Visualisation of complex flows using texture-based techniques

D. J. Warne[1]   J. Young[2]   N. A. Kelson[3]

Abstract

Detailed representations of complex flow datasets are often difficult to generate using traditional vector visualisation techniques such as arrow plots and streamlines. This is particularly true when the flow regime changes in time. Texture based techniques, which are based on the advection of dense textures, are novel techniques for visualising such flows. We review two popular texture based techniques and their application to flow datasets sourced from active research projects. The techniques investigated were *Line integral convolution* [Cabral and Leedom, *SIGGRAPH'93*, pp.263–270, 1993], and *Image based flow visualisation* [van Wijk, *SIGGRAPH'02*, pp.745–754, 2002]. We evaluate these and report on their effectiveness from a visualisation perspective. We also report on their ease of implementation and computational overheads.

Contents

1 Introduction

Scientific visualisation plays a crucial role in the human comprehension of scientific data [4]. It is not only vital for the communication of findings, but can be indispensable for data exploration and model validation [20].

The goal of effective visualisation is to represent complex phenomena in a way that allows the observer to extract meaning from the underlying data [16, 15]. The visualisation technique employed requires careful consideration to ensure the data is faithfully represented and the observer is not misled [6, 19]. Therefore, it is crucial for scientists to be aware of modern developments in scientific visualisation which aim to enhance understanding through improved visual representations.

Complex flows present challenges for effective visualisation. In most cases, traditional techniques such as arrow plots and streamlines are difficult to use effectively; particularly when time dependent features are of major interest.

Texture based vector field visualisation methods are novel techniques based on the advection of textures (also referred to as images) by the underlying field. These techniques can be exceedingly effective at representing spatio-temporal flow information even in complex flows [9].

Texture based techniques are inaccessible to many researchers [19]. Our hope is that this article provides practical insight into the application of these techniques and hence lead to enhanced scientific understanding and research outcomes.

Section 2 provides an overview of vector field visualisation, and highlights common problems encountered when these techniques are applied to complex flows. Section 2 also describes the three research projects and the datasets used for evaluating the texture based techniques. Section 3 outlines the mathematical details of LIC and IBFV. Section 4 discusses the effectiveness of the visualisations and the contributions they make to the research outcomes. Performance and implementation considerations of the LIC and IBFV algorithms are also discussed. Section 5 concludes.

2 Background

2.1 Vector field visualisation

2.1.1 Overview

Vector field visualisation is a subfield of scientific data visualisation. It is dedicated to the visualisation of flow fields and thus has many practical applications [8]. Most readers would be familiar with traditional arrow plots or streamlines to visualise flows. Perhaps fewer would realise that these are just a subset of a rich collection of available techniques. Four major classes of vector visualisation techniques exist [14, 9].

- *Direct techniques* represent the data using discrete glyphs and graphical objects and thus require a minimal amount of pre-processing. The most common approach is to draw an arrow indicating velocity at each data point.

- *Texture based techniques* produce dense, image based representations and involve the filtering of a texture (typically a noise image) by the vector field. A filter is chosen that will visually correlate pixels that lie on the same trajectory; hence representing the flow in a 'continuous' manner.

- *Geometric techniques* employ geometric objects to represent flow features. Curves or surfaces represent particle traces (e.g., streamlines) or regions where some flow property is constant (e.g., velocity contours).

- *Feature based techniques* use derived data from the flow field to produce visualisations. The flow is pre-processed to extract features that represent something of interest to the researcher (e.g., the location of critical points). The derived data is then visualised by a technique appropriate for the resulting data type. Feature based techniques are often applied as a data reduction process.

2.1.2 Problems for complex flows

Traditional vector visualisation techniques fall into two categories: direct and geometric. Issues can arise with these techniques when the flow of interest involves complex dynamics.

Direct techniques produce discrete representations (e.g., arrow plot) and the observer must mentally interpolate graphical icons to gain an understanding of the flow. This can lead to perceptual ambiguities when the vector field varies significantly over a cell [15]. As the number of points increases, direct techniques also tend to produce visual clutter [17].

Geometric techniques such as streamlines and pathlines are ideal for depicting single particle traces. However, optimal strategies for seeding large numbers of particle traces to visualise flow fields across complete domains is non-trivial [13, 8]. As a result, geometric techniques may be ineffective as exploration tools, since important flow features (e.g., critical points) may be overlooked.

2.2 Summary of research project case studies

We have collaborated with a number of research groups where texture based flow visualisation techniques had the potential to solve a range of flow visualisation problems that were being encountered. A brief summary of each project is provided below for context and background.

2.2.1 Analysis of gross pollutant trap flow

Gross pollutant traps (GPTs) are stormwater quality improvement devices designed to trap pollutants dimensionally greater than $5\,\mathrm{mm}$. A method to analyse flow regimes through an experimental GPT was developed by Madhani et al. [12]. Flow datasets were collected using a high speed camera and particle image velocimetry software [12, 11].

2.2.2 Simulation of heterogeneous porous media flow

A numerical method to solve variable density transport problems in heterogeneous porous media was developed by Cumming et al. [2]. One problem of interest was the so-called Eldar problem, where initially a layer of dense salt water sits directly on top of a fresh water aquifer. The salt water then sinks into the fresh water aquifer, creating complex up-welling and down-welling flows [3].

2.2.3 Interpretation of a hydrodynamic model

The Pumicestone Passage is a barrier lagoon/estuary located in southeast Queensland, Australia. It is shallow with extensive tidal and mangrove areas. A hydrodynamic model was developed to gain an understanding of the advection of solutes (e.g., contaminates) through the passage [10].

3 Methods

3.1 Texture based techniques

As described in Section 2.1.1, texture based methods are a class of vector visualisation algorithms based on filtering images by an underlying flow field [4, 20]. While a range of techniques exists in this class (Laramee et al. [9] gave a detailed list), here we consider line integral convolution and image based flow visualisation.

3.1.1 Line integral convolution

Line integral convolution (LIC) [1] is a texture based technique designed for steady vector fields $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$. It produces an image I$'$ where the intensity
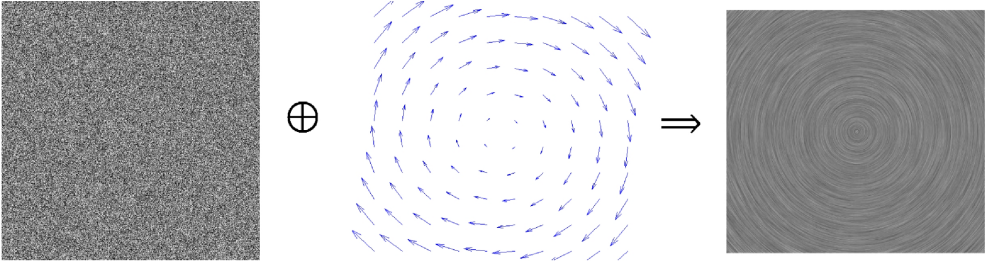
Figure 1: The LIC process.

of the image at point $\mathbf{p}$ is given by a convolution of a filter kernel $\mathrm{k}(\mathbf{x})$ with the intensity of an input image I evaluated along the streamline $\mathbf{s}\,(\mathbf{x},\mathbf{p})$, where $\mathbf{s}$ is defined as the curve that satisfies

$$\frac{\mathrm{d}\mathbf{s}}{\mathrm{d}\mathbf{x}} \times \mathbf{F}\,(\mathbf{s}) = \mathbf{0}\,. \tag{1}$$

Given these definitions, the LIC output image is calculated by evaluating

$$\mathrm{I}'\,(\mathbf{p}) = \int_{-\infty}^{\infty} \mathrm{k}\,(\tau)\,\mathrm{I}\,[\mathbf{s}\,(\mathrm{t}-\tau,\mathbf{p})]\,\mathrm{d}\tau \tag{2}$$

for all $\mathbf{p}$ (i.e., for every pixel in the image).

In practice, for each pixel in I we perform a forward and backward particle trace and then compute an average over all the pixels that the trajectory intersects. This results in an image where pixel intensities are spatially correlated according to the flow field as in Figure 1.

3.1.2 Image based flow visualisation

Image based flow visualisation (IBFV) is fundamentally an extension of LIC designed to handle time dependent flows $\mathbf{F} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. The streamline in Equation (2) is replaced with a pathline $\mathbf{P}\,(\mathbf{x},\mathbf{p},\mathrm{t})$, where $\mathbf{P}$ is defined as the

curve that satisfies

$$\frac{d\mathbf{P}}{dt} = \mathbf{F}\left(\mathbf{P}, t\right).\tag{3}$$

Furthermore, the input image I is modulated such that

$$I\left(\mathbf{p}, t\right) = \left(\omega t + \phi(\mathbf{p})\right) \mod I_{max},\tag{4}$$

where I_{max} is the maximum texture intensity, $\phi(\mathbf{p}) \in [0, I_{max}]$ is the phase shift for each point $\mathbf{p}$, and ω is a scaling factor. Substituting Equation (3) and Equation (4) into Equation (2) yields the convolution integral for IBFV,

$$I'\left(\mathbf{p}, t\right) = \int_{-\infty}^{\infty} k\left(\tau\right) I\left[\mathbf{P}\left(t - \tau, \mathbf{p}, t\right), t\right] d\tau.\tag{5}$$

The image at time $t + \Delta t$ is an affine combination of output images at time t with the input image at t.

3.2   Application to research case study datasets

All projects described in Section 2.2 were based upon 2D time dependent vector data fields that lay on unstructured grids. In each case, the research team sought significantly improved visualisations that could provide detailed information of the flow. Traditional techniques were initially employed; however, these led to visual clutter, difficulties with the representation of time dependence, or an inability to effectively trace and understand a flow feature of interest. We applied the LIC and IBFV techniques to these datasets and achieved improved visualisation outputs.

A significant problem encountered by the GPT project team was visual clutter (see Section 2.2.1). By applying LIC to a single snapshot of the collected data, a detailed view of the circulatory features of flow was achieved. Figure 2 illustrates the difference between the original cluttered arrow plot and the dense, image based LIC representation. The results and details of this visualisation technique were reported by Madhani et al. [12].
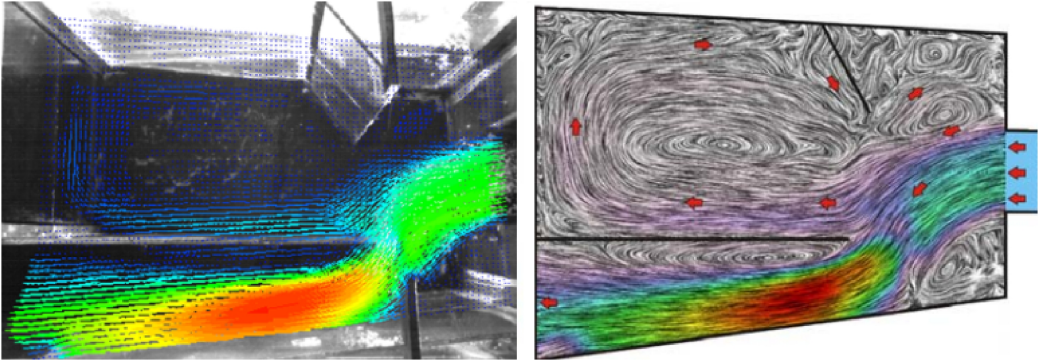
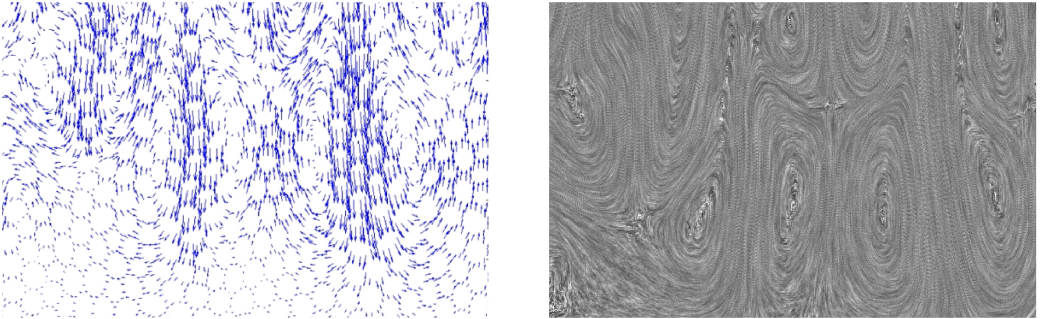Figure 2: Visualisation of the gross pollutant trap [12]: left, direct techniques; right, LIC algorithm.

Figure 3: Turbulent flows cause by the Eldar problem: left, direct techniques; right, single IBFV animation frame.

The evolution of the Eldar problem is a complex, time dependent flow. Traditional techniques were used to generate animated arrow plots. The researchers reported that these visualisations did not produce clear visual indicators of the flow or motion of vortices. In contrast, animations generated with IBFV achieved these well.

Example frames for comparison are given in Figure 3. Note that still images do not communicate the full effectiveness of the IBFV animations and we

Figure 4: The tidal null-point of the Pumicestone Passage: left, Direct technique: right, single IBFV animation frame.

encourage readers to consult this article's supplementary animation materials[1] to better appreciate the improvements.

An important flow feature of the Pumicestone Passage is a tidal null-point—a region of zero flow which divides the tidal flow and which moves spatially over time. This feature is well documented [10]. Using IBFV virtual dye experiments (see Section 4.1), the tidal null-point was easily located (i.e., the space between the red and green dye flows in Figure 4). The researchers had previously found it impossible to track this null-point using animated arrow plots.

---

[1]Download from http://eprints.qut.edu.au/54439/

4 Results

4.1 Effectiveness of visualisations

Through generation of effective visualisations using texture based techniques, we have contributed to the research outcomes of each project described in Section 2.2. This section summarises these contributions and discusses the features of texture based techniques that make them effective when applied to complex flows.

One of the most obvious advantages of texture based techniques is their dense spatial coverage. Both LIC and IBFV visually correlate pixels in space that lie on the same particle trajectory. As a result, texture based techniques are inherently effective at representing local and global flow regimes concurrently.

Critical points are important features of flow fields and a study by Laidlaw et al. [7] quantitatively showed that observers were able to identify critical points faster and more accurately with LIC than with other methods. Our experience with LIC and IBFV confirms these results. This was particularly true for the Edlar problem simulations, which were not amenable to meaningful visualisations using traditional animated arrow plots. The application of IBFV provided an effective visualisation tool that enabled observation of key details of the flows evolution, including movement of critical points. This provided both insight into the flow and visual model validation.

A failing of the LIC algorithm is that directional flow information is intractable (see Figure 1). This can be overcome by using animated LIC images [4] or indeed repeated application of the IBFV algorithm over a single steady field. Both methods produce animations that reveal orientation and direction of steady flow fields. Where animation cannot be employed (e.g., in a printed report or article), LIC combined with traditional techniques generate more effective visualisations. Figure 2 is an example of this.

IBFV-generated flow animations can be used for the effective visualisation of

complex time dependent flows. The spatial and temporal correlation features of ibfv combine to clearly portray particle trajectories and highlight critical points. Strong visual cues that depict flow provide the observer with context and an intuitive way to interpret velocity [6, 18].

Texture based methods routinely incorporate virtual dye experiments and thus track the long time particle evolution [18, 5]. This is implemented by utilising multiple advection buffers, each storing separate images (e.g., noise image, dye spot image) that are composited together at the rendering step [20]. Virtual dye experiments enabled the identification of distinct flow regions in the gpt [12, 11] which was a significant contribution to the research outcome. Application of virtual dye experiments to the Pumicestone Passage hydrodynamic model [10] enabled the verification of theories about the effects of tidal flows on creeks and provided a method for obtaining qualitative information about drainage of contamination sources.

4.2   Performance and implementation considerations

The computational overheads of texture based methods have historically made them unattractive for many users [9]. This is largely due to the dense nature of the technique: a particle trace is performed for every image pixel. Hence, these techniques are a members of the asymptotic complexity class $\Theta(n^3)$ where n is the dimension of the image.

Many software and hardware techniques have been applied over the years to reduce this overhead [9, 19], such as parallel implementations exploiting multicore processors and use of accelerators including graphics processing units (gpus). Furthermore, coupling these techniques with the high performance of modern gpus and cpus, interactive flow visualisations can be implemented on commodity hardware.

Implementation availability is the most likely cause for these techniques to not be used more widely [19]. However, the implementation of efficient texture

based algorithms only requires a basic understanding of graphics programming. For IBFV the algorithm is stated simply as follows.

1. *Warp mesh by the vector field*—Mesh vertices are advected by the flow through integration. Vertex connectivities are not changed, resulting in a distorted mesh.

2. *Render the mesh, texture mapped with previous image*—The previous frame is drawn on the warped mesh. Texture coordinates are not changed, resulting in a distorted version of the previous frame.

3. *Overlay a new noise image with opacity* $\alpha \in [0, 1]$—An affine combination of the distorted previous frame I and a new noise image N is generated. That is, $I' = (1 - \alpha)I + \alpha N$.

4. *Copy image to texture buffer*—To be used in step 2 for generating the next frame.

We developed our own C/OpenGL implementation of the IBFV algorithm. The code may be obtained by contacting the corresponding authors. A future objective is the integration of our texture based vector visualisation codes into several popular visualisation packages.

5   Conclusions

Texture based techniques can be exceedingly effective at visualising complex, time dependent flows. Unfortunately, access to and knowledge of these techniques appears limited. Through the three case studies described here, we have demonstrated that

- the application of texture based visualisation techniques allow flow fields to be represented in fine detail, and

- the increased visual effectiveness not only aids scientific communication, but can also have a direct effect on research outcomes.

References

[1] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH'93*, pages 263–270, 1993. doi:10.1145/166117.166151 C6

[2] B. Cumming, T. Moroney, and I. Turner. A mass-conservative control volume-finite element method for solving Richards' equation in heterogeneous porous media. *BIT Numer. Math.*, 51(2):845–864, 2011. doi:10.1007/s10543-011-0335-3 C6, C14

[3] H.-J. G. Diersch and O. Kolditz. Variable-density flow and transport in porous media: approaches and challenges. *Advances in Water Resources*, 25:899–944, 2002. doi:10.1016/S0309-1708(02)00063-5 C6

[4] C. D. Hansen and C. R. Johnson. *The visualization handbook*. Elsevier Butterworth-Heinemann, Burlington, MA, 2005. C2, C6, C11

[5] B. Jobard, G. Erlebacher, and M. Yousuff Hussaini. Lagrangian-Eulerian advection of noise and dye textures for unsteady flow visualization. *IEEE Trans. Visualization and Computer Graphics*, 8(3):211–222, 2002. doi:10.1109/TVCG.2002.1021575 C12

[6] P. R. Keller and M. M. Keller. *Visual cues: practical data visualization*. IEEE Computer Society Press, Los Alamitos, CA, 1993. C3, C12

[7] D. H. Laidlaw, R. M. Kirby, J. S. Davidson, T. S. Miller, M. da Silva, W. H. Warren, and M. Tarr. Quantitative comparative evaluation of 2D vector field visualization methods. In *Proc. IEEE Conf.*

Visualization '01, pages 143–150, 2001.
doi:10.1109/VISUAL.2001.964505 C11

[8] R. S. Laramee, G. Erlebacher, C. Garth, T. Schafhitzel, H. Theisel, X. Tricoche, T. Weinkauf, and D. Weiskopf. Applications of texture-based flow visualization. *Engineering Applications of Computational Fluid Mechanics*, 2(3):264–274, 2008. http://www.mpi-inf.mpg.de/~weinkauf/publications/abslaramee08.html C4, C5

[9] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualisation: dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–221, 2004. doi:10.1111/j.1467-8659.2004.00753.x C3, C4, C6, C12

[10] G. Larsen. Modelling hydrodynamic processes within Pumicestone Passage, northern Moreton Bay, Queensland. Master's thesis, Queensland University of Technology, School of Natural Resource Sciences, 2007. http://eprints.qut.edu.au/16634/ C6, C10, C12, C14

[11] J. T. Madhani, J. Young, and R. J. Brown. Image based flow visualisation of experimental flow fields inside a gross pollutant trap. In *AFMC 2012*, 2012. http://www.afms.org.au/conference/18/174-Brown.pdf C5, C12

[12] J. T. Madhani, J. Young, N. A. Kelson, and R. J. Brown. A novel method to capture and analyze flow in a gross pollutant trap using image-based vector visualization. *Water Air Soil Pollut.: Focus*, 9:357–369, 2009. doi:10.1007/s11267-009-9225-y C5, C8, C9, C12, C14

[13] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. Feature extraction and visualisation of flow fields. In *Eurographics 2002 State of the Art Reports*, pages 69–100, 2002. http://diglib.eg.org/EG/DL/Conf/EG2002/STARs/S4_FlowVis_Post.pdf C5

[14] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003. doi:10.1111/j.1467-8659.2003.00723.x C4

[15] A. Telea. *Data visualization: principles and practice.* A K Peters, Ltd, Wellesley, MA, 2008. C3, C5

[16] T. Theoharis, G. Papaioannou, N. Platis, and N. M. Patrikalakis. *Graphics & visualization: principles and algorithms.* A K Peters, Ltd, Wellesley, MA, 2008. C3

[17] E. R. Tufte. *The visual display of quantitative information.* Graphics Press, Cheshire, Conn, 2001. C5

[18] J. J. van Wijk. Image based flow visualization. In *SIGGRAPH'02*, pages 745–754, 2002. doi:10.1145/566570.566646 C12

[19] J. J. van Wijk. Views on visualization. *IEEE Trans. Visualization and Computer Graphics*, 12(4):421–432, 2006. doi:10.1109/TVCG.2006.80 C3, C12

[20] D. Weiskopf. *GPU-based interactive visualization techniques.* Springer-Verlag, Berlin, Heidelberg, 2007. C2, C6, C12

Author addresses

1. **D. J. Warne**, High Performance Computing and Research Support, Queensland University of Technology, Queensland 4001, Australia. mailto:david.warne@qut.edu.au

2. **J. Young**, High Performance Computing and Research Support, Queensland University of Technology, Queensland 4001, Australia. mailto:j.young@qut.edu.au

3. **N. A. Kelson**, High Performance Computing and Research Support, Queensland University of Technology, Queensland 4001, AUSTRALIA. `mailto:n.kelson@qut.edu.au`