Arrows vs. Streamlines vs. Textures

Streamlines: selective

Arrows: simple
Arrows vs. Streamlines vs. Textures

Streamlines: selective
Arrows: simple

Textures: 2D-filling
Provide full spatial coverage
Vector Field Visualization: Texture-based Method
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)

- 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 vis (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).

(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).
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

(Daniel Weiskopf, Gordon Erlebacher, and Thomas Ertl)
### Overview — Texture-Based Methods

#### Steady Flow Visualization Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Noise design</th>
<th>Implementation</th>
<th>Image quality</th>
<th>Feature missing</th>
<th>Extensions</th>
<th>Performance</th>
<th>Graphics cards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot Noise</td>
<td>tricky</td>
<td>tedious</td>
<td>low</td>
<td>yes</td>
<td>few</td>
<td>low</td>
<td></td>
</tr>
<tr>
<td>LIC</td>
<td>easy</td>
<td>easy</td>
<td>high</td>
<td>no</td>
<td>many</td>
<td>low</td>
<td></td>
</tr>
</tbody>
</table>

#### Unsteady Flow Visualization Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Temporal coherence</th>
<th>Spatial coherence</th>
<th>Performance</th>
<th>Graphics cards</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFLIC</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>not required</td>
</tr>
<tr>
<td>HATA</td>
<td>good</td>
<td>poor (very noisy)</td>
<td>near-interactive rates</td>
<td>special-purpose</td>
</tr>
<tr>
<td>IBFV</td>
<td>good</td>
<td>insufficient (noisy / blurred)</td>
<td>interactive rates</td>
<td>general-purpose</td>
</tr>
<tr>
<td>LEA</td>
<td>good</td>
<td>insufficient (obscure direction)</td>
<td>interactive</td>
<td>not required</td>
</tr>
<tr>
<td>UFAC</td>
<td>trade-off between noisy image &amp; flickering animation</td>
<td>interactive</td>
<td>special-purpose</td>
<td></td>
</tr>
</tbody>
</table>
Recent Advances


SOME DETAILS
Line Integral Convolution — LIC

Line Integral Convolution (LIC) was presented by Brian Cabral and Casey Leedom (ACM SIGGRAPH93). (cited by 1683 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.

white noise ➔ the texture is freely warped / driven by the flow without any intrinsic resistance
A point in the flow field, the counterpart of a pixel in the output LIC image.

The correlated pixels along the streamline index the input noise for the texture values.

The target pixel value in the LIC image is computed by convolution.

\[ \rho(\tau + d\tau) = \rho(\tau) + \int_{\tau}^{\tau + d\tau} \nu(\rho(\tau)) \, d\tau \]

\[ d(\rho(\tau)) / d\tau = \nu(\rho(\tau)) \]

\[ T'((\rho(0))) = \frac{\int_{-L}^{L} K(\tau) T(\rho(\tau)) \, d\tau}{\int_{-L}^{L} K(\tau) \, d\tau} \]
Top-left: gray-scale LIC  Top-right: color-mapped LIC
Bottom-left: contrasted LIC  Bottom-right: high-pass filtered LIC
Animation successively shifting the phase of a periodic convolution kernel 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.*

![Diagram of LIC process](image)
Line Integral Convolution — LIC Variants

Sparse noise \oplus\text{ Ramp convolution kernel} \iff \text{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 $\geq 2$) **takes an output LIC image as the input** to the next LIC cycle prior to **final high-pass filtering** (e.g., Laplacian filter).
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)

\[
F(p_k; k) = (1 - \alpha) F(p_{k-1}; k - 1) + \alpha G(p_k; k)
\]

\[
F(p_k; k) = \alpha \sum_{i=0}^{k-1} (1 - \alpha)^i G(p_{k-i}; k - i)
\]

Initialize a sequence of \(TEX\_NUM\) temporally-spatially low-pass filtered noise textures \(BKOND\) (resolution: \(TEX\_RES \times TEX\_RES\)) as the background images.

```java
for(i = 0; i < 255; i++)
    TEX_LUT[i] = i < 127 ? 0 : 255;
for(j = 0; j < TEX_RES; j++)
    for(i = 0; i < TEX_RES; i++)
        PHASE_LUT[j][i] = rand() % 256;
for(k = 0; k < TEX_NUM; k++)
    for(j = 0; j < TEX_RES; j++)
        for(i = 0; i < TEX_RES; i++)
            \{ time = 256 * k / TEX_NUM;
                index = (PHASE_LUT[j][i] + time) % 255;
                BKOND[k][j][i] = TEX_LUT[index];
            \}
```
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.

(Detlev Stalling, ZIB, Germany)
ISA vs. IBFVS

ISA Pipeline
- Image Space-Based Vector Field Projection
- Image Advection Mesh Computation
- Image Space-Based Texture Mapping
- Noise Injection and Blending
- Image Overlay Application

Dynamic Case: $k = k + 1$

IBFVS Pipeline
- Polygonal Advection Mesh Computation
- Object Space-Based Mesh Projection
- Object Space-Based Texture Mapping
- Noise Injection and Blending
- Image Overlay Application

Dynamic Case: $k = k + 1$

[Laramee et al. TVCG03]
Coherent Texture on Surfaces

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

The surface and the vector field in color (R,G,B)

3D-2D projection

Projected 2D vector field in color (R,G)

Texture mapping

2D noise texture

The noise texture pyramid

Consistent LIC

[Huang et al. TVCG13]
TEXTURE-BASED VISUALIZATION FOR 3D FLOW
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 $N-1$

\{(1)\}

\text{if } (i>0) \text{ do 1D Z-axis advection from } S_{i-1} \text{ to } S_i \text{ (2)}

\text{if } (i<N-1) \text{ do 1D Z-axis advection from } S_{i+1} \text{ to } S_i \text{ (3)}

\text{do 2D IBFV-based advection in the slice } S_i\}
Acknowledgment

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