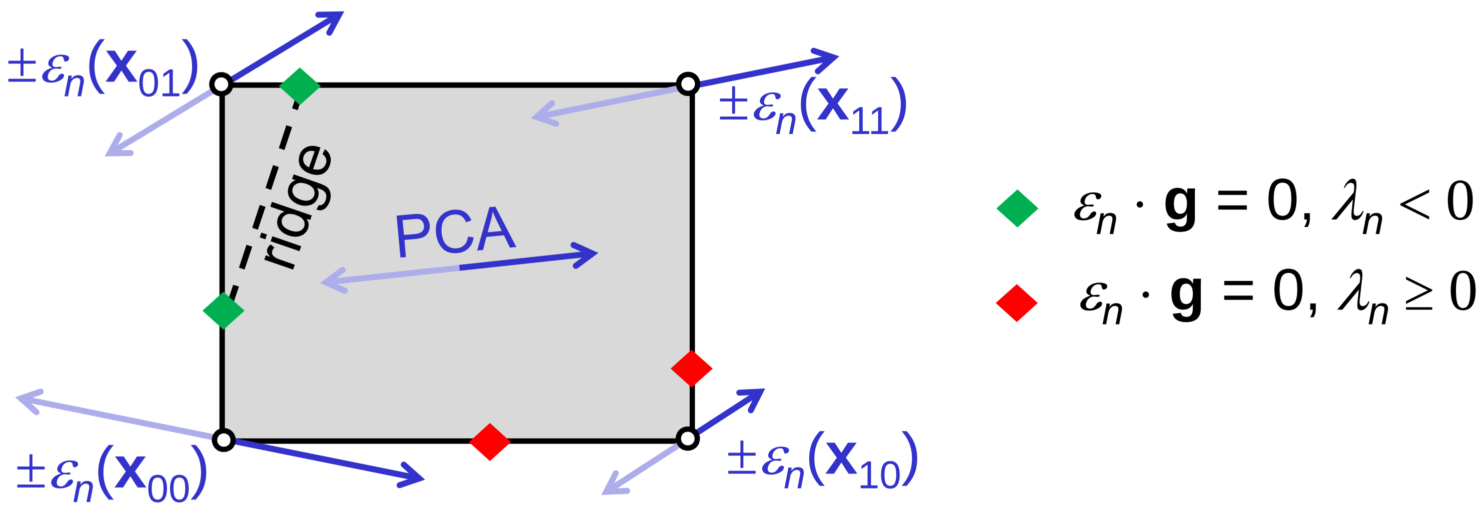
Ridge computation

Efficient computation of height ridges (of a scalar field $s(\mathbf{x})$ in n-space):

- compute derived fields $\mathbf{g} = \nabla s$, $\mathbf{H} = \nabla \mathbf{g}$
- for ridges of dimension 1 use Parallel Vectors method:
  - find places where **g** and **Hg** are parallel vectors
  - test if 2nd directional derivative is negative in directions $\perp \mathbf{g}$
- for ridges of co-dimension 1 (i.e. of dimension n-1) use Marching Ridges method (Furst et al. 2001):
  - compute eigenvalues of **H**: $\lambda_1 \geq \ldots \geq \lambda_n$
  - ε_n : eigenvector for λ_n ($\varepsilon_n \perp$ ridge)
  - solve for $\varepsilon_n \cdot \mathbf{g} = 0$ (single scalar equation!)

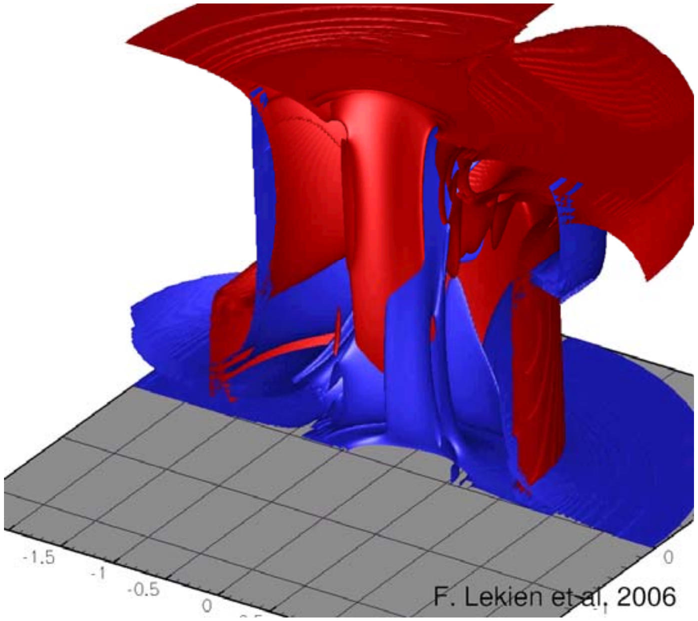


- Problem: ε_n is not a vector field (ambiguous directions). Marching Ridges does the following per cell:
  - orient ε_n at nodes of cell by PCA
  - evaluate $\varepsilon_n \cdot \mathbf{g}$ at nodes
  - interpolate zero crossings on edges
  - use zero crossings with $\lambda_n < 0$
  - generate triangles for *Marching Cubes* case



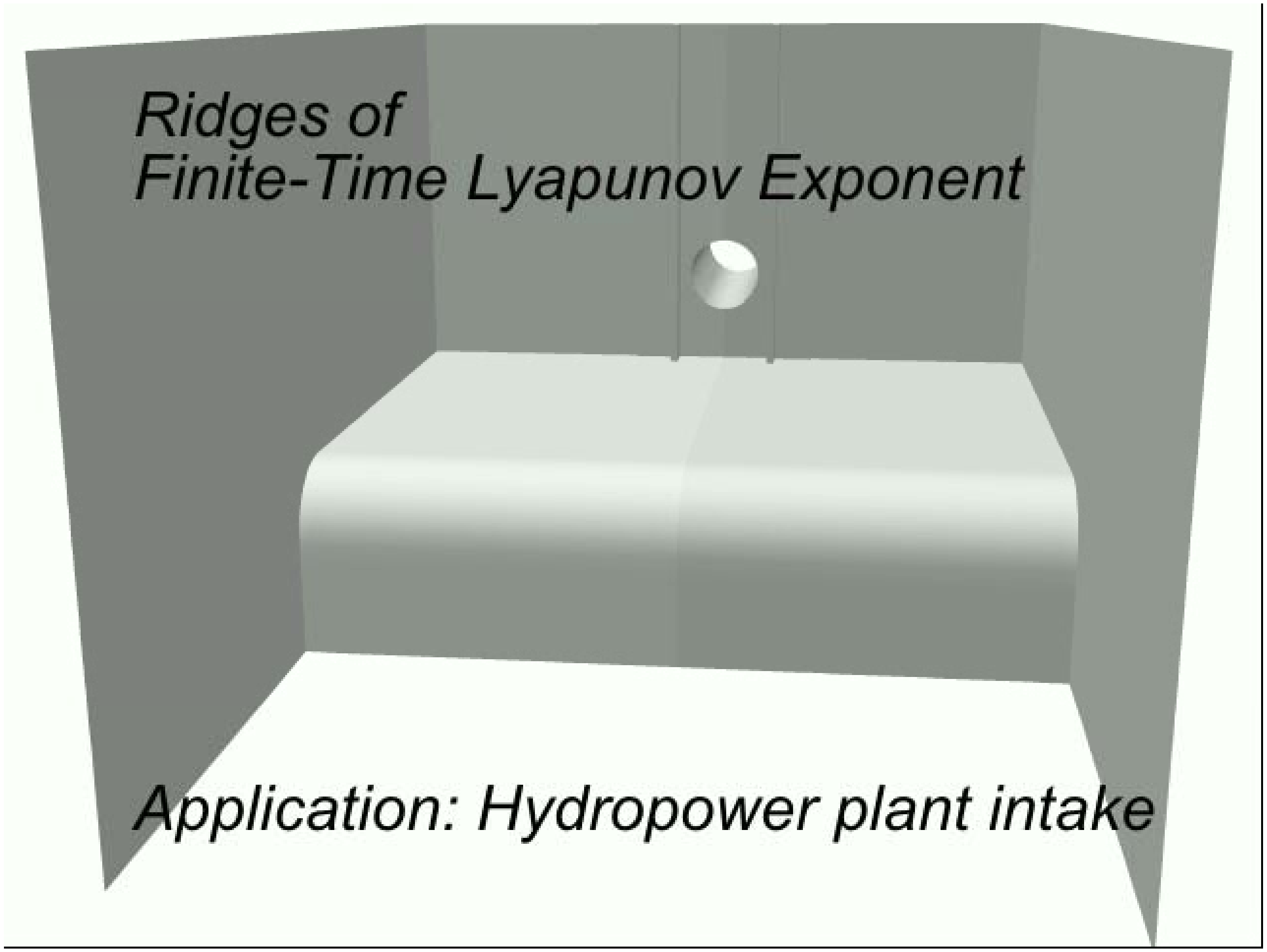
LCS as separation surfaces in 3D

Example: 3D simulation data (Rayleigh-Bénard convection), LCS for positive and negative time.



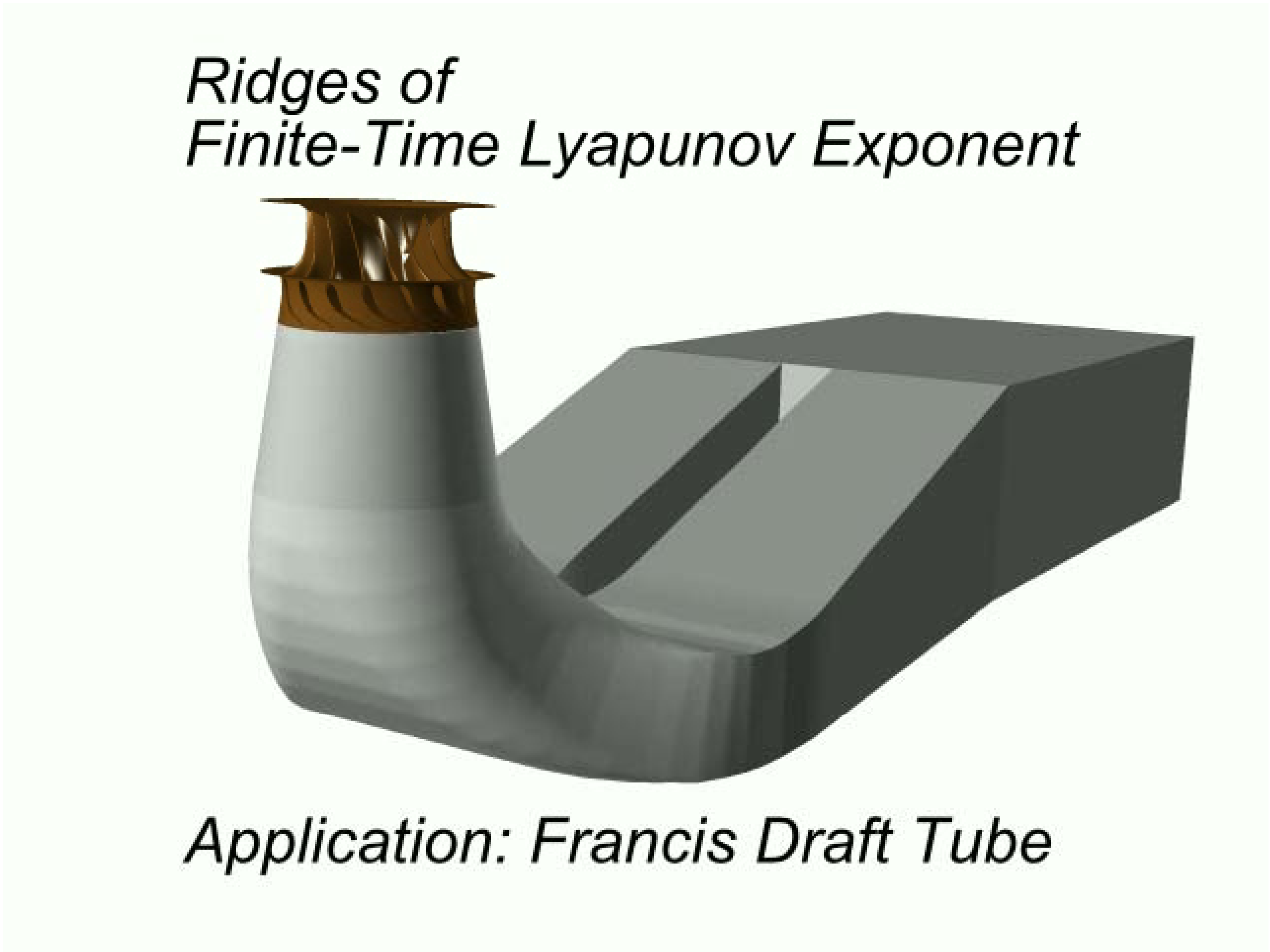
(Image: F. Lekien)

LCS as separation surfaces in 3D



(Video: F. Sadlo)

LCS as separation surfaces in 3D



(Video: F. Sadlo)