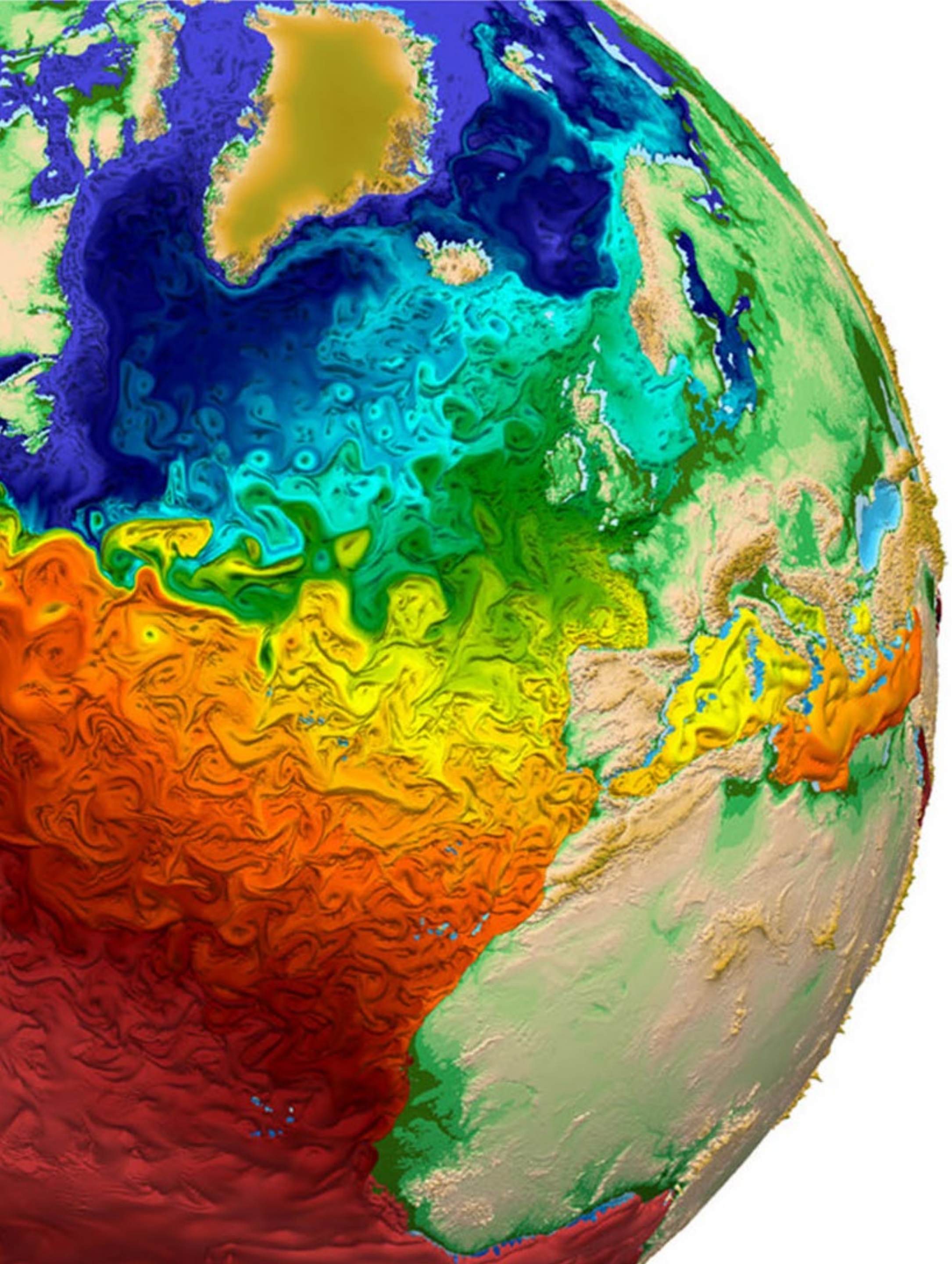


Vector Visualization

Line Integral Convolution

Scientific Visualization

Professor Eric Shaffer

ILLINOIS

Line Integral Convolution (LIC) Principle

So Far

- mostly discrete visualizations (glyphs, streamlines, stream ribbons)

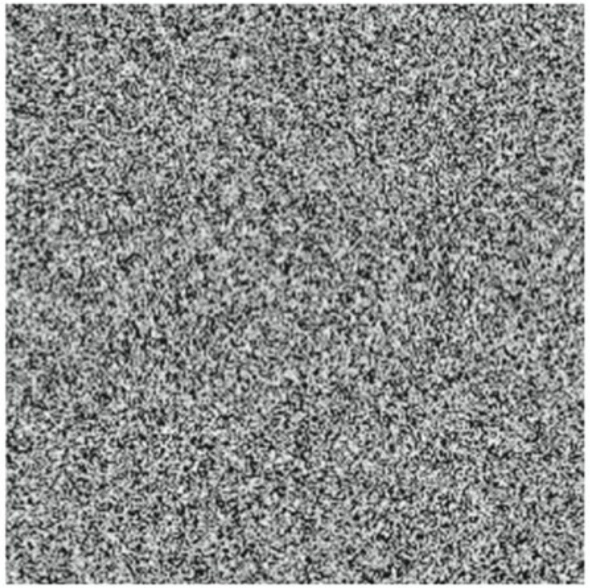
Goal

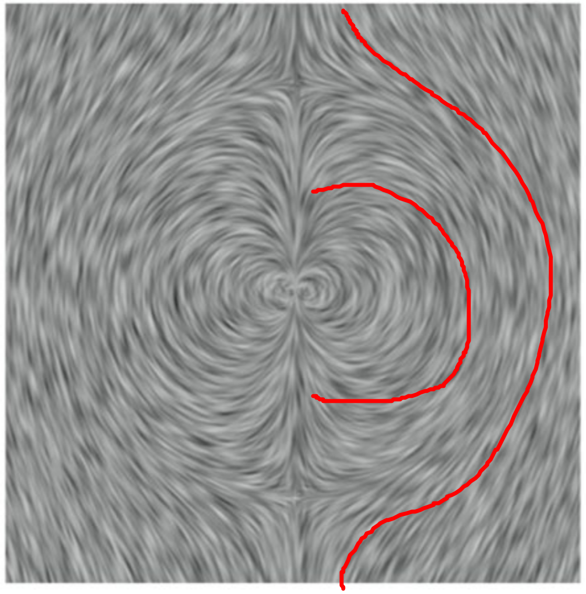
- a dense, pixel-filling, continuous, vector field visualization

Line integral convolution

- highly coherent images along streamlines
- highly contrasting images across streamlines

by blurring noise along the streamlines of v





ILLINOIS

Line Integral Convolution (LIC) Principle

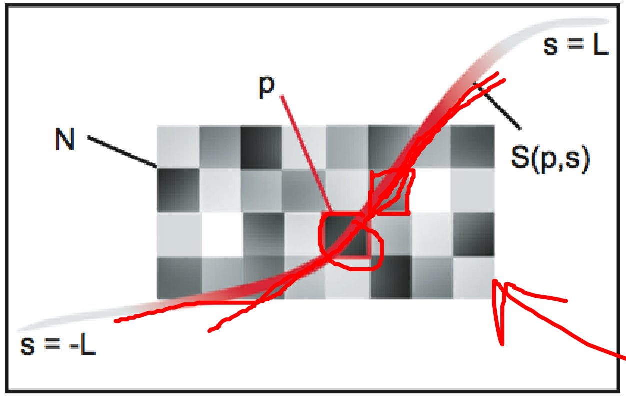
So Far

- mostly discrete visualizations (glyphs, streamlines, stream ribbons)

Goal

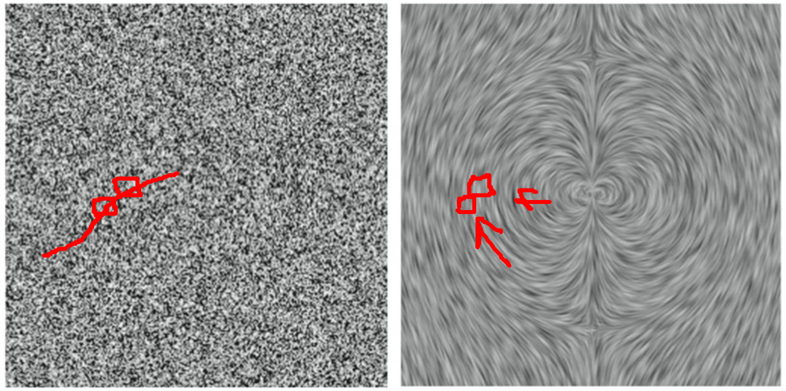
- a dense, pixel-filling, continuous, vector field visualization

Principle



$$T(p) = \frac{\int_{-L}^{L} N(S(p,s))k(s)ds}{\int_{-L}^{L} k(s)ds}$$

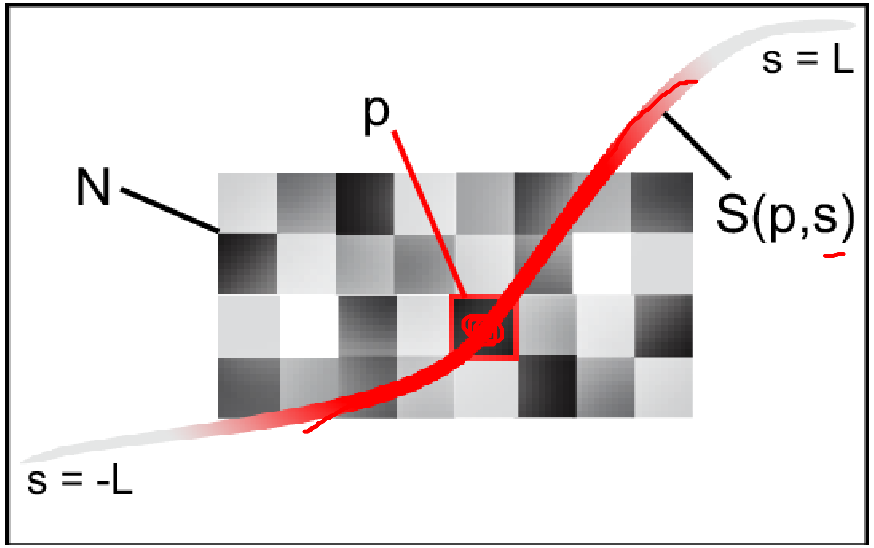
gray value at pixel p
N = noise texture



- take each pixel p of the screen image
- trace a streamline from p upstream and downstream (as usual)
- blend all streamlines, pixel-wise
  - multiplied by a random-grayscale value at p
  - with opacity decreasing (exponentially) on distance-along-streamline from p
- identical to *blurring* noise along the streamlines of $\mathbf{v}$

ILLINOIS

Understanding Line Integral Convolution (LIC)



LIC: Line Integral Convolution

$$T(p) = \frac{\int_{-L}^{L} N(S(p,s))k(s)ds}{\int_{-L}^{L} k(s)ds}$$

$$k(s) = e^{-s^2}$$

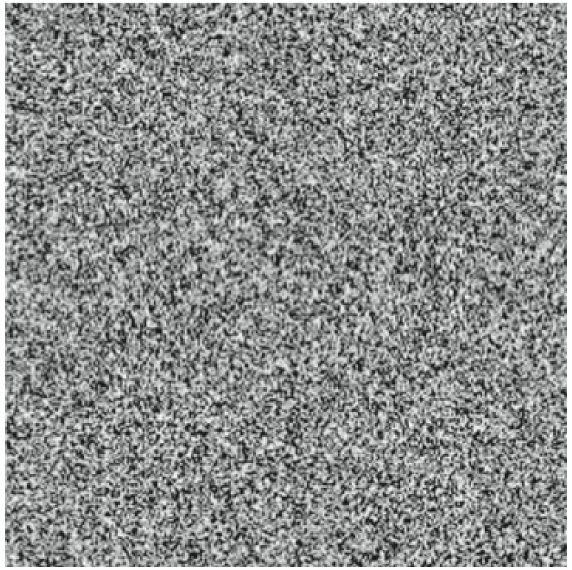
N : noise texture

$S(p,s)$: streamline of seed point P

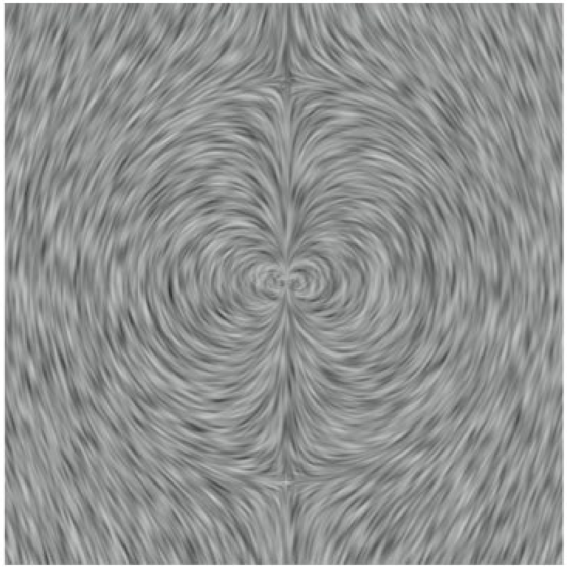
$k(s)$: weighting or blurring function

L : width of blurring function

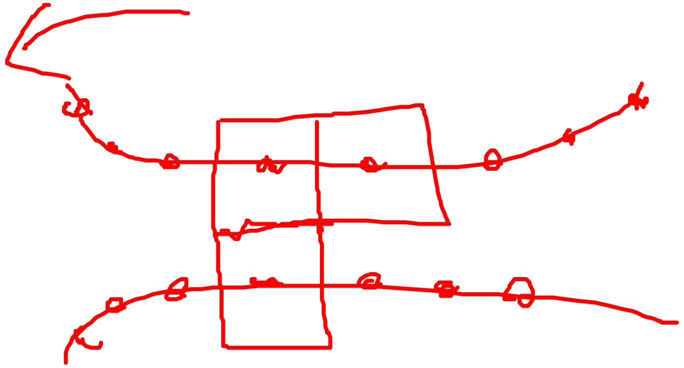
Texture-based Methods



noise texture



line integral convolution (LIC)

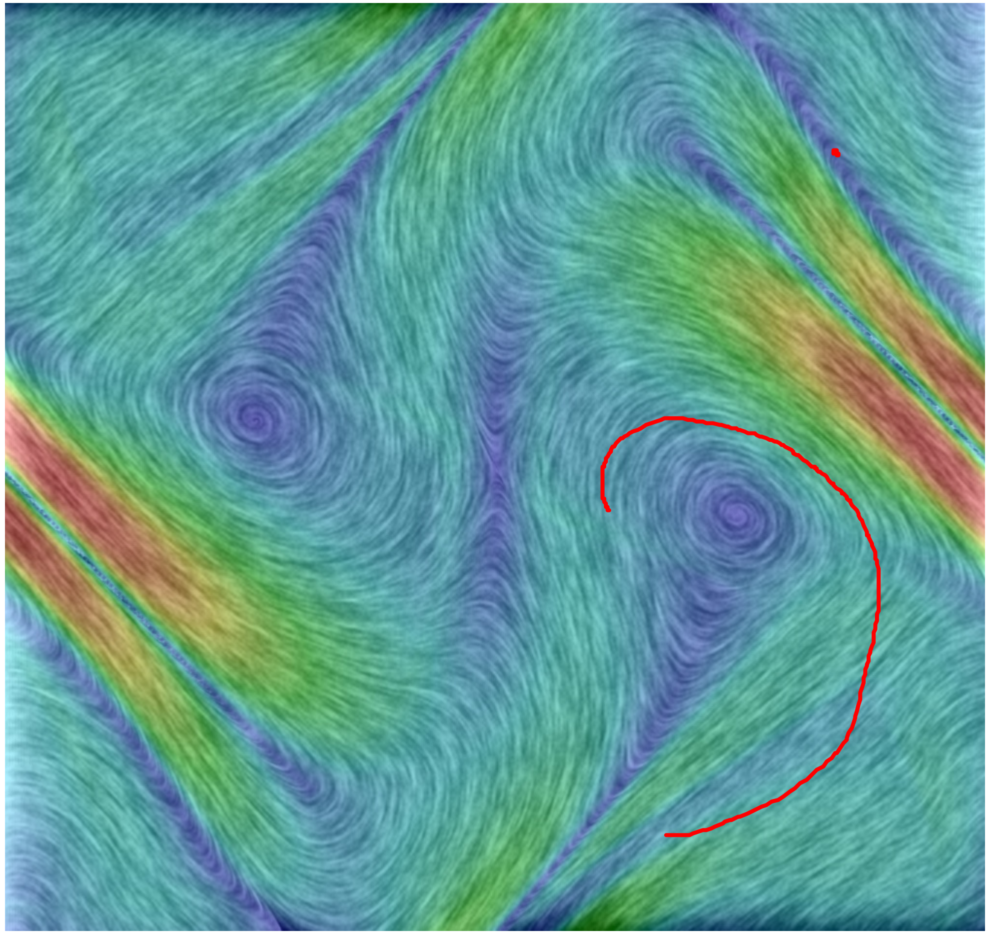


Line integral convolution

- highly coherent images along streamlines
- highly contrasting images across streamlines

- Fast implementation possible using texture-mapping capabilities of modern graphics processing units (GPUs)

Example: LIC with a Colormap



Vector magnitude:
Color

Vector direction:
Graininess