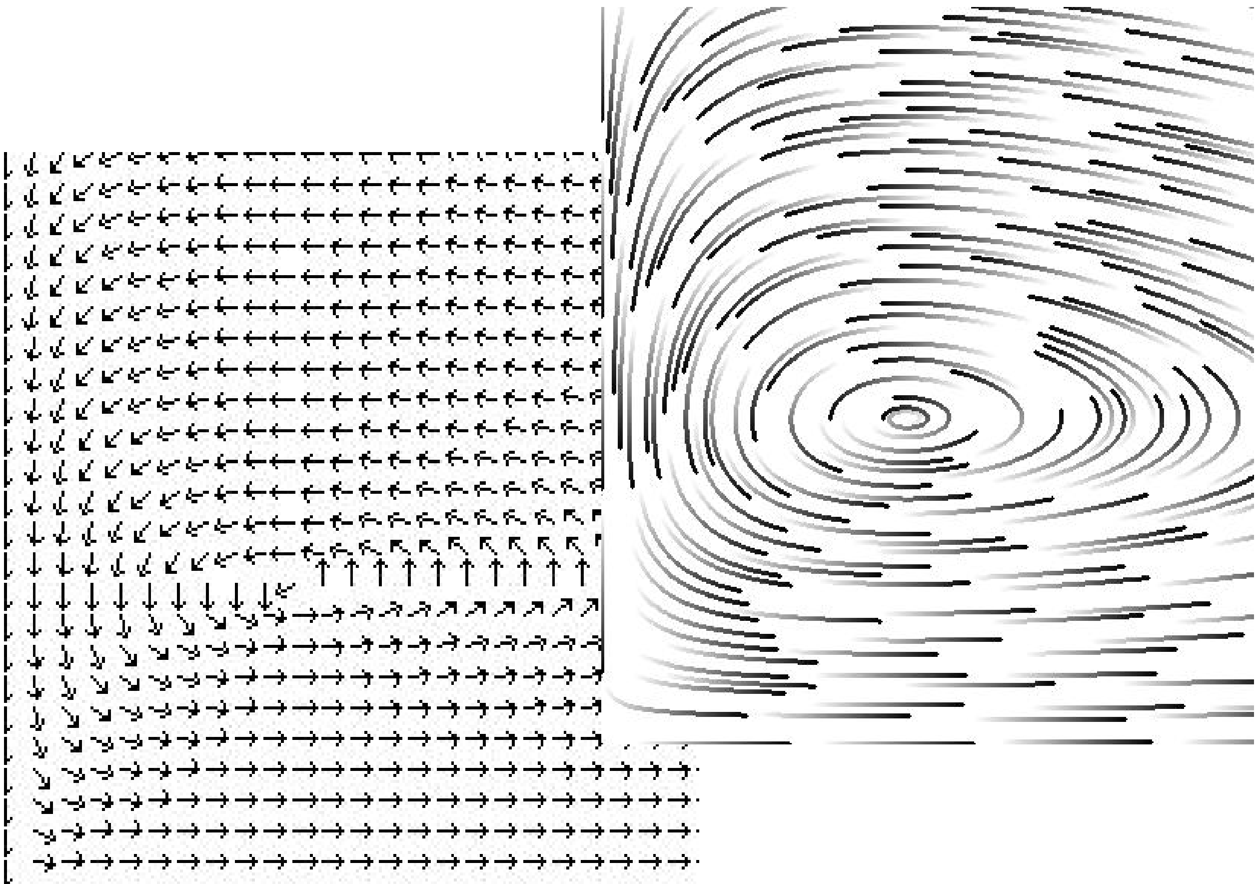
Arrows vs. Streamlines vs. Textures

Streamlines: selective

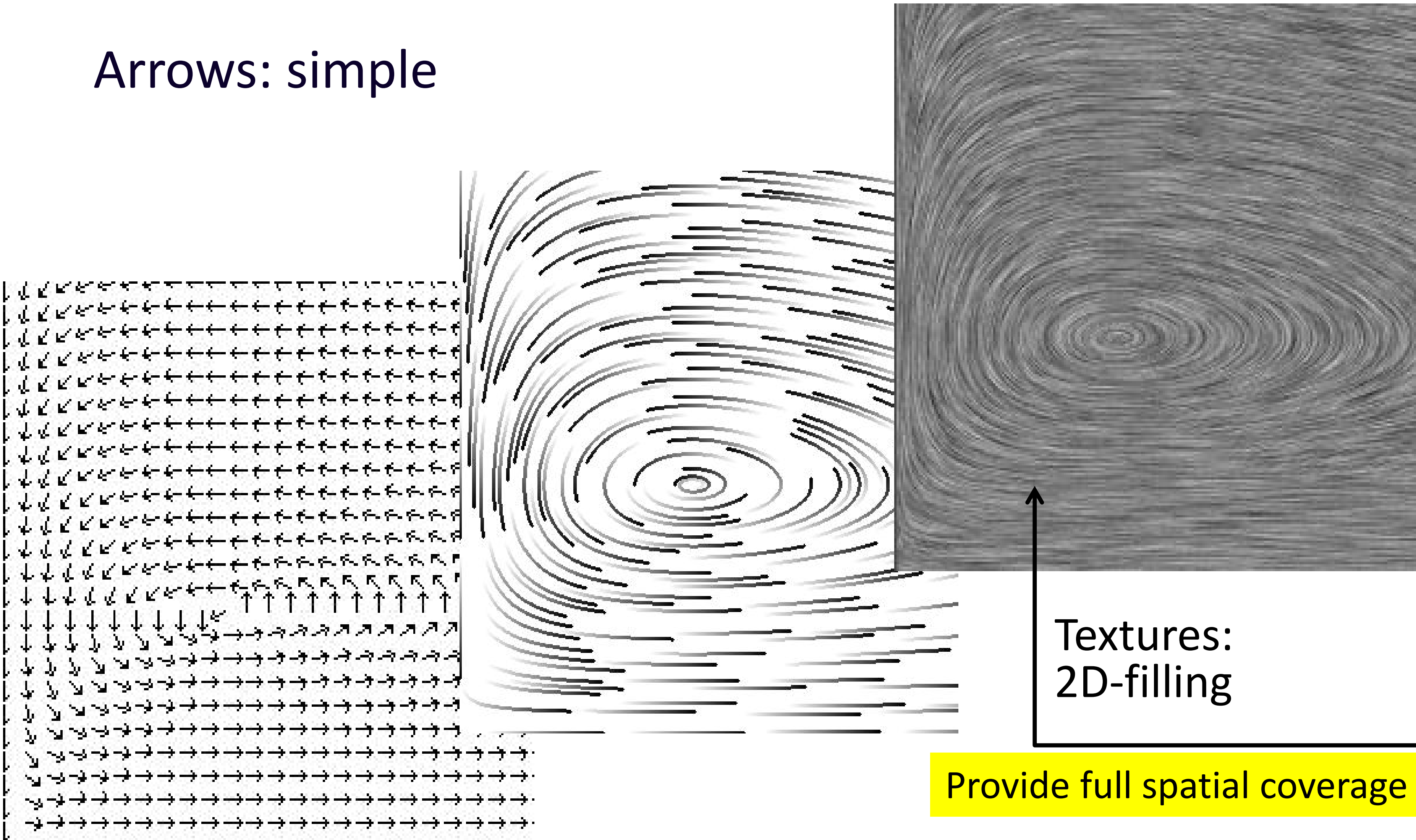
Arrows: simple



Arrows vs. Streamlines vs. Textures

Streamlines: selective

Arrows: simple



Vector Field Visualization: Texture-based Method

Goal: understand the basic idea behind texture-based method (no need to remember all techniques); understand the mechanism of LIC and IBFV; be able to implement LIC

A BRIEF OVERVIEW

Overview — Texture-Based Methods

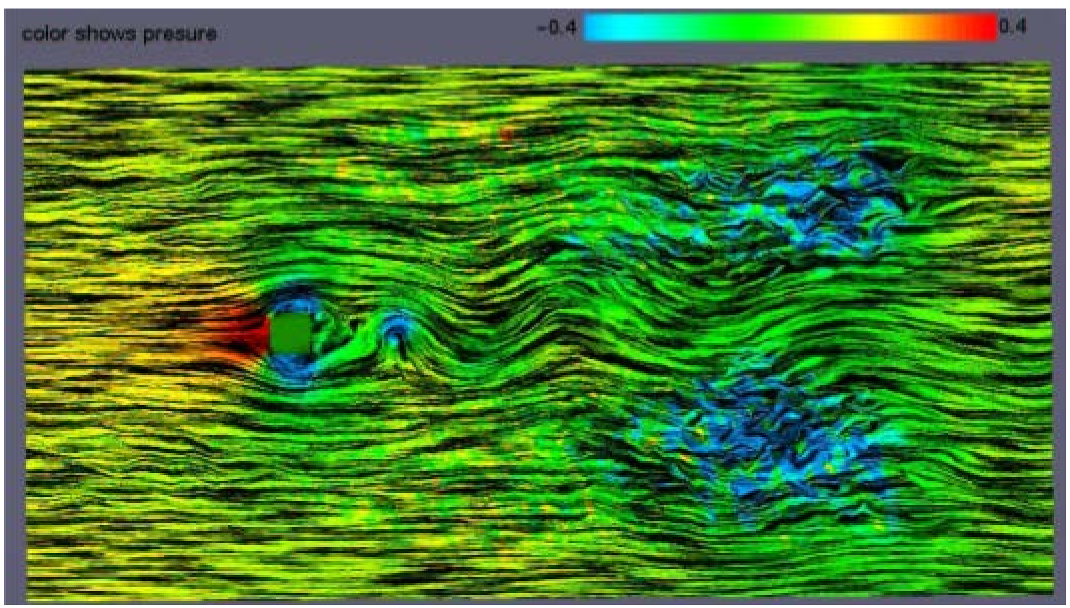
➢ Spot Noise

✧ **One of the first texture-based** techniques (*Van Wijk, Siggraph1991*).

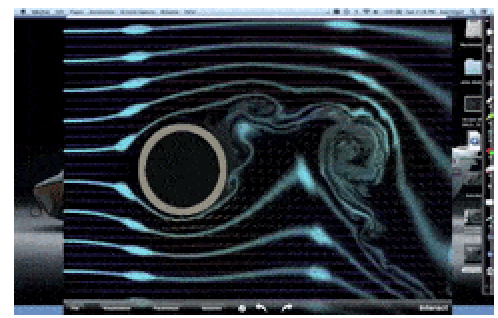
✧ **Basic idea**: *distribute a set of intensity function, or spot, over the domain, that is wrapped by the flow over a small step*.

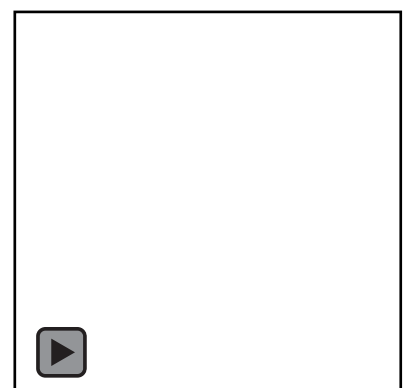
✧ **Pro**: mimic the smear effect of oil; encode magnitude; can be applied for both steady and unsteady flow.

✧ **Con**: tricky to implement; low quality; computationally expensive.



[De Leeuw and Van Liere]





Overview — Texture-Based Methods

➢ Line Integral Convolution (LIC)

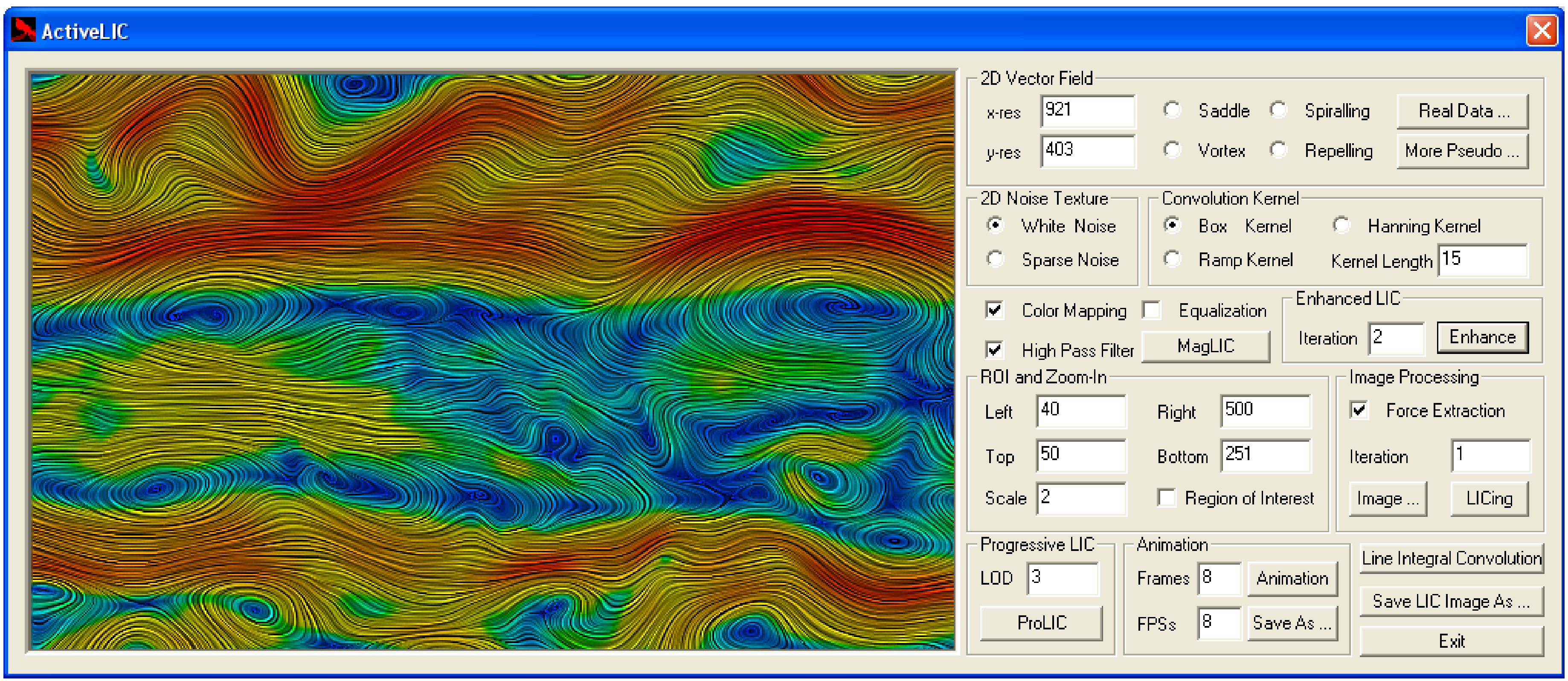
pixel-based

✧ One of the most popular techniques (*Brian Cabral & Leith Leedom, SIGGRAPH93*).

✧ **Basic idea**: *Low-pass filters **white noise** along* pixel-centered symmetrically bi-directional ***streamlines*** to exploit spatial correlation in the flow direction.

✧ **Pro**: High-quality image with fine features; easy implementation; and many variants.

✧ **Con**: Computationally expensive; limited to steady flow visualization.



Overview — Texture-Based Methods

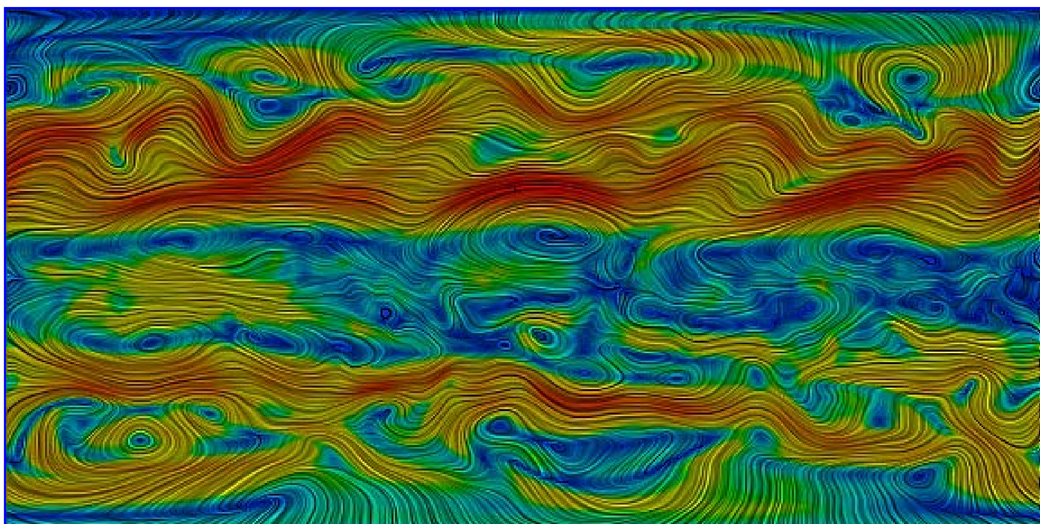
➢ Unsteady Flow LIC (UFLIC)

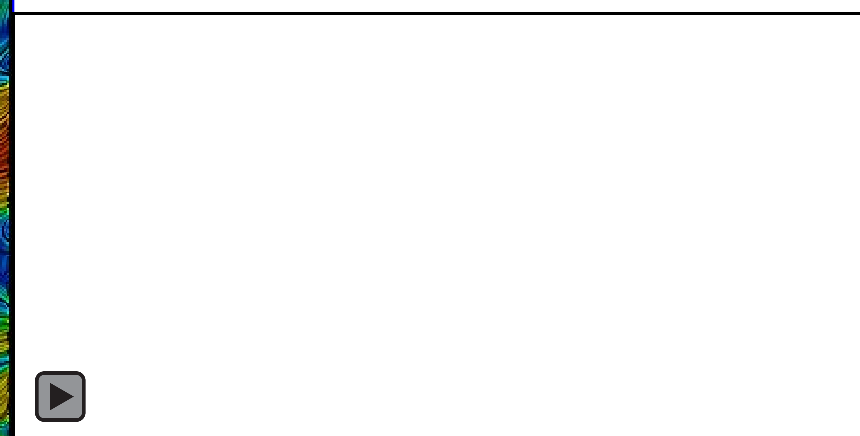
✧ The first texture-based **unsteady** flow visualization method (by *Han-Wei Shen and David Kao, IEEE Visualization 97 & IEEE TVCG 98*).

✧ **Basic idea:** Time-accurately scatters particle values of *successively fed-forward textures* along ***pathlines*** *over several time steps* to convey the footprint / contribution that a particle leaves at downstream locations as the flow runs forward.

✧ **Pro**: High temporal coherence & high spatial coherence & hardware-independent.

✧ **Con**: Low computational performance due to *multi-step* (≈ 100) *pathline integration.*





Overview — Texture-Based Methods

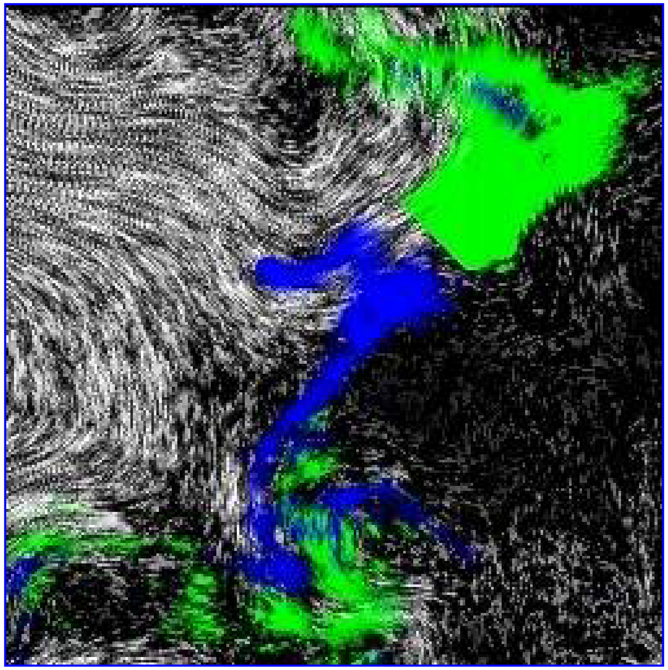
➢ Hardware-Accelerated Texture Advection (HATA)

✧ The **first hardware-based** texture synthesis technique for unsteady flow (by *Bruno Jobard and et al., IEEE Visualization 00*).

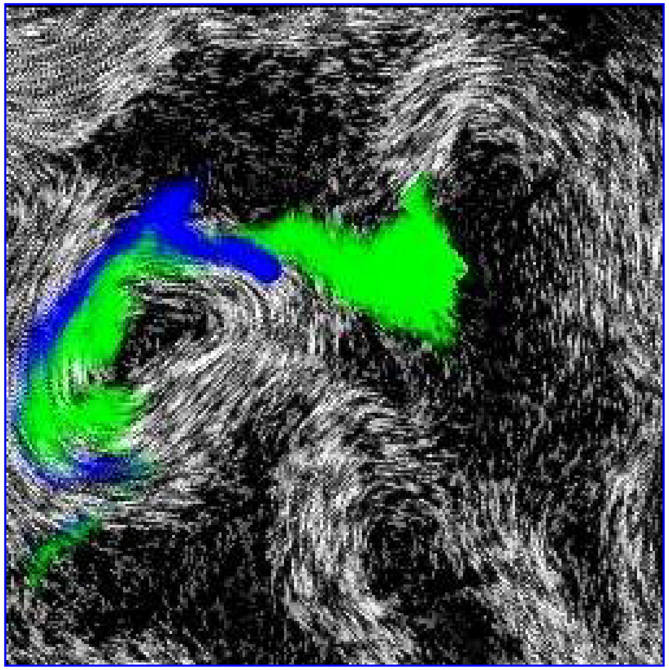
✧ **Basic idea:** Exploits *indirect pixel-texture addressing* for fast flow advection, & additive / subtractive texture blending for fast texture convolution in an efficient pipeline.

✧ **Pro:** Near-interactive frame rates based on special-purpose graphics cards; for both steady and unsteady flow; good temporal coherence .

✧ **Con:** poor spatial coherence (very noisy).



exponential kernel



(Bruno Jobard, Gordon Erlebacher, and M. Yousuff Hussaini)

Overview — Texture-Based Methods

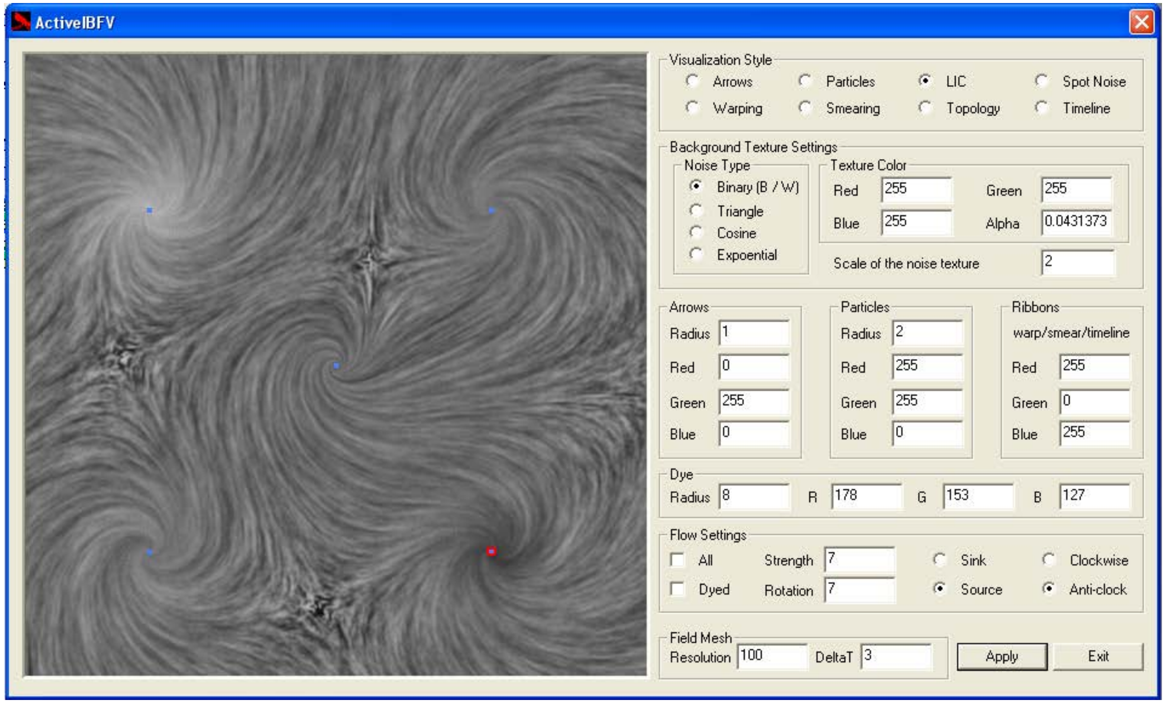
➢ Image-Based Flow Visualization (IBFV)

✧ One of the most versatile and the easiest-to-implement hardware-based methods (by *Jarke J. van Wijk, SIGGRAPH02*).

✧ **Basic idea:** Designs a sequence of *temporally-spatially low-pass filtered* noise textures and cyclically blends them with an iteratively advected (using *forward single-step pathline integration*) image (which is initially a BLACK rectangle).

✧ **Pro:** Interactive frame rates and easy simulation of many visualization techniques; good temporal coherence .

✧ **Con:** insufficient spatial coherence (*noisy or blurred*).



Overview — Texture-Based Methods

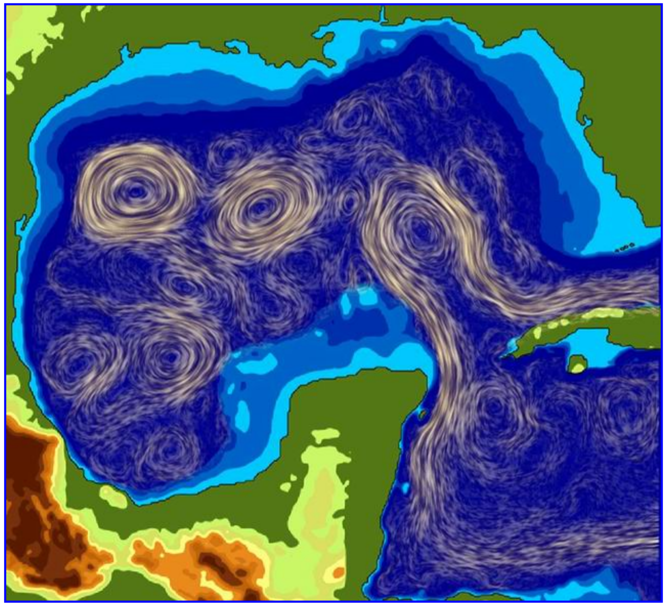
➢ Lagrangian-Eulerian Advection (LEA)

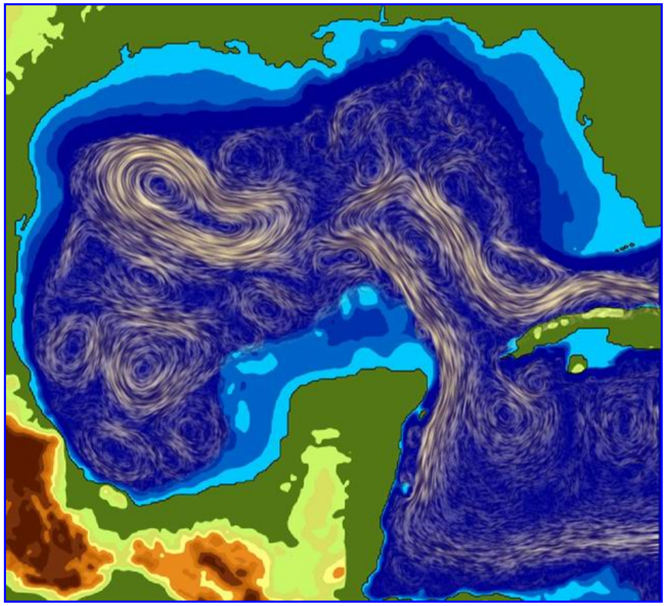
✧ A fast hardware-independent unsteady flow visualization method (by *Bruno Jobard and et al, IEEE TVCG 02*).

✧ **Basic idea:** Employs *backward single-step pathline integration* to search the previous frame for the contributing particle (Eulerian) which scatters the texture value to the target pixel of the current frame (Lagrangian) & blends successive textures.

✧ **Pro:** Interactive frame rates and supportive of arbitrarily-shaped field domains; good temporal coherence .

✧ **Con:** insufficient spatial coherence (obscure direction).





(Bruno Jobard, Gordon Erlebacher, and M. Yousuff Hussaini)

Overview — Texture-Based Methods

➢ Unsteady Flow Advection-Convolution (UFAC)

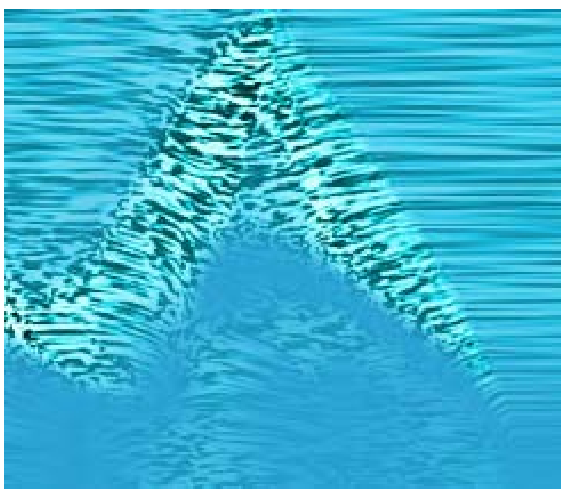
✧ A separable temporal-spatial texture synthesis method for unsteady flow fields (by *Daniel Weiskopf and et al, IEEE Visualization 03*).

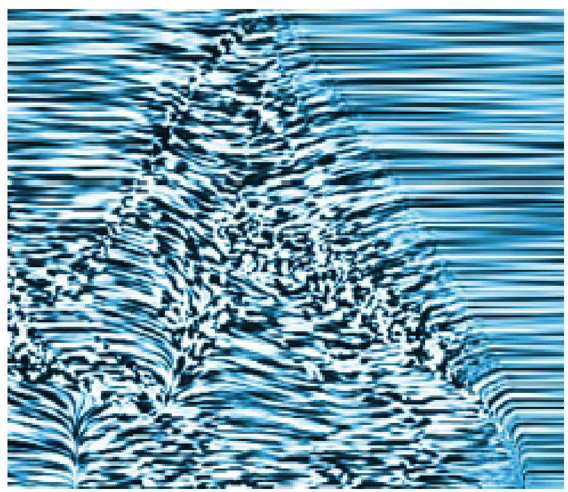
✧ **Basic idea:** Establishes temporal coherence by *property advection along pathlines* while building spatial correlation by *texture convolution along streamlines.*

✧ With explicit, direct, and separate control over temporal coherence and spatial coherence to balance visualization speed and quality.

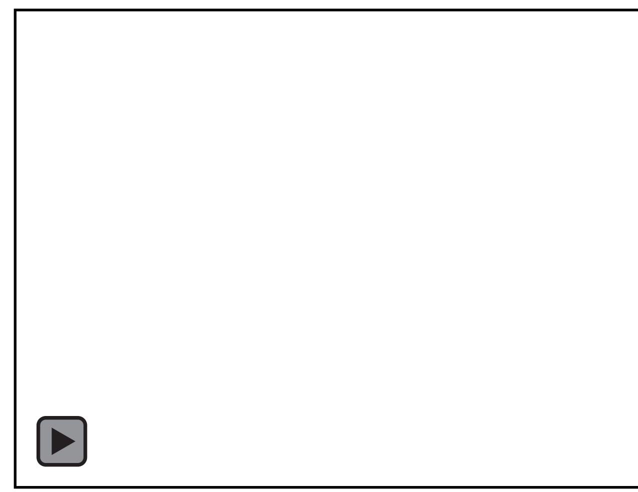
✧ **Pro:** Interactive rates on graphics cards with fragment (e.g., pixel shader) support.

✧ **Con:** Temporal-spatial inconsistency — either flickering animation or noisy image.





Noisy images with (left) / without (right) velocity masking



Good frames in a flickering animation

(Daniel Weiskopf, Gordon Erlebacher, and Thomas Ertl)

Overview — Texture-Based Methods

➢ Steady Flow Visualization Methods

Method	Noise design	Implementation	Image quality	Feature missing	Extensions	Performance
Spot Noise	tricky	tedious	low	yes	few	low
LIC	easy	easy	high	no	many	low

➢ Unsteady Flow Visualization Methods

Method	Temporal coherence	Spatial coherence	Performance	Graphics cards
UFLIC	high	high	low	not required
HATA	good	poor (very noisy)	near-interactive rates	special-purpose
IBFV	good	insufficient (noisy / blurred)	interactive rates	general-purpose
LEA	good	insufficient (obscure direction)	interactive	not required
UFAC	trade-off between noisy image & flickering animation		interactive	special-purpose

Recent Advances

Robert S. Laramee, Helwig Hauser, Helmut Doleisch, Benjamin Vrolijk, Frits H. Post, and Daniel Weiskopf, **The state of the art in flow visualization: dense and texture-Based techniques.** in *Computer Graphics Forum (CGF)*, Vol. 23, No. 2, 2004, pages 203-221.

Guo-Shi Li, Xavier Tricoche, Daniel Weiskopf, and Charles Hansen. Flow Charts: **Visualization of vector fields on arbitrary surface**. IEEE Transactions on Visualization and Computer Graphics, 14(5), pp. 1067-1080, 2008.

Jin Huang, Wenjie Pei, Chunfeng Wen, Guoning Chen, Wei Chen, and Hujun Bao. **Output-coherent image-space LIC for surface flow visualization**. IEEE Pacific Visualization Symposium 2012.

Jin Huang, Zherong Pan, Guoning Chen, Wei Chen, and Hujun Bao. **Image-space texture-based output-coherent surface flow visualization**, IEEE Transactions on Visualization and Computer Graphics, Vol. 19 (9): pp. 1476-1487, 2013.

Victor Matvienko, Jens Krüger. **Dense flow visualization using wave interference**. IEEE Pacific Visualization Symposium 2012.

Victor Matvienko, Jens Krüger. **Explicit frequency control for high-quality *texture*-based flow**, IEEE Visualization 2015.

SOME DETAILS

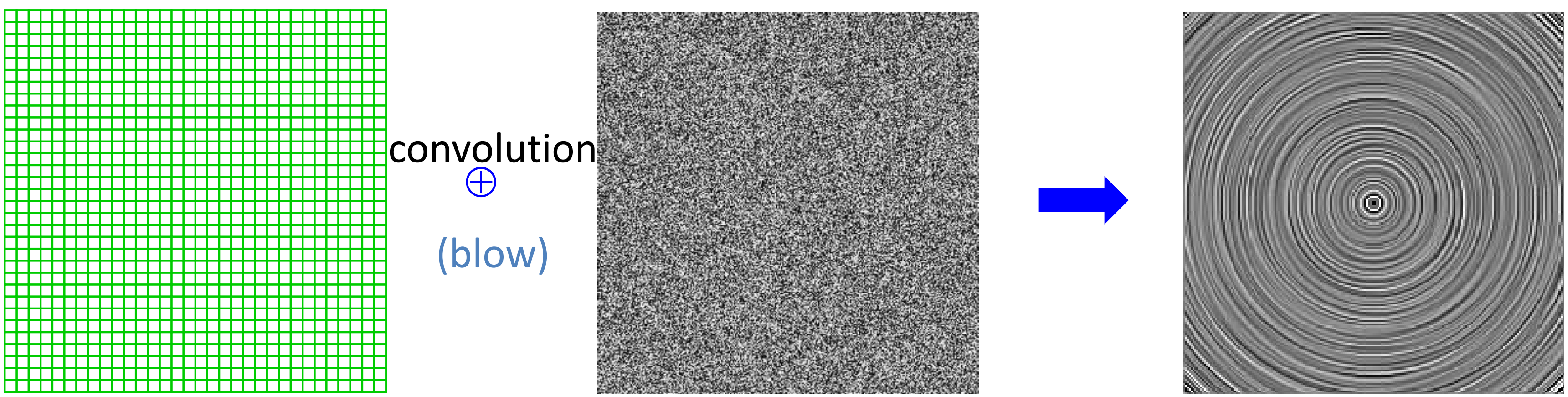
Line Integral Convolution — LIC

Line Integral Convolution (LIC) was presented by *Brian Cabral* and *Casey Leedom* (*ACM SIGGRAPH93*). (cited by 1788 so far)

➢ Basic Idea

✧ LIC convolves *white noise* using a *low-pass filter* along *pixel-centered symmetrically bi-directional streamlines* to exploit spatial correlation in the flow direction — anisotropic low-pass filtering along flow lines.

✧ LIC synthesizes an image that provides *a global dense representation* of the flow, analogous to *the resulting pattern of wind-blown sand*.

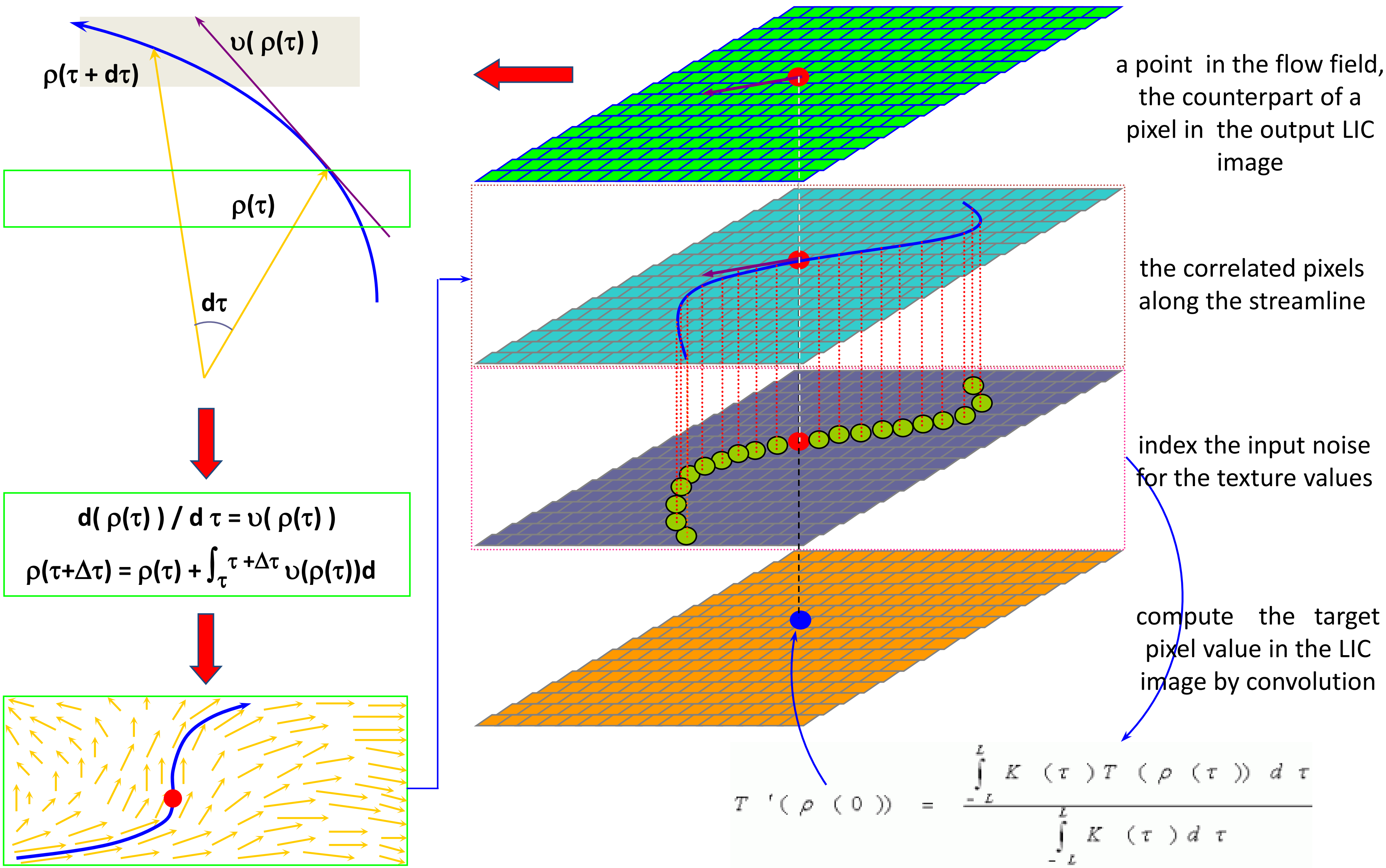


flow field (wind) | white noise (fine sand) | LIC image (pattern)

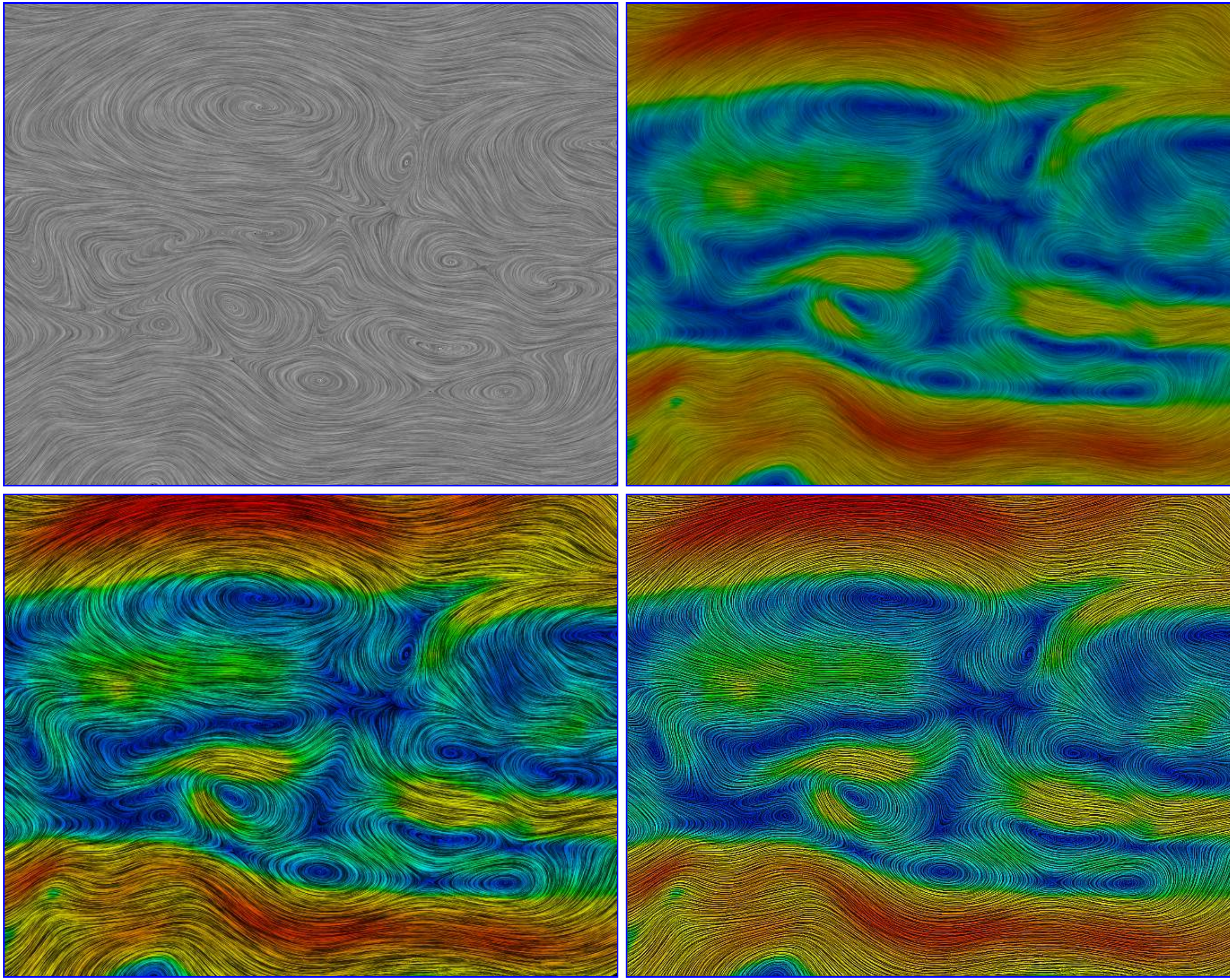
white noise → the texture is freely warped / driven by the flow without any intrinsic resistance

Zhanping Liu @ MSU / HPC / VAIL

Line Integral Convolution — LIC



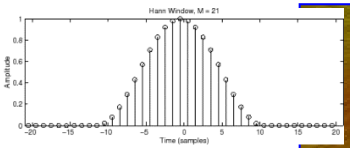
Zhanping Liu @ MSU / HPC / VAIL

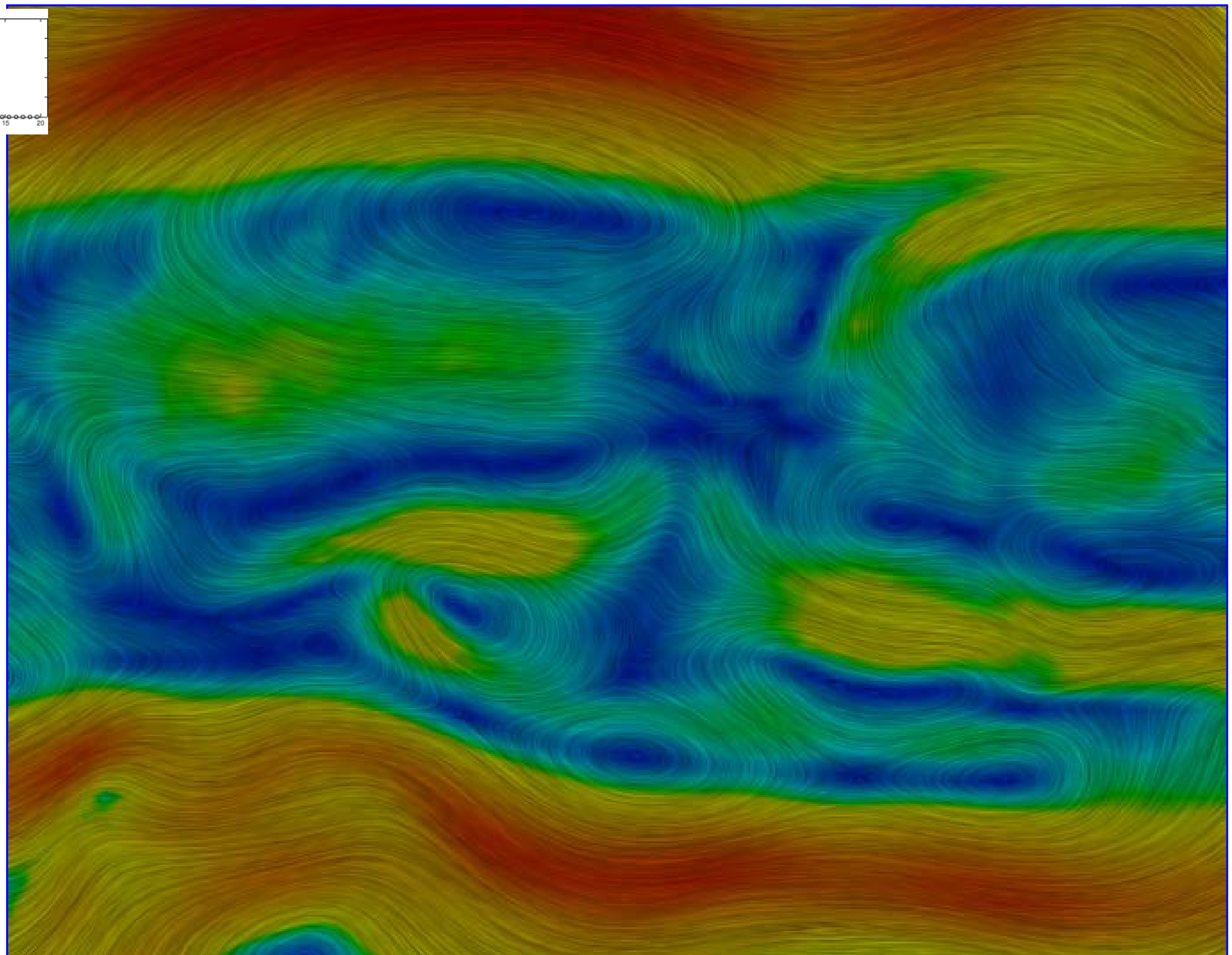


Top-left: gray-scale LIC
Bottom-left: contrasted LIC
Top-right: color-mapped LIC
Bottom-right: high-pass filtered LIC

Line Integral Convolution — LIC variant

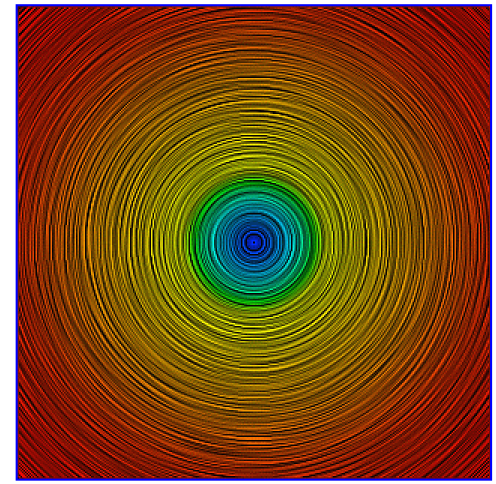
- **Animation** **successively shifting the phase** of <u>a periodic convolution kernel</u> such as Hanning filter ("Motion Without Movement", CG´ 91)



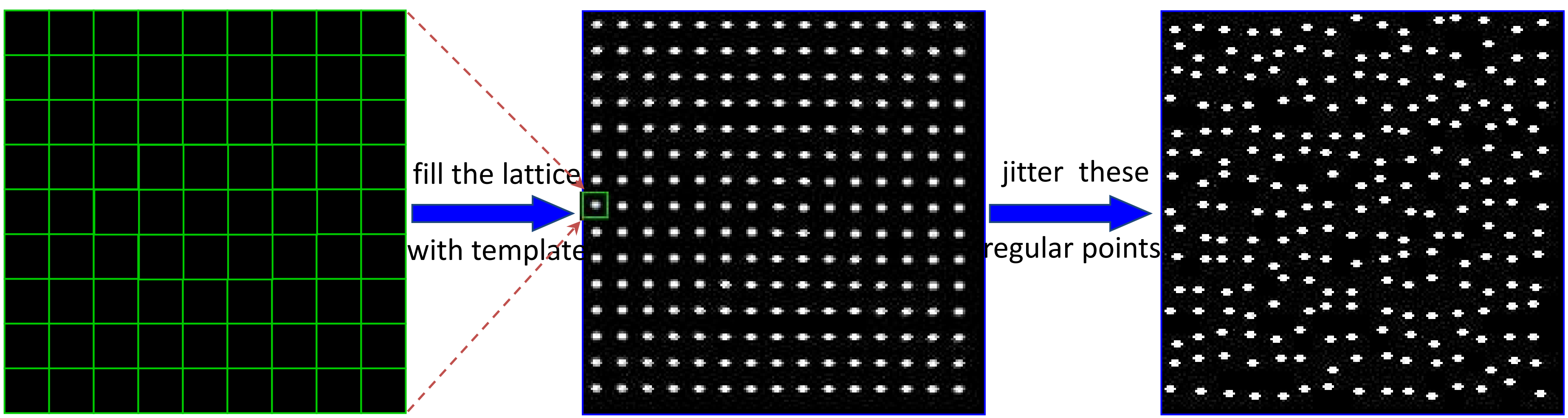


Line Integral Convolution — LIC Variants

➢ OLIC (Oriented LIC)

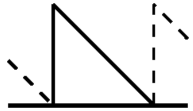


- ✧ R. Wegenkittl and et al. (*Computer Animation 97*).
- ✧ A LIC image shows the flow direction while failing to show the orientation (clockwise or counter-clockwise ?).
- ✧ A ramp filter offers orientation cue by *intensity tapering.*
- ✧ *Sparse noise* offers *enough space for intensity-tapering.*
- ✧ White points of some size are placed at the lattice and then *slightly jittered.*

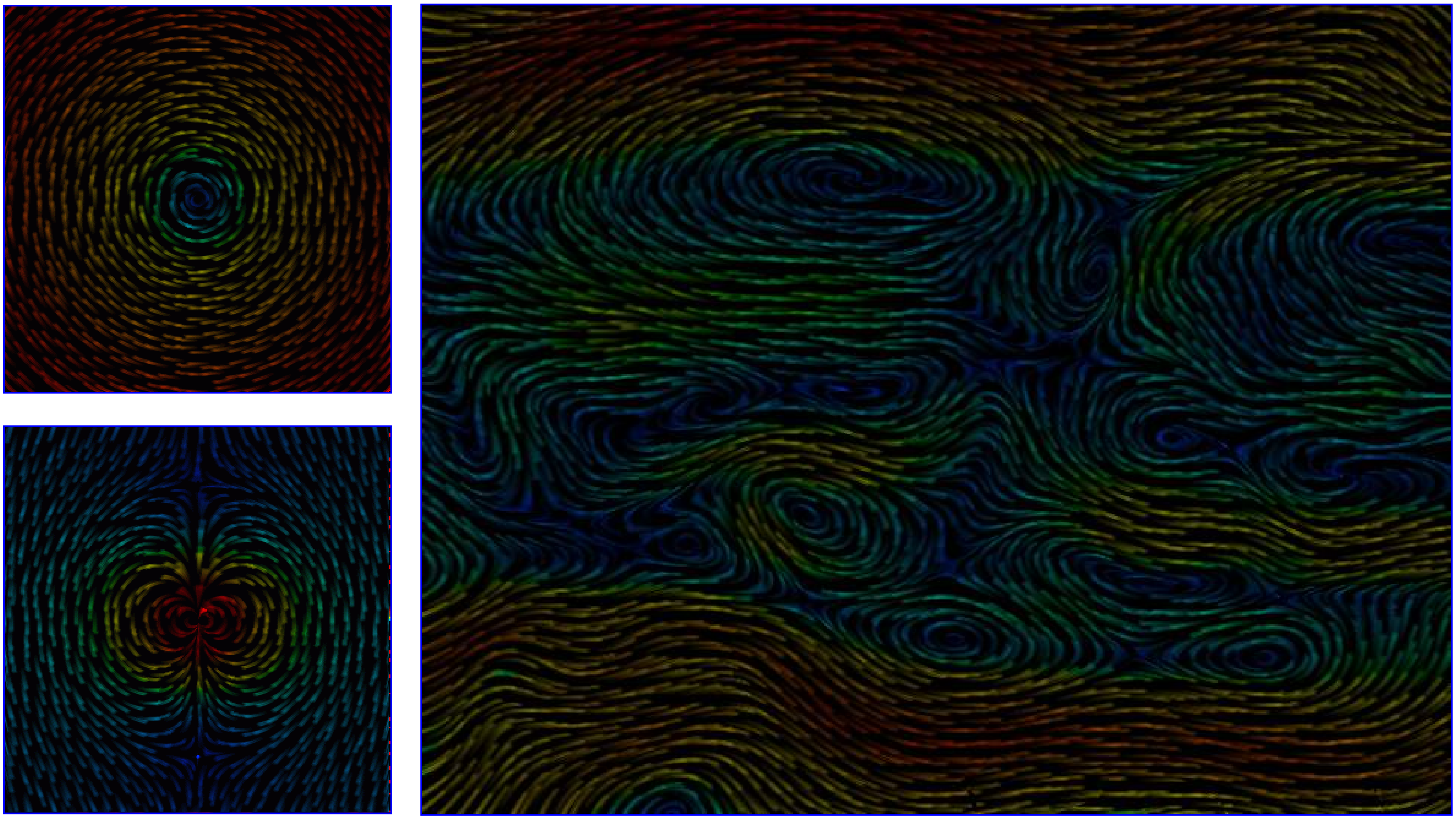


the design of sparse noise

Line Integral Convolution — LIC Variants



Sparse noise ⊕ Ramp convolution kernel ⇨ OLIC (flow orientation in a LIC image)



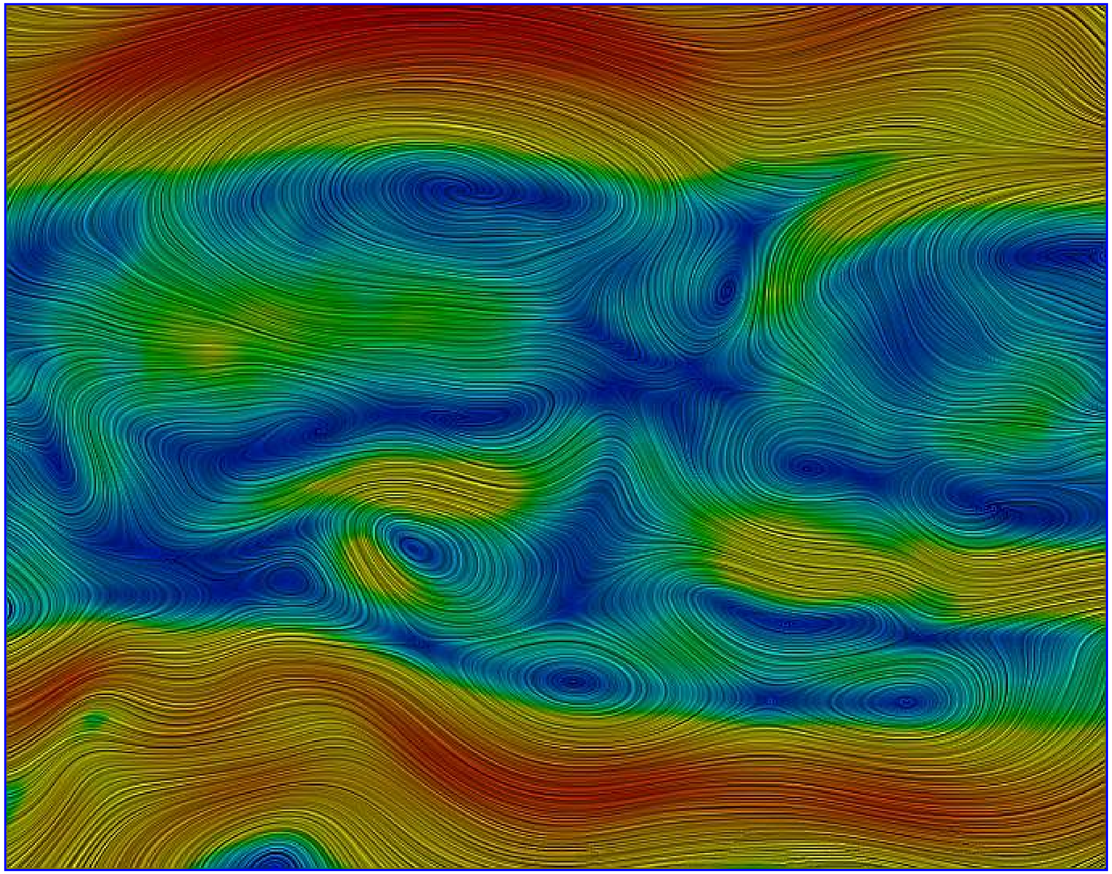
Line Integral Convolution — LIC Variants

➢ Enhanced LIC

✧ A. Okada and D. L. Kao (*IS & T / SPIE Electronics Imaging 97*).

✧ *Enhances the appearance of streamlines* — neither noisy nor blurred.

✧ Iteratively (iteration times >= 2) *takes an output LIC image as the input* to the next LIC cycle prior to *final high-pass filtering* (e.g., Laplacian filter).



Line Integral Convolution — LIC Variants

A quite fancy LIC image results from using *sparse noise in enhanced LIC.*

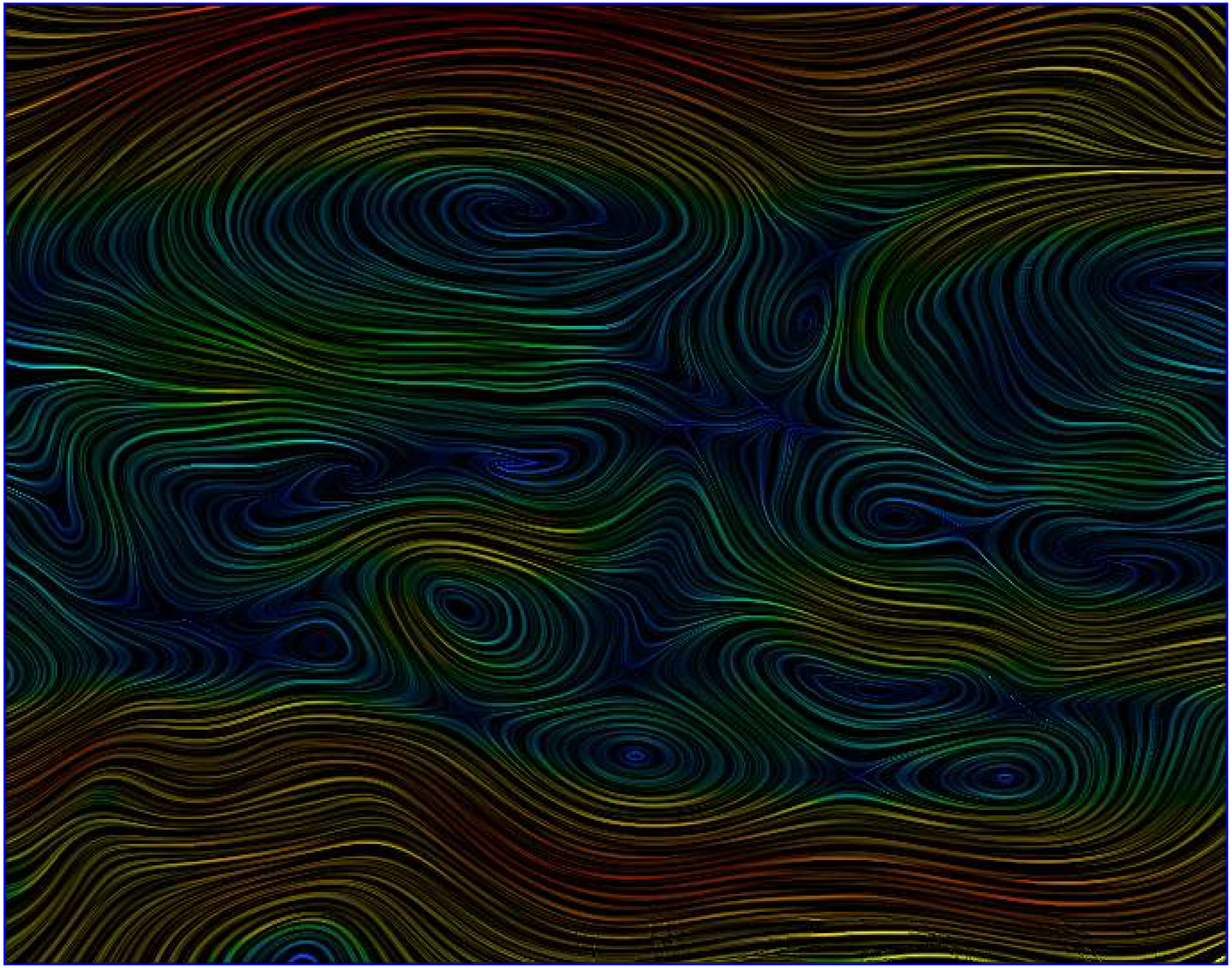
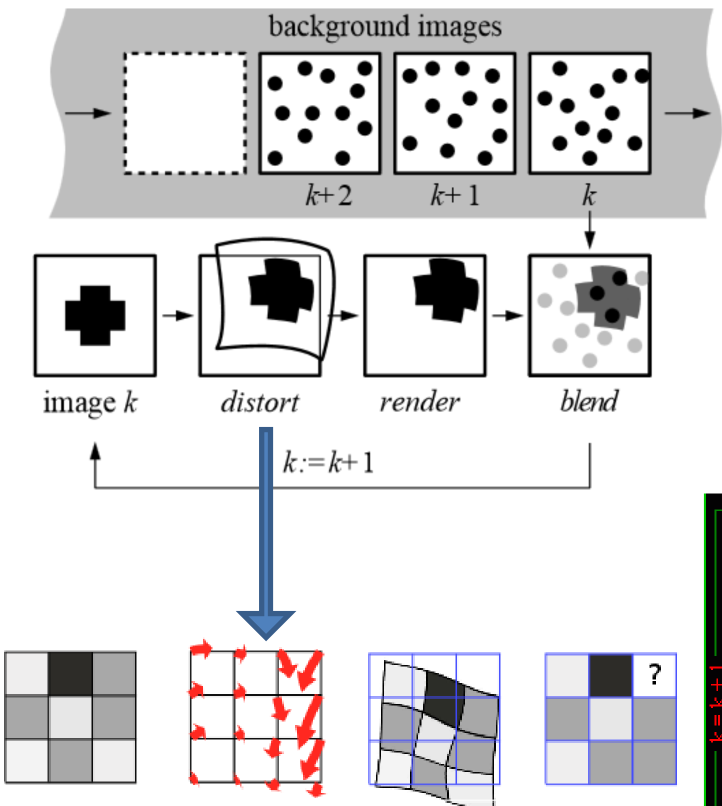


IMAGE-BASED FLOW VISUALIZATION (IBFV)

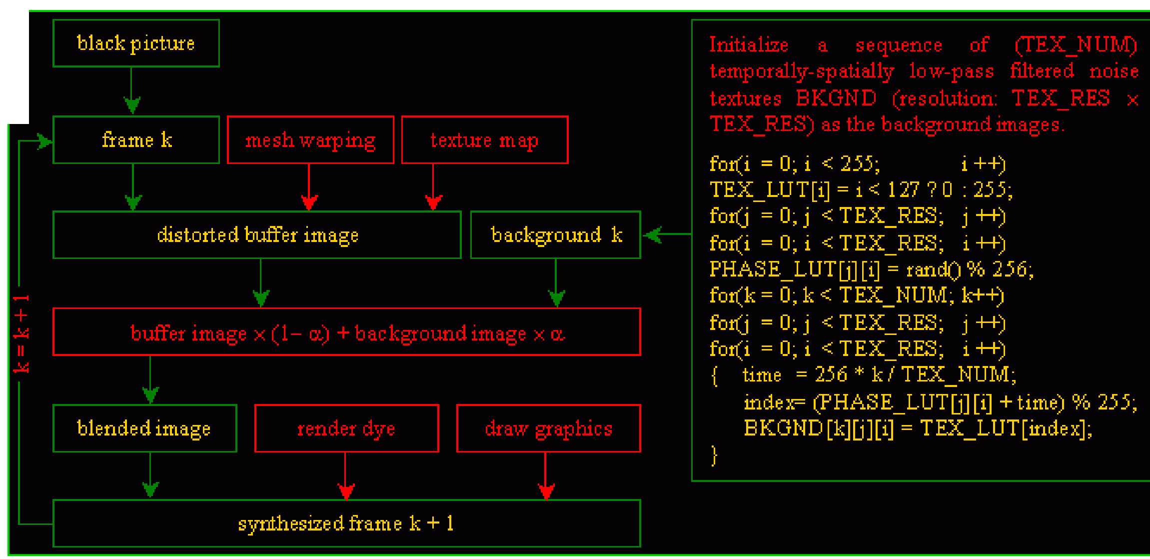
Details — Texture-Based Methods

➢ Image-Based Flow Visualization (IBFV)



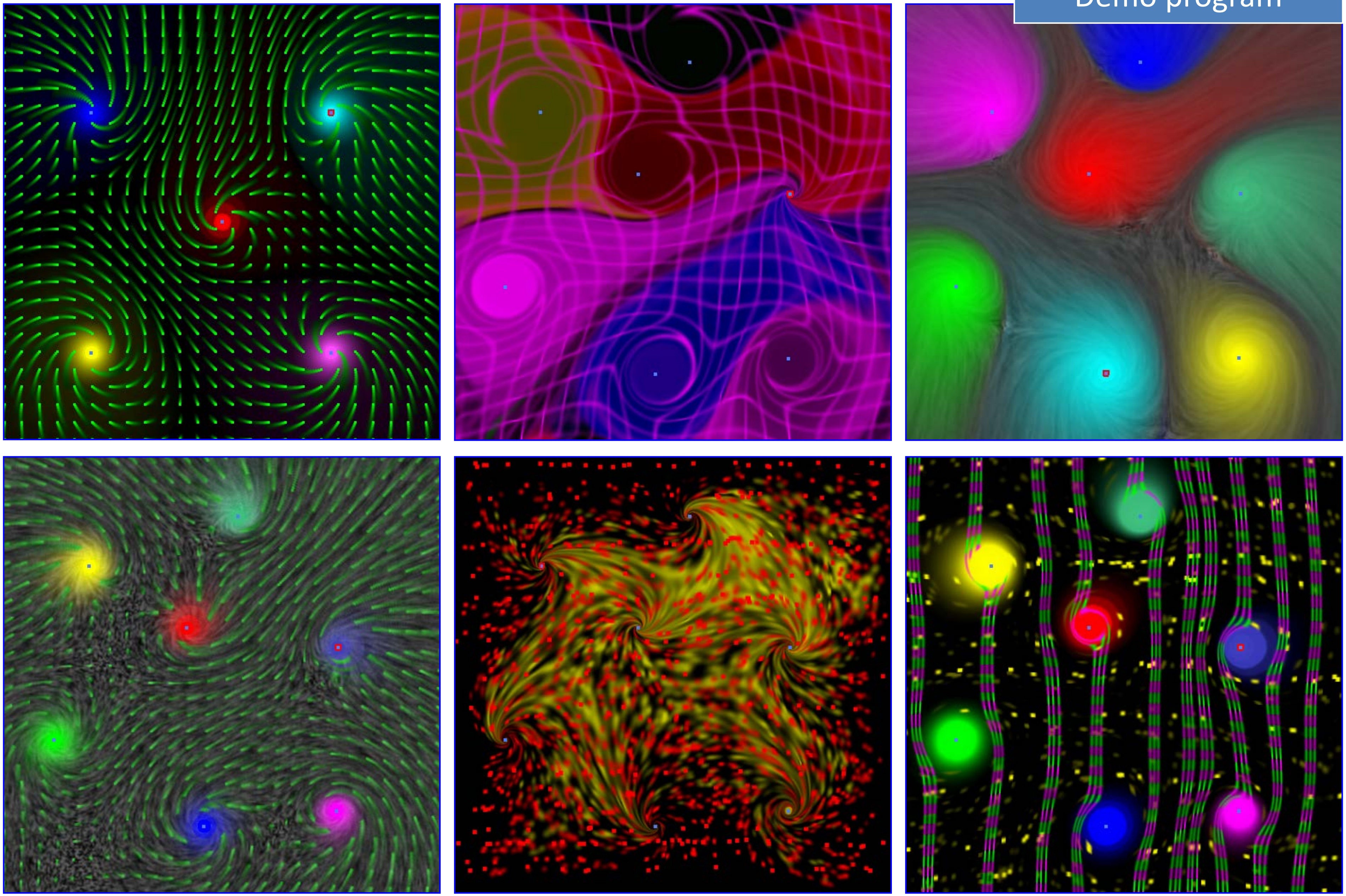
$$F(\mathbf{p}_k; k) = (1-\alpha)F(\mathbf{p}_{k-1}; k-1) + \alpha G(\mathbf{p}_k; k)$$

$$F(\mathbf{p}_k; k) = \alpha \sum_{i=0}^{k-1} (1-\alpha)^i G(\mathbf{p}_{k-i}; k-i)$$



Zhanping Liu @ MSU / HPC / VAIL

Demo program



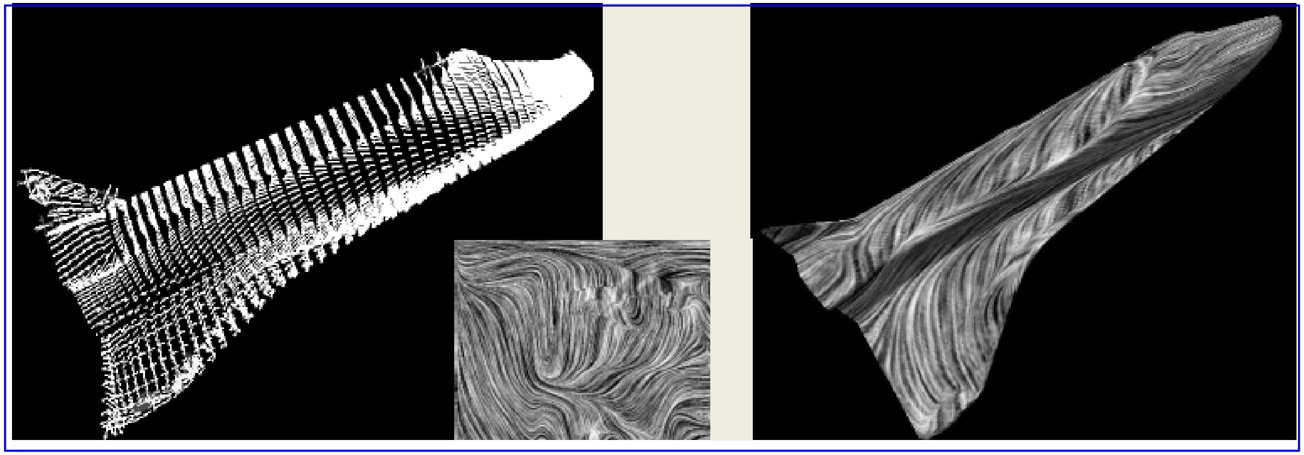
A variety of visualization techniques such as particles, arrow plots, streamlines, timelines, spot noise, LIC, and flow topology can be easily simulated by tuning IBFV parameters

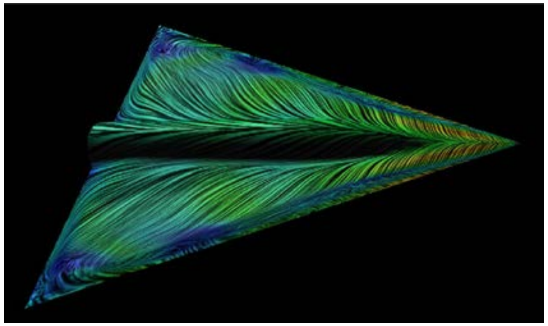
TEXTURE-BASED VISUALIZATION FOR SURFACE FLOW

Line Integral Convolution — LIC Variants

➢ Surface LIC

- ✧ Dense visualization of flows on curved surfaces
- ✧ **Parametric surface LIC — on well-defined surfaces**
  - ◆ On a parameterized CFD surface (model).
  - ◆ On a parameterized stream surface extracted by *Advancing Front* from 3D flows.
  - ◆ *Maps vectors from physical space to parametric space by nonlinear transform.*
  - ◆ *Generates a 2D LIC texture in parametric space.*
  - ◆ *Maps the 2D LIC texture back onto the curved surface (physical space).*
  - ◆ Compensates texture distortions from non-isometric physical-parametric space mapping by using carefully-designed input noise and adaptive kernel length.



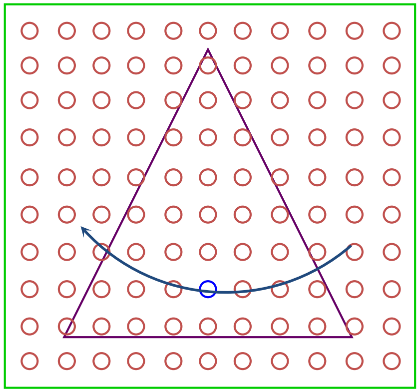


(Lisa Forssell et al., IEEE TVCG 95)

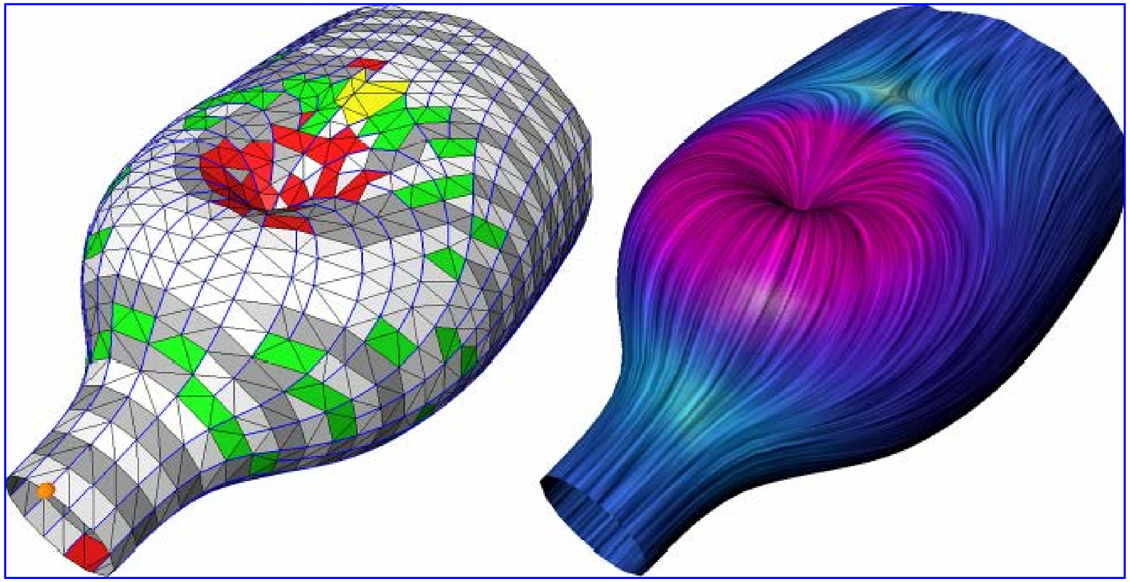
Line Integral Convolution — LIC Variants

✧ Triangulated surface LIC — on arbitrarily complex surfaces

- On extracted iso-surfaces or other implicit surfaces through a volume flow.
- Adopts fast and robust streamline integration directly on a triangular domain.
- Obviates non-isometric space mapping to avoid texture distortions.
- Uses *solid noise* (usually by a procedural noise function).
- Obtains the value of each texel (texture element) sampled in a triangle via LIC.
- Efficiently *packs numerous triangular-textures* into a few rectangular-texture blocks stored in memory for fast texture retrieval at low memory cost.
- Maps each triangular texture onto the target triangle in rendering.



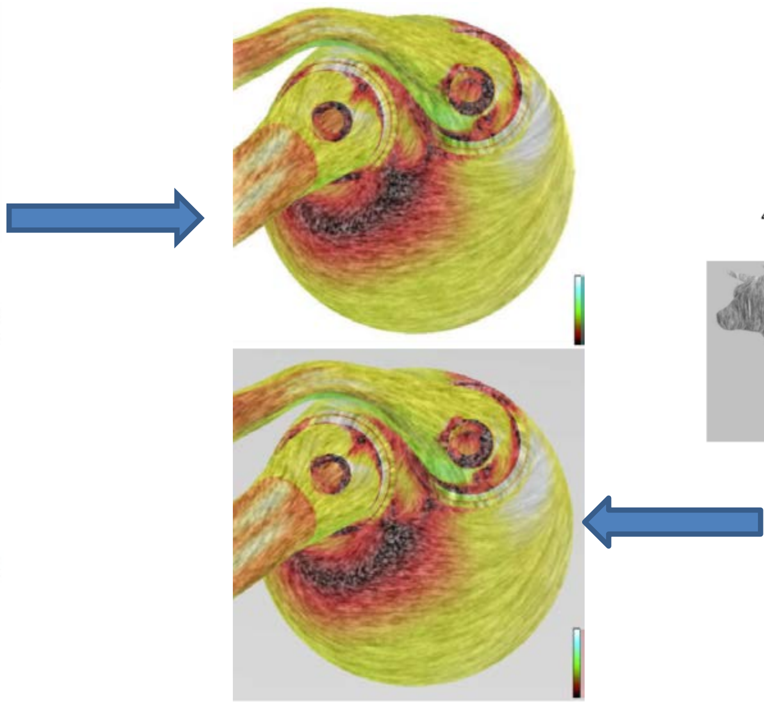
compute each texel value

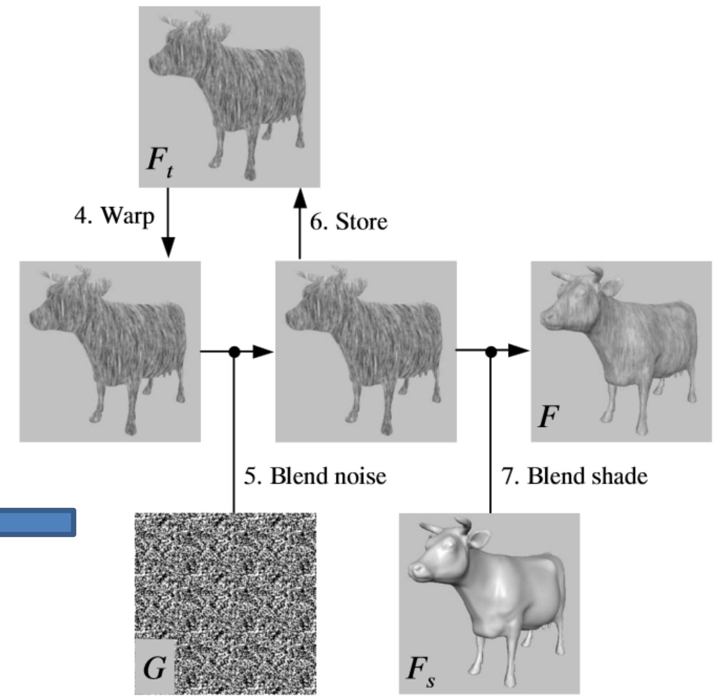


(Detlev Stalling, ZIB, Germany)

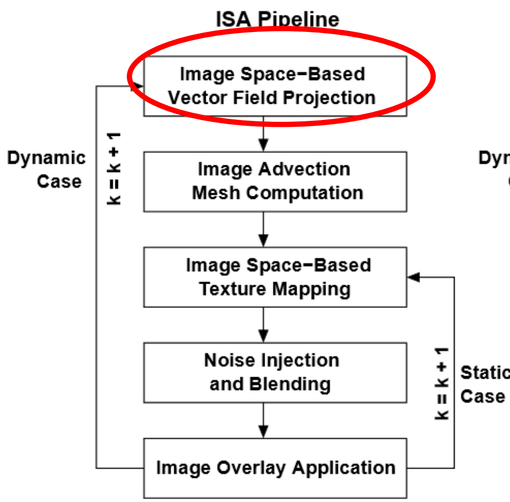
ISA vs. IBFVS

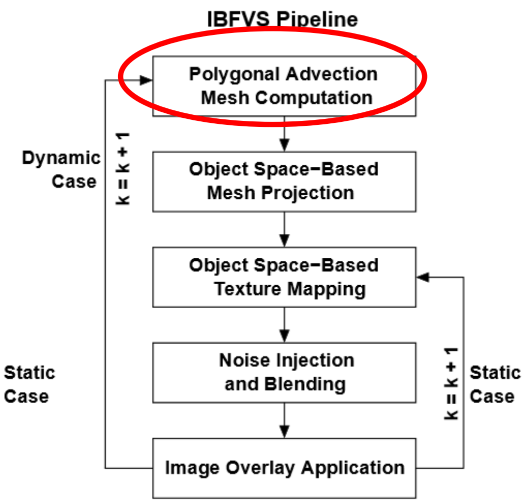
ISA





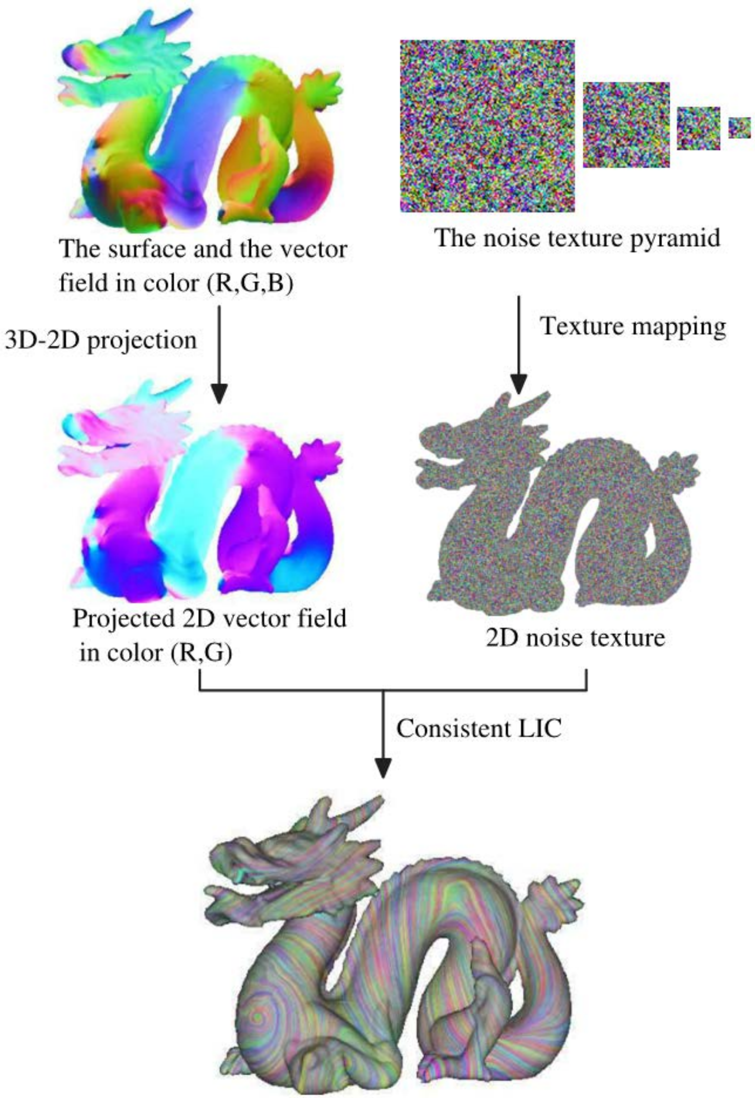
IBFVS

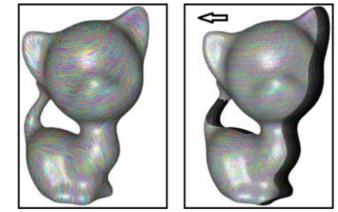




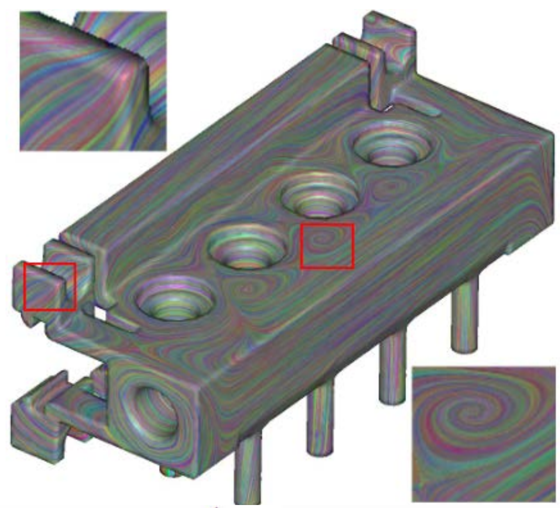
[Laramee et al. TVCG03]

Coherent Texture on Surfaces





Address the inconsistency of flow image when the view point is changed.



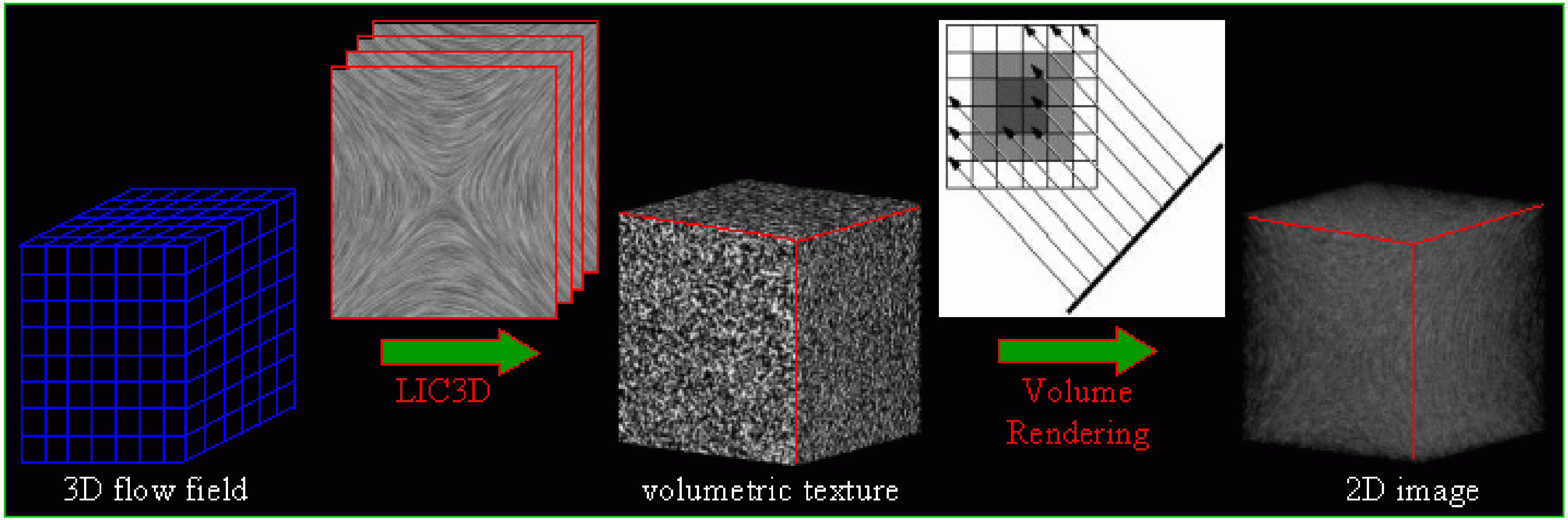
[Huang et al. TVCG13]

TEXTURE-BASED VISUALIZATION FOR 3D FLOW

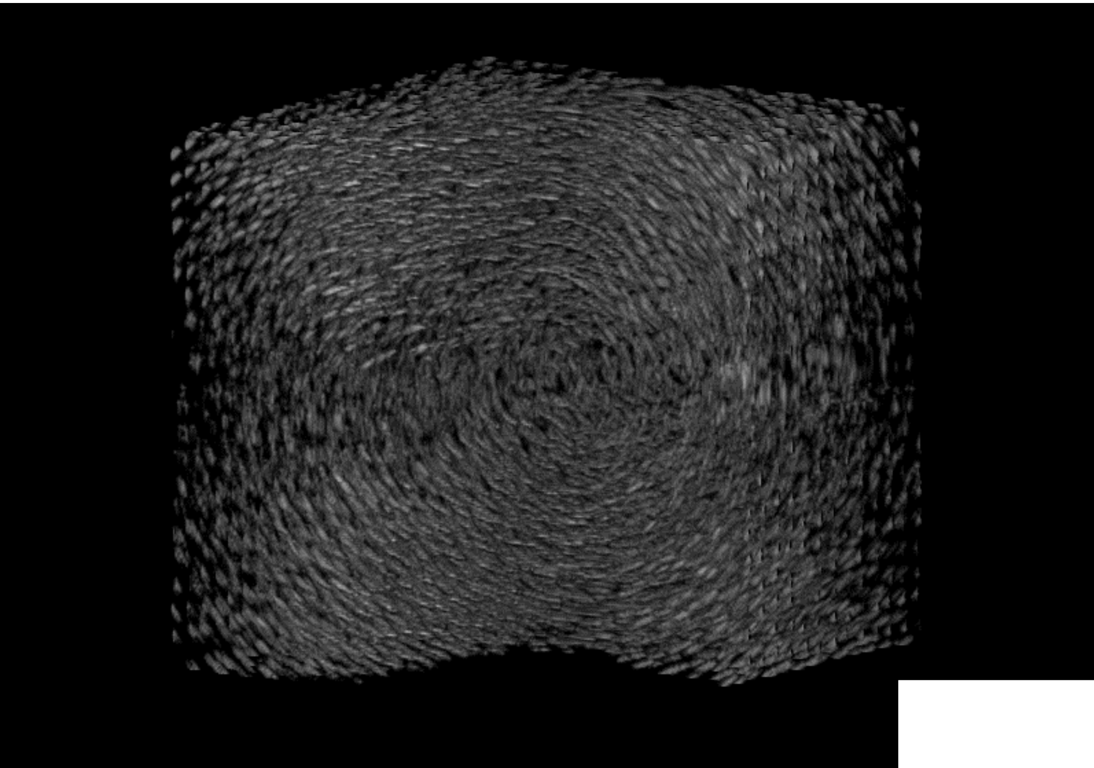
Line Integral Convolution — LIC Variants

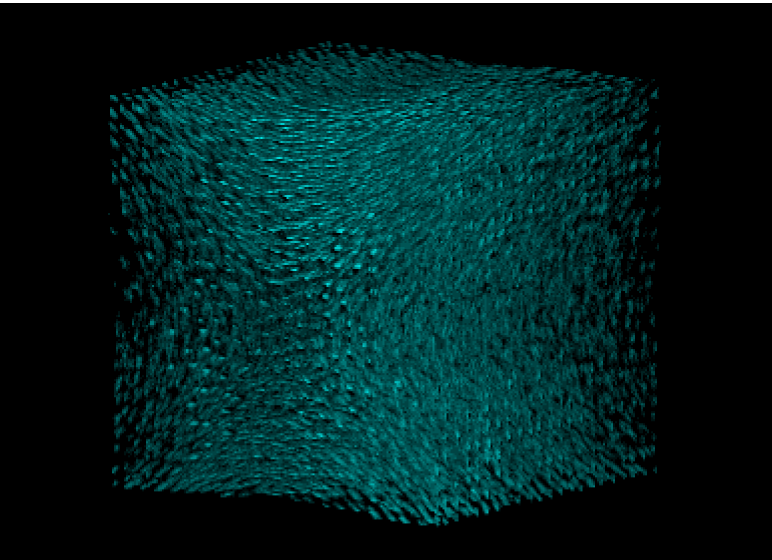
➢ Volume LIC

- ✧ Victoria Interrante and Chester Grosch (*IEEE Visualization 97*).
- ✧ A straightforward extension of LIC to 3D flow fields.
- ✧ Low-pass filters *volumetric noise* along 3D streamlines.
- ✧ Uses *volume rendering* to display resulting 3D LIC textures.
- ✧ Very time-consuming to generate 3D LIC textures.
- ✧ Texture values offer no useful guidance for transfer function design due to *lack of intrinsic physical info* that can be exploited to distinguish components.
- ⇨ Very challenging to clearly show *flow directions and interior structures through a dense texture volume.*



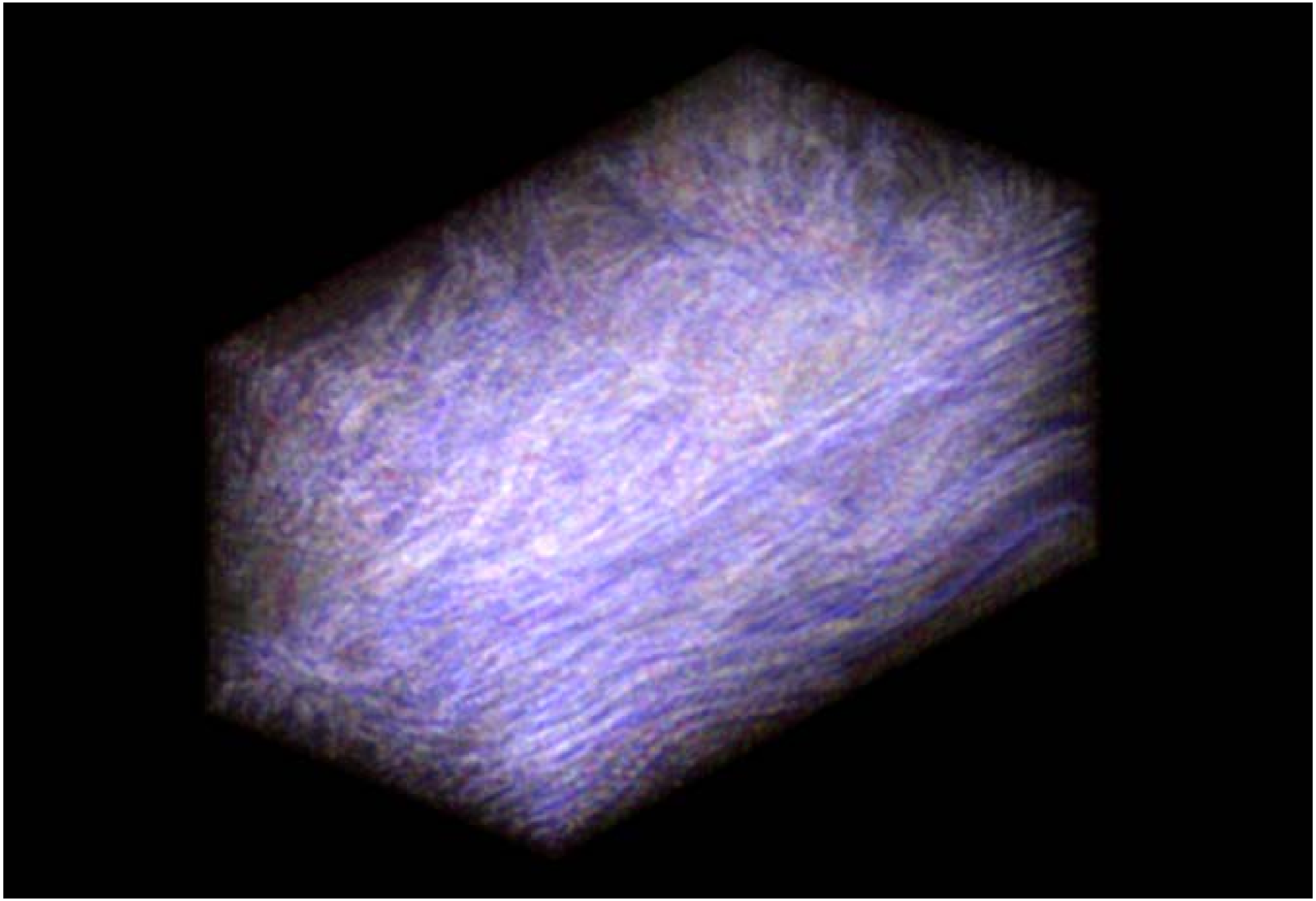
Sparse noise + Hybrid Hanning-Ramp kernel (Zhanping Liu and et al., *Journal of Image and Graphics 2001*)

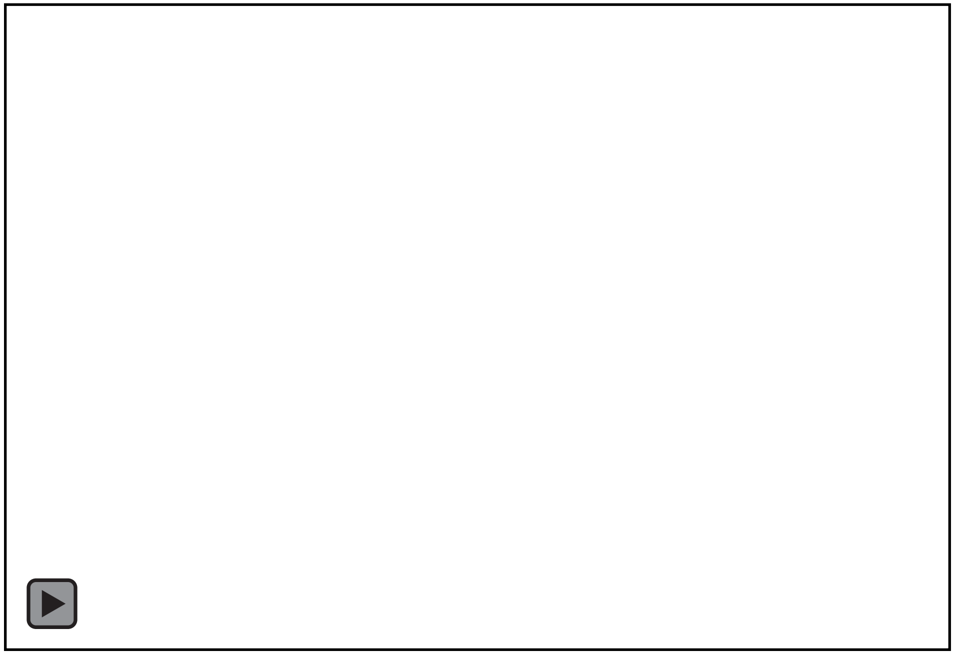


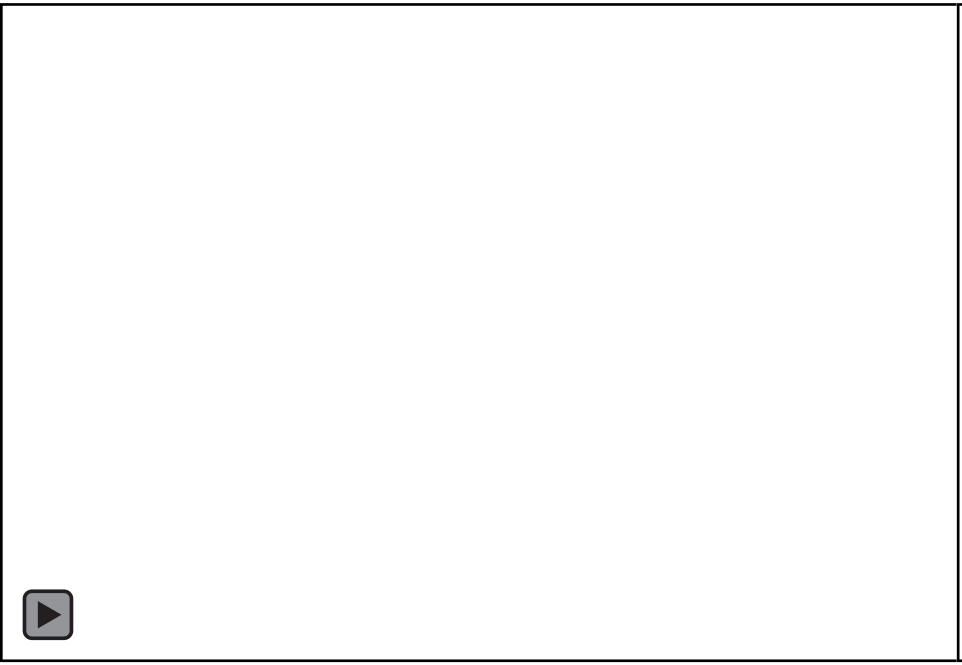


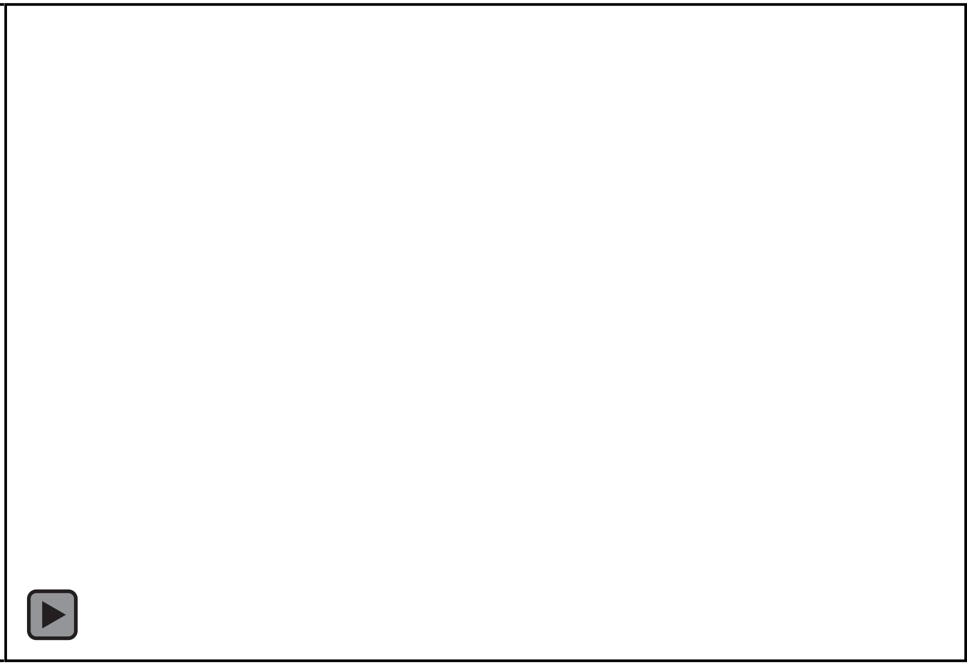
Unsteady Flow LIC — VAUFLIC

Image generated by using a texture-based transfer function

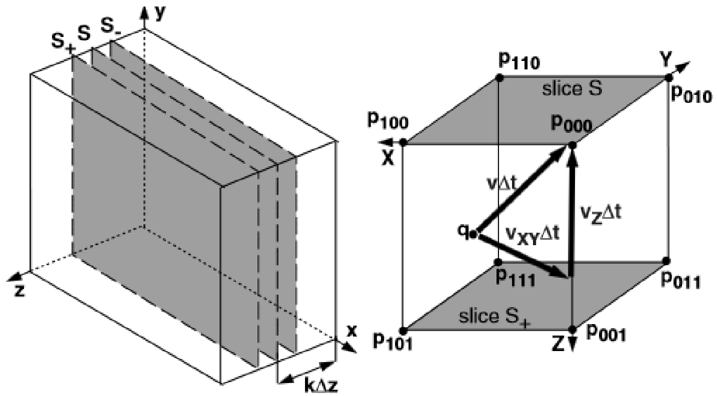




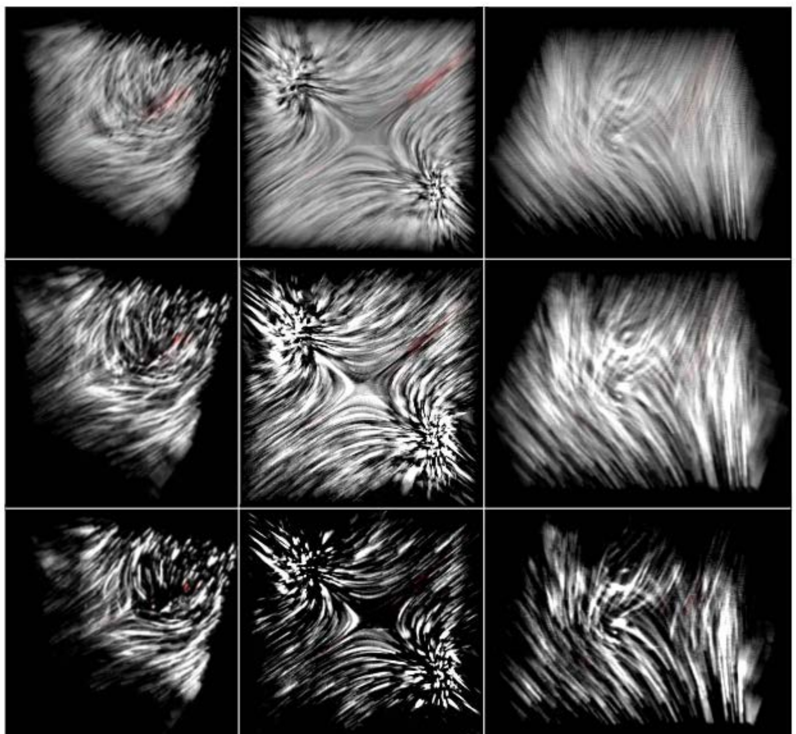


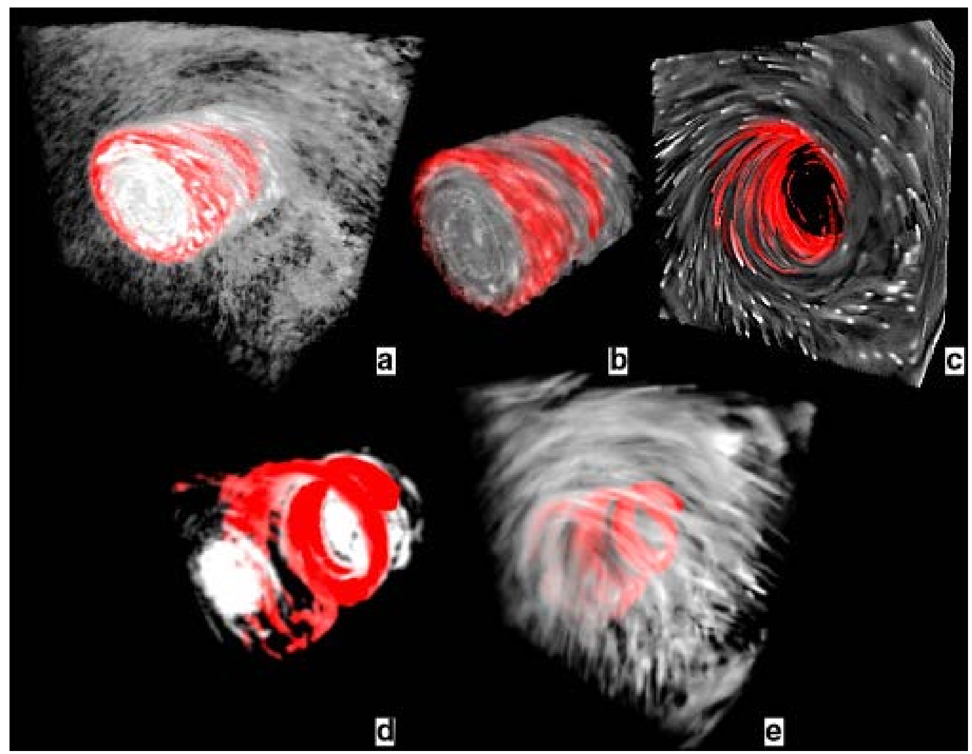


3D IBFV



for i = 0 to N-1
{
    if (i>0) **(1)**
        do 1D Z-axis advection from S_{i-1} to S_i
    if (i<N-1) **(2)**
        do 1D Z-axis advection from S_{i+1} to S_i
    do 2D IBFV-based advection in the slice S_i **(3)**
}





[Telea and van Wijk Vis03]

Additional reading

- Helwig Hauser, Robert S. Laramee, Helmut Doleisch, Frits H. Post, and Benjamin Vrolijk, **The State of the Art in Flow Visualization: Direct, Texture-based, and Geometric Techniques**, *TR-VRVis-2002-046* Technical Report, VRVis Research Center, Vienna, Austria, December 2002.

- Robert S. Laramee, Helwig Hauser, Helmut Doleisch, Benjamin Vrolijk, Frits H. Post, and Daniel Weiskopf, **The State of the Art in Flow Visualization: Dense and Texture-Based Techniques.** in Computer Graphics Forum (CGF), Vol. 23, No. 2, 2004, pages 203-221.

Acknowledgment

- Thanks for the materials from
  - Dr. Zhanping Liu, Kentucky State University
  - Dr. Robert Laramee, Swansea University, UK