



INTERACTIVE VISUALIZATION OF UNCERTAINTY IN FLOW FIELDS USING TEXTURE-BASED TECHNIQUES

Ralf P. Botchen¹, Daniel Weiskopf², Thomas Ertl¹

**¹ Institute for Visualization and Interactive Systems
University of Stuttgart, Germany**

**² Graphics, Visualization, and Usability Lab (GrUVi)
Simon Fraser University, Canada**

Keywords: *Flow Visualization, Uncertainty Visualization, Graphics Hardware*

ABSTRACT

We describe texture-based flow visualization techniques that bring out the uncertainties in real-world measured flow data or highlight the deviation of scenarios simulated with different numerical techniques. One visualization approach is based on a generic texture-filtering process that improves the perception of uncertainty-affected regions; the other approach focuses on a user-adjusted color coding of uncertainty. Both methods are implemented on graphics hardware and facilitate interactive visualization. The usefulness of these techniques is demonstrated for examples of simulation and PIV data sets.

1 INTRODUCTION

The visualization of vector fields has evolved to an important and interesting area of research over the last decades. The application fields for flow visualization range from scientific research to engineering disciplines; thereby a multitude of different visualization techniques have evolved [11], that have the ability of representing local and global flow features, for steady and unsteady vector fields. For a proper understanding of a real-world measured data set, it is not only important to visualize the underlying flow field, but also indicate how accurate and reliable the measuring system and thus the acquired data is. In a related context, it is of concern to compare results of different simulation algorithms and show how and where they differ.

On the way from data acquisition to the final step of visualization, there exist a number of possibilities how uncertainties or errors can affect the data [15, 22]. Some of them can be minimized or actually eliminated. For those that cannot be removed, it is advantageous to be aware of them. For example, in industrial applications it can be of great value to engineers to know the extent of uncertainty that occurs in the measured data, since the lack of this information can lead to false conclusions and thus significantly impact the process of data evaluation.

Following Griethe and Schumann [6], we use uncertainty as a generic term throughout this paper. Depending on the underlying problem, uncertainty can be understood as error, imprecision, inaccuracy,

or in general as data quality problem, just to name a few. In most cases, uncertainty is a quantitative measure represented by one scalar value like standard deviation, probability, or the difference between two data values.

Although it is a well known fact that nearly all data is affected by errors or uncertainties, most visualization techniques ignore this issue. The reason for this is that it is challenging to find a good representation of uncertainty and combine this representation with an existing visualization method. In this work, we extend our previous algorithmic techniques [1] and apply them to practical data sets from CFD (computational fluid dynamics) and flow measurements. Texture advection is used as basic technique for vector field visualization because it is efficient and versatile. We follow two different, yet compatible approaches to incorporate uncertainty values in the visualization: first, an additional texture filtering step that changes the spatial frequencies of the visualization by smearing out particle traces (Section 5); second, color coding layered on top of texture advection (Section 6). We demonstrate that these visualization methods lend themselves to fast GPU (graphics processing unit of modern graphics cards) implementations that facilitate interactive data exploration.

2 RELATED WORK

There has been an increasing interest in uncertainty visualization in a broad field of research areas in the last few years. Pang *et al.* [15] provide a classification of uncertainty visualization techniques, based on the visualization techniques from which the uncertainty data is derived. Two related publications [12, 22] present a wide range of uncertainty visualization techniques, which vary from glyphs, streamlines, and envelopes for flow-based uncertainties to animation. Here, it is also suggested that uncertainty is used to manipulate the geometric shape of objects or affect the oscillation of moving objects. Another approach that controls motion according to the error extent is presented by Brown [2], where the amount of error steers the degree of visual vibration of an object. In volume rendering, Johnson and Sanderson [8] use scalar uncertainty values to manipulate the thickness of an isosurface. Kniss *et al.* [9] present a direct volume rendering technique that allows users to explore the uncertainty of surface boundaries. They apply color maps based on sensitivity (change in boundary position per unit change in importance) to highlight uncertainty areas. Pang [14] presents visualization techniques that are capable of handling large amounts of geo-spatial data, including the additional variate of uncertainty. The reader is referred to Grieth and Schumann [6] for an overview of the definition of uncertainty, various uncertainty classifications, and a collection of uncertainty visualization techniques.

This paper builds upon conventional texture advection as a basic technique for flow visualization. Early versions of texture advection [13] were extended to 2D Lagrangian-Eulerian Advection [7], 2D Image Based Flow Visualization [18], or GPU-based texture advection [21]. Texture advection is directly related to Line Integral Convolution (LIC) [3]: most examples of texture advection can be considered as LIC with an exponential filter kernel [4]. All these texture-based techniques have the advantage of a dense representation, which densely covers the domain by streamlines or other characteristic curves. Therefore, texture-based flow visualization overcomes the seed-point positioning problem, i.e., the issue of choosing appropriate starting points for particle tracing. We refer to Laramee *et al.* [11] for an overview of the state of the art in texture-based flow visualization.

3 SIMULATION AND MEASURING TECHNIQUES IN A NUTSHELL

In this section we give a short overview of CFD simulation techniques that intend to represent the behavior of turbulent flow in some way. The governing equations are the Navier-Stokes equations, which represent the conservation of mass, momentum, and energy, and can describe the behavior of fluids or gas in turbulent flows. They are a system of nonlinear partial differential equations of second order that express the dependency of velocity and pressure as function of time and space. For an incompressible fluid with constant density and divergence $\nabla \cdot \mathbf{u} = 0$, they can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{\rho} \mathbf{f} - \frac{1}{\rho} \nabla P + \nu \Delta \mathbf{u}, \quad (2)$$

where ρ denotes the mass density, the vector \mathbf{f} is the body force and stands for the acceleration due to external force (e.g. gravity), P is the scalar pressure, and ν the dynamic viscosity of the medium. The following techniques are most commonly used in CFD.

Direct Numerical Simulation (DNS) directly solves the Navier-Stokes equations numerically without any turbulence model. The used algorithms vary in the method that solves the equations, e.g. explicit or implicit techniques can be applied. For explicit methods, the time step has to be proportional to the spatial grid in order to keep the computation stable. Implicit methods allow for bigger time steps to improve computation performance. On the one hand, these simulations are capable of representing all temporal and spatial scales of turbulence; on the other hand, they require very large computational resources. Especially for turbulent flow with many small structures, DNS needs a high-resolution grid to conserve the kinetic energy, which leads to exceeding computational costs. Therefore, DNS is mainly applied to small grids or flows with relatively low Reynolds numbers.

Reynolds-Averaged Navier-Stokes (RANS) methods solve the Reynolds equations, which are time-averaged Navier-Stokes equations. These equations can be acquired by separating flow velocity into mean and fluctuating parts, introducing new terms known as Reynolds stresses. In contrast to DNS, this approach can be applied to flows in complex geometry and at high Reynolds numbers at very low computational costs and is therefore commonly used in CFD. However, RANS methods require the use of turbulence models that characterize all unsteady turbulent motion. Unfortunately, no turbulence model has been developed to this day, that is capable of representing the turbulent motion accurately for a wide variety of flows. This limitation of RANS has led to increasing interest in other simulation techniques.

Large-Eddy Simulation (LES) is based on Kolmogorov's [10] theory of self similarity, and was first proposed by Smagorinsky [16]. LES is capable of resolving the major part of the kinetic energy of turbulent motion, where only the small structures have to be modeled. The popularity of LES is based on the fact that the kinetic energy content of turbulent flow decreases with increasing wave number, whereby the major portion of the Reynolds stress can be solved. LES techniques solve space-filtered and/or time-filtered Navier-Stokes equations. Therefore, it is possible to directly compute the large scales (i.e. the large eddies) with less computational costs. The effects of the smaller flow structures have to be modeled by a sub-grid scale (SGS) model, since low-pass filtering introduces unknown quantities. The main advantage of LES is that the computational costs are substantially smaller than for DNS (but higher than for RANS), since not all scales have to be solved.

Detached-Eddy Simulation (DES) [17] is a hybrid approach for the prediction of turbulent motions in flow fields with high Reynolds numbers. It uses a single turbulence model that has the capacity of an SGS model in regions with a grid resolution fine enough for LES computation, and RANS in all other areas. More precisely, RANS is applied in zones where the turbulent length scale is less than the maximum grid dimension or near solid boundaries. Since the computational demands of pure LES increase significantly in the vicinity of walls, zonal approaches like DES are often employed to save computation costs. A weak point of those hybrid approaches is that RANS and LES have different requirements to the underlying grid and, therefore, the success of the computation highly depends on the quality of the selected sub-grids for the different methods.

Particle Image Velocimetry (PIV) is a real-world measuring technique that, for example, was used to acquire the data shown in Section 7. The basic principle of PIV is to inject particles into the flow and to measure the movement of these particles between two light pulses. Usually, a planar laser light sheet technique is used. In very short intervals, the target area is illuminated twice by a double-pulsed laser and recorded by the CCD array of a digital camera. Since the CCD chip must be able to capture each light pulse in separate image frames, the resolution in time is bound by the image frequency of the camera. Afterwards, appropriate algorithms evaluate consecutive images and determine the displacement of particles in the flow.

This paper primarily focuses on example data acquired by LES and DES simulations or PIV measurements, as discussed in detail in Section 7. LES and DES computations are used in comparative visualizations that interpret differences between different simulation results as generic uncertainty. In contrast, PIV data is affected by measurement errors that are described as uncertainty.

4 SEMI-LAGRANGIAN TEXTURE ADVECTION

We use semi-Lagrangian texture advection as basic technique for flow visualization because it is an efficient and flexible method that can handle steady and unsteady flow alike. Our implementation is based on the texture advection method by Van Wijk [18] and the semi-Lagrangian scheme by Jobard *et al.* [7]. This section briefly reviews semi-Lagrangian texture advection.

Particles or injected dye are represented in an Eulerian fashion on a uniform grid in the form of a sampled property field $\rho(\mathbf{x})$, with points \mathbf{x} on the flow domain. A uniform grid can also be regarded as a texture—with texels (texture elements) being the cells of that grid. From a Lagrangian point of view, particles follow pathlines governed by

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}(\mathbf{x}(t), t),$$

with the vector field $\mathbf{v}(\mathbf{x}, t)$ and time t . Formal integration leads to the pathline

$$\mathbf{x}(t_1) = \mathbf{x}(t_0) + \int_{t_0}^{t_1} \mathbf{v}(\mathbf{x}(t), t) dt. \quad (3)$$

Combining an Eulerian particle description and a Lagrangian particle transport, we obtain a semi-Lagrangian update scheme:

$$\rho(\mathbf{x}(t_0), t_0) = \rho(\mathbf{x}(t_0 - \Delta t), t_0 - \Delta t). \quad (4)$$

Here, a backward lookup is employed at a location of the previous time step, $\mathbf{x}(t_0 - \Delta t)$, along with an integration backwards in time. Euler integration is often used because only short “streaklets” are

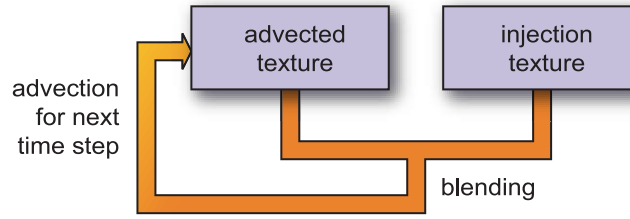


Fig. 1 Texture advection and injection by alpha blending.

constructed: $\mathbf{x}(t_0 - \Delta t) = \mathbf{x} - \Delta t \mathbf{v}(\mathbf{x}, t_0)$. Bilinear interpolation is employed as reconstruction filter for the property field.

An advantage of texture advection is its support of different visualization styles, ranging from large-structure dye patterns to a dense collection of particle traces known from LIC [3]. The visualization style is controlled by injecting different types of patterns that are blended with the property field ρ after each advection step [18]: dye advection is achieved by injecting large, smooth dye sources and LIC-like images are obtained by injecting noise images. Figure 1 illustrates the texture advection and alpha-blending process. Another important advantage is that texture advection allows for efficient, interactive GPU implementations [21].

5 TEXTURE-BASED UNCERTAINTY VISUALIZATION

Texture advection as described in the previous section is capable of showing the vector field of the flow (i.e. a single vector at each data point), but needs to be extended to visualize the associated uncertainty. The goal of this section is to include the visualization of uncertainty or error.

We assume that uncertainty is described by a scalar value, e.g. the magnitude of a measurement error or data difference. The original uncertainty value may be mapped to a derived measure, e.g. by a linear or a non-linear function, in order to emphasize uncertainty ranges and obtain useful values. The result of this mapping is denoted by the uncertainty measure $u(\mathbf{x}, t)$. Uncertainty visualization addresses the display of $u(\mathbf{x}, t)$.

This section targets uncertainty visualization incorporated in the texture representation of the flow. The crucial observation is that uncertainty of the vector data results in an uncertainty of pathlines. Therefore, the particle transport should take into account $u(\mathbf{x}, t)$ by modifying the advection process, as proposed in our previous work [1]. In fact, the pipeline from Figure 1 is just extended by one additional image-filtering step that works on the advected texture. This uncertainty filter affects the advected property field ρ according to the following 2D convolution:

$$\rho_{\text{filtered}}(\mathbf{x}) = \int_{V(\mathbf{x}, \mathbf{v})} f(u, \tilde{\mathbf{x}}, \mathbf{v}) \rho(\mathbf{x} + \tilde{\mathbf{x}}) d^2 \tilde{\mathbf{x}}. \quad (5)$$

In contrast to many other convolution filters, the filter f may be space-variant and flow-dependent, and so is the integration domain $V(\mathbf{x}, \mathbf{v})$. The goal of the convolution filter is to modify the spatial frequency perpendicular to particle traces: it essentially smears out particle traces. While LIC or traditional texture advection only modify the spatial frequencies along the flow, uncertainty-aware texture advection also affects the spatial frequencies perpendicular. The specific character of the uncertainty visualization is

controlled by the choice of filter kernel and integration domain. Two sets of examples are described below.

The first example is *cross advection*, which employs filtering along a line perpendicular to the vector field. In this way, particles are additionally transported perpendicular to the flow direction, smearing out the particle trace. The discretized version of filtering reads

$$\rho_{\text{filtered}} = \sum_{i \in \{-1,0,1\}} f_i \rho_i, \quad (6)$$

with the filter kernel f_i and the property values ρ_i that are sampled along the perpendicular line:

$$\rho_i = \rho(\mathbf{x} + i\mu\Delta t M_{\text{rot}} \mathbf{v}(\mathbf{x}, t)). \quad (7)$$

We use a symmetric filter kernel that is normalized (i.e. $\sum f_i = 1$) to maintain a constant overall brightness. The amount of smearing is controlled by the relative step size μ , which is determined by $u(\mathbf{x}, t)$. The matrix M_{rot} implements a rotation by 90 degrees in order to obtain a filtering perpendicular to the vector field.

A variant of cross advection uses a maximum function instead of the weighted sum in Eq. (6):

$$\rho_{\text{filtered}} = \max\{\rho_i | i = 1, 2, 3\}. \quad (8)$$

The maximum function avoids that streaklines are faded out to small property values in regions of large uncertainty, which is especially noticeable for sparsely seeded streaklines. However, the maximum function does not guarantee a constant overall brightness. In fact, images tend to increase in intensity and, thus, the maximum-function approach is mainly used for sparse representations with only few, distinct streaklines.

Error diffusion is an alternative example of uncertainty visualization. Here, an isotropic filter kernel is used for the filtering equation (5). The amount of smearing is controlled by the uncertainty: $u(\mathbf{x}, t)$ determines the width (i.e. size) of the filter kernel. The larger the uncertainty, the wider the kernel. Typically, we use a 2D Gaussian filter kernel that is normalized to maintain a constant overall brightness. Because an error-affected data point influences its adjacent region, this filtering process imitates natural diffusion. The maximum function is not recommended as filter function for error diffusion because it would lead to an extreme increase in brightness caused by the larger footprint of error diffusion (as compared to cross advection).

Error diffusion and cross advection can be considered as image-processing methods that work on 2D textures. Both techniques can be directly mapped to efficient GPU programs, the details of which can be found in [1]. Cross advection basically uses a GPU fragment program to step along the perpendicular direction and sum the terms from Eq. (6). Error diffusion sums the terms within the support of the 2D filter kernel. A naive implementation of 2D image filtering could lead to a poor visualization speed for large kernels because the number of filter kernel entries and the computational costs increase quadratically with the resolution of filter width. We overcome this problem by successively applying a small filter kernel. In the case of Gaussian filtering, successive application results in a Gaussian kernel of larger width. We use a discrete, separated 3×3 Gaussian filter, where the number of filtering steps is controlled by the uncertainty value. Furthermore, a fine adjustment of the filter is obtained by adapting the entries in the 3×3 filter mask. As demonstrated in Section 7, GPU implementations lead to interactive visualizations even for very large data sets in the range of 10^6 grid points.

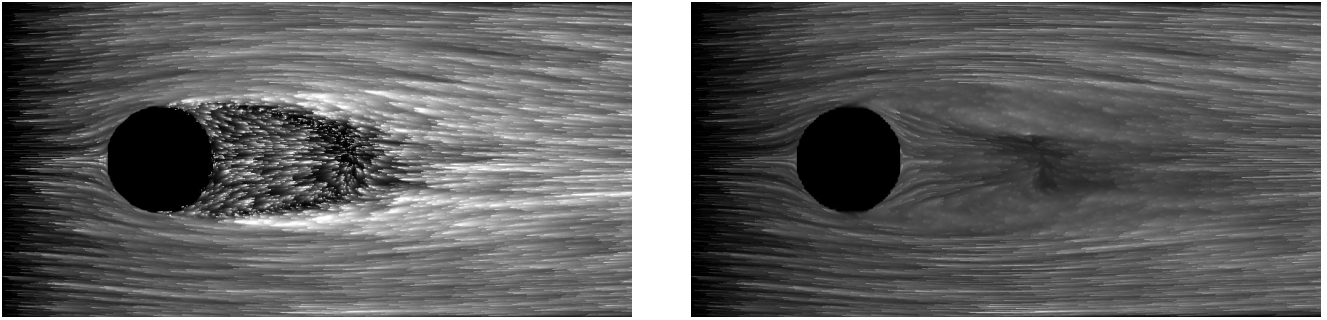


Fig. 2 Two examples of uncertainty-dependent filtering techniques, representing $z/D \approx -1.016$ —the 84th layer of the LES cylinder data set. Left: cross advection. Right: error diffusion.

Figure 2 illustrates the effect of cross advection (with maximum filter) and error diffusion. The test data set for this figure originates from a LES simulation of flow around a cylinder and a corresponding uncertainty measure that are both described in more detail in Section 7. The visualizations show the highest amount of “smearing” in regions behind the top and bottom of the cylinder, indicating those regions as areas of highest uncertainty.

6 COLOR-BASED UNCERTAINTY VISUALIZATION

Although texture-based uncertainty visualization already leads to an intuitive visual encoding of uncertainty information, additional visual cues can be used to further enhance perception. Besides spatial visual structures, which are exploited by texture-based visualization, color generally plays a dominant role in human visual perception. An advantage is that color and shape or position are highly separable perceptual dimensions [20, p. 180]—color and texture patterns can be easily combined without impairing each other. Therefore, we simultaneously use color and texture patterns for improved visualization.

The first approach directly maps the uncertainty information $u(\mathbf{x}, t)$ to color by employing a 1D color table. This color table takes the uncertainty as the 1D input parameter and outputs an RGB (red, green, blue) value. The resulting color is combined with the gray-scale property texture from the advection process by blending or modulation.

A popular way of using a color table is to display quantitative data. It is important that the color table is designed in a way that supports the accurate perception of quantitative data, for example, by using a perceptually uniform color scale. More details on the design of color tables are discussed by Ware [19]. Figure 3 (right) illustrates color coding of uncertainty for the same LES data set that is used in Figure 2. For comparison, Figure 3 (left) shows texture advection without color coding.

Another application of color tables uses color to separate regions—essentially labeling those regions. In this sense, color is mainly used to encode nominal information; color can be extremely effective as a nominal code [20, p. 123]. Each color represents an interval of uncertainty values. The advantage of such a discrete color coding is that regions can be more easily recognized than with a quantitative color scale. The discrete color coding is particularly useful when a qualitative understanding of the visualization is more important than reading off quantitative data. Figure 4 (left) demonstrates discrete color coding for the LES test data set.

Another useful method is uncertainty edge detection. This approach is a 2-pass rendering technique.

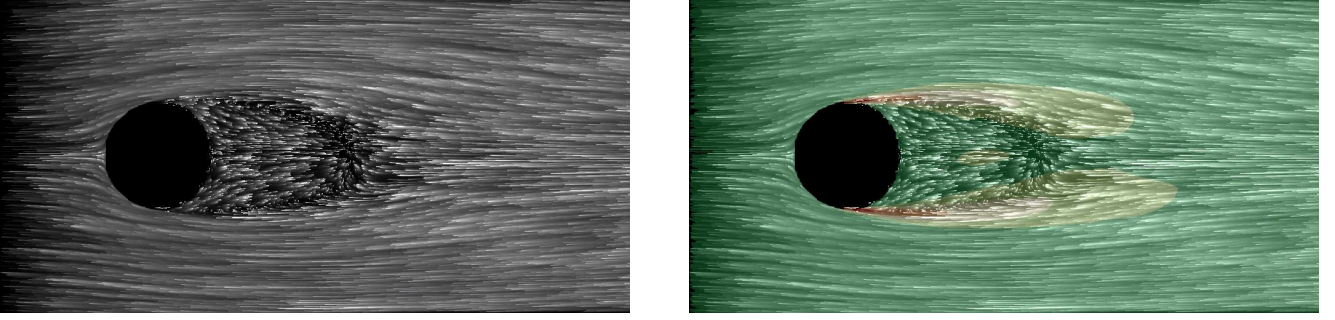


Fig. 3 Left: traditional texture advection of the 84th layer ($z/D \approx -1.016$) of the LES cylinder data set. Right: the same layer with uncertainty-controlled color coding by a green-yellow-red map.

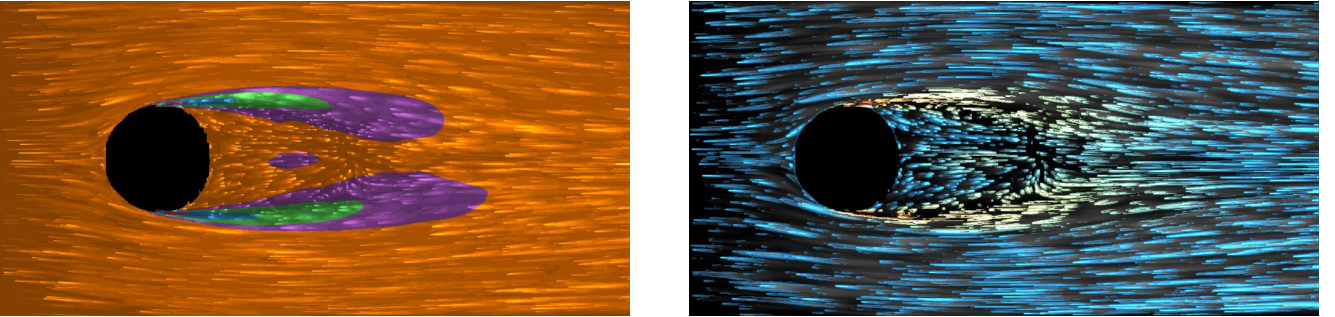


Fig. 4 Examples of texture-based flow visualization techniques showing the 84th layer ($z/D \approx -1.016$) of the LES cylinder data set. Left: qualitative color mapping with discrete colors. Right: uncertainty-edges technique combined with a spectral (blue to red) color map.

In the first stage, traditional texture advection is employed as described in Section 4. In the second stage, an edge-detection filter—namely a 3×3 Sobel operator—is applied to the result of stage one, extracting all edges, which generally are located around the injected particle and along the side of its short streakline. In the same step, a color mapping of the uncertainty value is applied. If no edge exists at the evaluated region, the result from stage one—the color of the original particle traces—is used instead. An advantage of this technique is that it does not modify or filter the original streaklines, computed in the first stage and, thus, additional information (e.g. pressure or density) could be encoded. A slight disadvantage of this technique is that the edges can clutter the final visualization. Therefore, it is better suited for a sparse particle injection, where streaklines are clearly separated. Figure 4 (right) demonstrates uncertainty edge detection for the LES test data set.

7 RESULTS AND ANALYSIS

We have tested our visualization methods for wall-mounted finite cylinder data sets computed on an unstructured grid with 12.3 million grid points, for both LES and DES. Since our GPU visualization techniques are designed for Cartesian grids, to exploit the SIMD (single instruction multiple data) architecture of modern graphics hardware, it was necessary to resample the original data set on an equidistant grid. Therefore, we placed a cube with the dimensions $x \in [-1.5, 4.5]$, $y \in [-1.5, 1.5]$, and $z \in [0.0, -3.0]$ in the interesting region of the original data set, and resampled it with $256 \times 128 \times 128$ sample points.

Table 1 Rendering performance in frames per second, measured on a Pentium IV with 3.4 GHz and an NVIDIA GTX 7800 GPU. Techniques are (I) conventional texture advection, (II) cross advection, (III) error diffusion, (IV) uncertainty color coding, and (V) uncertainty edges, respectively.

viewport	I	II	III	IV	V
1024×512	901	429	507	822	539

The simulation for both original data sets was performed on the same grid with a Reynolds-number of 200,000. The final data is averaged, using 3083 single time steps. Computation timings for the original cylinder data sets are given for a cluster computer with 42 IBM pSeries 690 PCs with 1.3 GHz. For the 12.3 million grid points, it took the cluster 3.1 minutes for LES and 5.0 minutes for DES per simulated time step. More details on the original data sets and its acquisition can be found in [5]. Rendering performance results for all techniques are shown in Table 1.

The visualization of the DES and LES simulations is an example of comparative visualization, i.e., the difference between different simulation results serves as uncertainty measure. All images of the simulated data sets shown in this paper, use the magnitude of the velocity difference vector as uncertainty extend, whereas other values like pressure, density or temperature can be used instead, if they are a matter of particular interest. High uncertainties appear in regions with vortex separations near the surface or in shear layers behind the cylinder. Figure 5 shows two different visualizations of the DES data set. The left image uses a green color map, sequentially enhancing saturation with increasing uncertainty. This kind of mapping is used in most applications since color and uncertainty are intuitively coupled. Nevertheless, the use of a diverging color map, as shown in the right image, can emphasize the visualization. While blue regions with very low uncertainties fade to the background, the red-colored regions with high uncertainties are emphasized. In this results section, we exclusively use a discrete color table when color coding is applied.

In Figures 6 and 7, we compare two different layers of both simulations. Figure 6 shows the difference of velocity magnitude in the computed shear layers behind the cylinder with a sequential red color table. Figure 7 illustrates vortex separations near the upper edge of the cylinder with a spectral color map. Noticeable are the small differences in flow directions of both data sets.

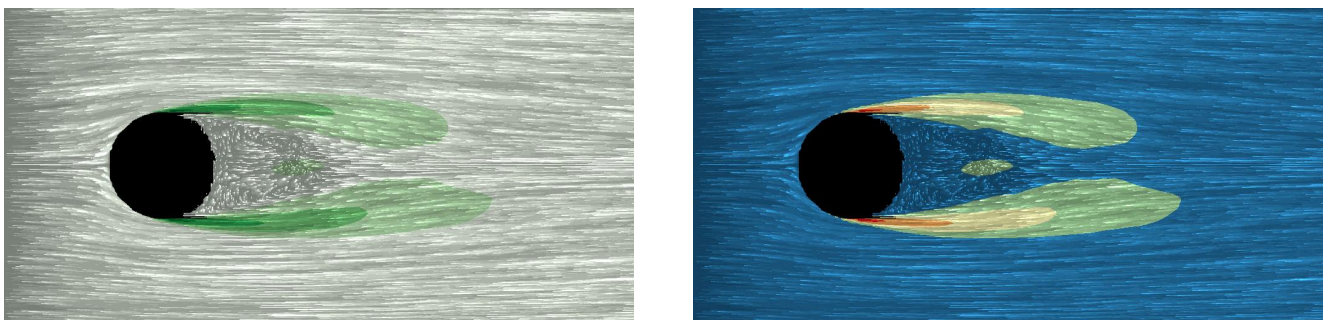


Fig. 5 The 84th layer ($z/D \approx -1.016$) of the DES data set, with two different color maps applied. Left: sequential green color coding. Right: diverging spectral (blue-red) color table.

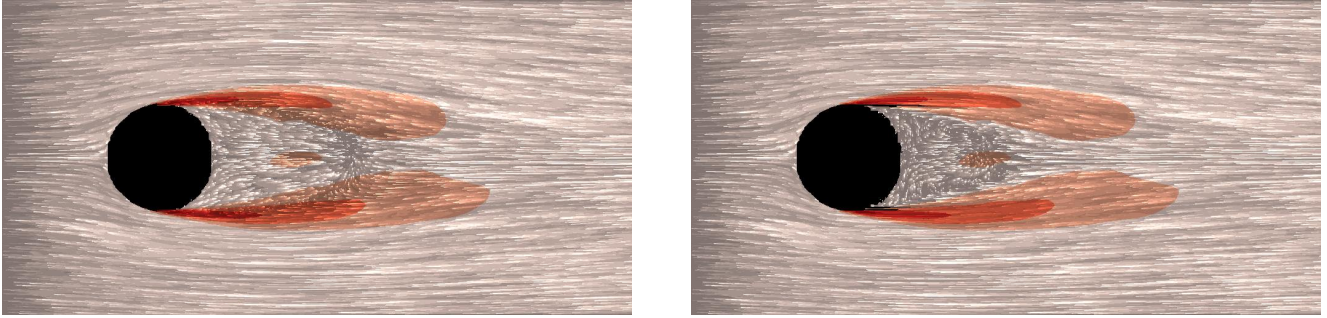


Fig. 6 Comparing the 84th layer ($z/D \approx -1.016$) of the LES (left) and DES (right) data sets, encoded by sequential red colors.

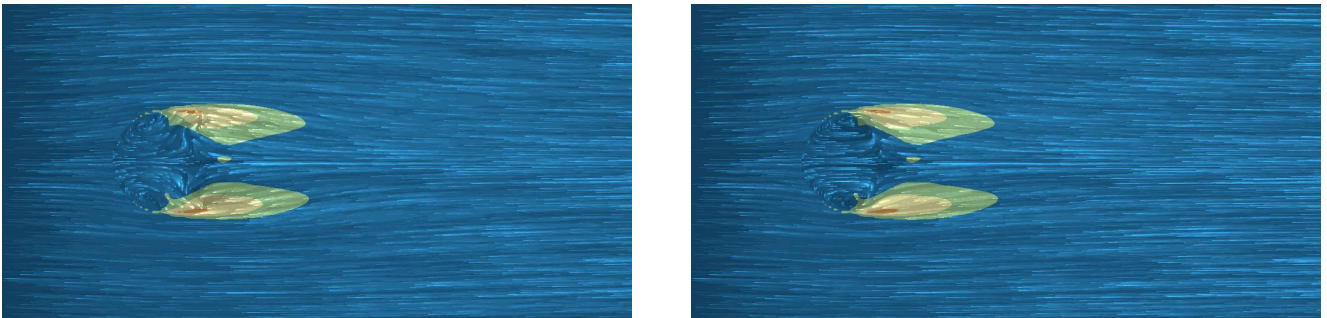


Fig. 7 Comparison of LES (left) and DES (right) with a spectral (blue-red) color map. Shown is the 40th layer ($z/D \approx -2.055$), located right on top of the cylinder.

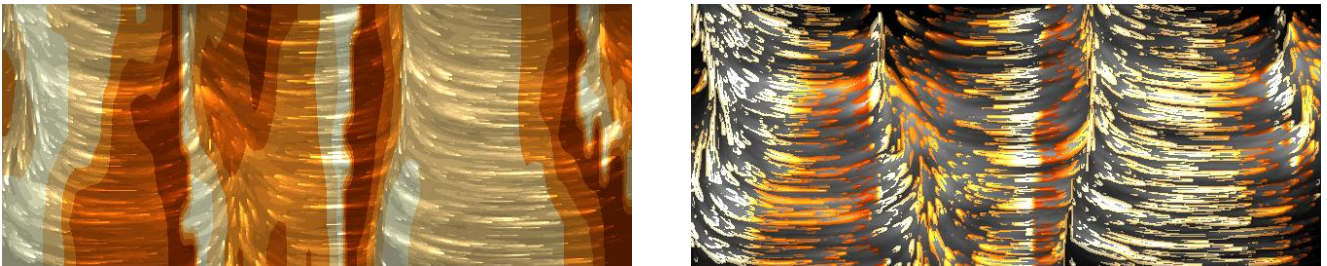


Fig. 8 Illustration of layer 6 of the PIV data set, rendered with a sequential (orange) color map. Left: blended with the texture-advection result. Right: applied to uncertainty edges.

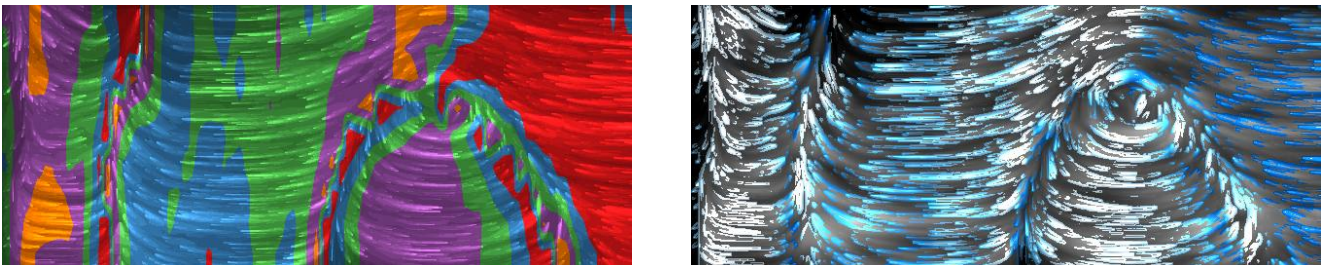


Fig. 9 The 8th layer of the PIV data set. Left: a combination of cross advection and a discrete, qualitative color map. Right: uncertainty edge detection combined with a sequential color map.

Another test data set was measured by a 3D S-PIV method in a laminar water channel. The original data has 81×45 measurement points in each of nine separated layers and contains one time step of water inflow, streaming through the channel and forming vortices in the current. For the final data, the experiment was conducted 25 times and averaged. As uncertainty value we use the root mean square over all 25 measurements [1].

Figures 8 and 9 show the sixth and eighth layer of the measured data set with different color maps. The visualization shows that the areas of high uncertainty lie in regions where the flow moves in the direction of the third dimension, i.e., a weakness of this PIV method is the accuracy in the third dimension, which is plausible since the measurements are performed on 2D layers.

The left images of Figures 8 and 9 apply a color map to the complete texture-based flow visualization, i.e., to the whole domain. In contrast, the right images only apply color coding to uncertainty edges; here, uncertainty is only shown in regions where particles are visible, freeing image estate in the other areas. As mentioned before, these unneeded areas could be used to display an additional variate (e.g. pressure). Both images in Figure 8 and the right image in Figure 9 use a sequential color table that maps the degree of uncertainty from low (white) to high values (highest color saturation). In Figure 9 (left), the qualitative color mapping represents five different regions of uncertainty: high uncertainties are represented by red or blue colors, whereas regions with no or little uncertainty are mapped to orange and purple color and mid-range values are displayed as green color. In this way, the viewer's attention is drawn to regions with strong deviations.

8 CONCLUSION

We have presented texture-based flow visualization methods that are capable of handling uncertainty as an additional variate in order to manipulate the affected regions of the flow. The main advantage of our approaches is their flexibility and compatibility with traditional visualization techniques. Moreover, the real-time capability of all methods allows the user to change parameters while interactively exploring the flow. In future work, we will extend the capability of our system to handle higher-dimensional uncertainty values. Furthermore, the extension to 3D flow visualization is left open as challenge for future work.

ACKNOWLEDGEMENTS

The authors are indebted to Octavian Frederich for his help in providing the simulation data and the adjuvant discussions. We also like to thank Prof. Ulrich Rist, Andreas Babucke, and Kudret Baysal for providing the PIV data set. The first author specially thanks Sarah Schnauffer for the kind help of editing all transfer functions, and the second author acknowledges support from NSERC (Discovery Grant).

REFERENCES

- [1] R. P. Botchen, D. Weiskopf, and T. Ertl. Texture-based visualization of uncertainty in flow fields. In *Proceedings of IEEE Visualization 2005*, pages 647–654, 2005.
- [2] R. Brown. Animated visual vibrations as an uncertainty visualisation technique. In *Proceedings of GRAPHITE 2004*, pages 84–89, 2004.
- [3] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of ACM SIGGRAPH 1993*, pages 263–270, 1993.

- [4] G. Erlebacher, B. Jobard, and D. Weiskopf. Flow textures: High-resolution flow visualization. In C. D. Hansen and C. R. Johnson, editors, *The Visualization Handbook*, pages 279–293. Elsevier, Amsterdam, 2005.
- [5] O. Frederich, E. Wassen, and F. Thiele. Flow simulation around a finite cylinder on massively parallel computer architecture. In *Proceedings of the International Conference on Parallel Computational Fluid Dynamics 2005*, Washington D.C., USA, 2005.
- [6] H. Griethe and H. Schumann. The visualization of uncertain data: Methods and problems. In *Proceedings of SimVis 2006*, pages 143–156. SCS Publishing House, 2006.
- [7] B. Jobard, G. Erlebacher, and M. Y. Hussaini. Lagrangian-Eulerian advection of noise and dye textures for unsteady flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):211–222, 2002.
- [8] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10, 2003.
- [9] J. M. Kniss, R. V. Uitert, A. Stephens, G.-S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *Proceedings of IEEE Visualization 2005*, pages 287–294, 2005.
- [10] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *C. R. (Doklady) Acad. Sci. URSS (N.S.)*, 30:301–305, 1941.
- [11] R. S. Laramée, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, 23(2):143–161, 2004.
- [12] S. K. Lodha, B. Sheehan, A. T. Pang, and C. M. Wittenbrink. Visualizing geometric uncertainty of surface interpolants. In *Graphics Interface '96*, pages 238–245, 1996.
- [13] N. Max and B. Becker. Flow visualization using moving textures. In *Proc. ICASW/LaRC Symp. Visualizing Time-Varying Data*, pages 77–87, 1995.
- [14] A. Pang. Visualizing uncertainty in geo-spatial data. In *Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology*, Arlington, 2001.
- [15] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997.
- [16] J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91(3):99–164, 1963.
- [17] P. R. Spalart, W. H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In *Proceedings of 1st AFOSR International Conference on DNS/LES*, pages 137–147, 1997.
- [18] J. J. van Wijk. Image based flow visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002.
- [19] C. Ware. Color sequences for univariate maps. *IEEE Computer Graphics and Applications*, 8(5):41–49, 1988.
- [20] C. Ware. *Information Visualization: Perception for Design*. Elsevier, Amsterdam, second edition, 2004.
- [21] D. Weiskopf, M. Hopf, and T. Ertl. Hardware-accelerated visualization of time-varying 2D and 3D vector fields by texture advection via programmable per-pixel operations. In *Proceedings of VMV 2001*, pages 439–446, 2001.
- [22] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996.