PDE Methods in Flow Simulation Post Processing

Joachim Becker Tobias Preußer Martin Rumpf

November 5, 2001

Abstract

Vector field visualization is an important topic in scientific visualization. The aim is to graphically represent field data in an intuitively understandable and precise way. Two novel methods are described which enable an easy perception of flow data. The texture transport method especially applies to timedependent velocity fields. Lagrangian coordinates are computed solving the corresponding linear transport equations numerically. Choosing an appropriate texture on the reference frame the coordinate mapping can be applied as a suitable texture mapping. Alternatively, the aligned diffusion methods serves as an appropriate scale space method for the visualization of complicated flow patterns. It is closely related to nonlinear diffusion methods in image analysis where images are smoothed while still retaining and enhancing edges. Here an initial noisy image is smoothed along streamlines, whereas the image is sharpened in the orthogonal direction. The two methods have in common that they are based on a continuous model and discretized only in the final implementational step. Therefore, many important properties are naturally established already in the continuous model.

1 Introduction

The visualization of field data, especially of velocity fields from CFD computations is one of the fundamental tasks in scientific visualization. A variety of different approaches has been presented. The simplest method to draw vector plots at nodes of some overlayed regular grid in general produces visual clutter, because of the typically different local scaling of the field in the spatial domain, which leads to disturbing multiple overlaps in certain regions, whereas in other areas small structures such as eddies can not be resolved adequately. The central goal is to obtain a denser, intuitively better receptible method. Furthermore it should be closely related to the mathematical meaning of field data, which is mainly expressed in its one to one relation to the corresponding flow. If a vector field $v : \Omega \times \mathbb{R}_0^+ \to \mathbb{R}^n$ for some domain $\Omega \subset \mathbb{R}^n$ is given, then the corresponding flow $\phi : \Omega \times \mathbb{R}_0^+ \to \Omega$ is described by the system of ordinary differential equations

$$\partial_t \phi(x,t) = v(\phi(x,t),t)$$

and the initial condition $\phi(x,0) = x$.
Single particle lines only very partially enlighten features of a complex flow field. Thus, we ask for some dense pattern which represents the flow ϕ globally on the computational domain. We propose two different approaches to toggle this important problem in vector field post processing. They have in common that they are both based on the solution of suitable PDE problems.
We may either solve the flux equation itself for suitable initial and boundary conditions. Thereby, in explicit we calculate a Lagrangian coordinate mapping with respect to a suitable coordinate frame, typically inflow position and inflow time. It can be applied as a texture map which overlays the computational space with a texture pattern and has been prescribed on the corresponding Lagrangian reference domain. If the flow problem incorporates inflow boundary conditions, the inlet coordinates and the inflow time can be chosen as a natural coordinate frame. This method can be interpreted as a simultaneous tracing of all points on the inlet studying their long time behaviour. Numerically we solve the underlying hyperbolic linear transport problem with respect to the given velocity $v(x,t)$ by a higher order discontinuous Galerkin method due to Cockburn et al. [6, 7]. The approach especially applies to timedependent velocity fields, where it actually depicts the geometry of particle lines. The texture mapping is continuous in time and therefore enables animating.
Alternatively, we may ask for a method to generate stretched streamline type patterns, which are aligned to the vector field $v(x,t)$ for a fixed time t. The possibility to successively coarsen this pattern is a further desirable property. Methods which are based on such a scale space and enhance certain structures of images are wellknown in image processing analysis. Actually nonlinear diffusion allows the smoothing of grey or color images while retaining and enhancing edges [17]. Now we set up a diffusion problem, with strong smoothing along streamlines and edge enhancing in the orthogonal directions. Applying this to some initial random noise image we generate a scale of successively coarser patterns which represent the flow field at some instance in time. Finite Elements in space and a semi implicit timestepping is applied to solve the diffusion problem numerically.
First results on the texture transport method have already been presented in [3]. We find it appropriate to pick them up here and compare them with the field aligned diffusion method as another PDE based approach. Before we explain in detail the two methods, let us discuss related work on vector field visualization and image processing. Later on we will identify some of the well known methods as equivalent to special cases, respectively asymptotic limits of the presented new methods.

2 Related Work

The spot noise method proposed by van Wijk [21] introduces spot like texture splats which are aligned by deformation to the velocity field in 2D or on surfaces in 3D. These splats are plotted in the fluid domain showing strong alignment patterns in the flow direction. The originally first order approximation to the flow was improved by de Leeuw and van Wijk in [9] by using higher order polynomial deformations of the spots in areas of significant vorticity. In an animated sequence these spots can be moved

along streamlines of the flow. Furthermore in 3D van Wijk [22] applies the integration to clouds of oriented particles and animates them by drawing similar moving transparent and illuminated splats.
The Line Integral Convolution (LIC) approach of Cabral and Leedom [5] integrates the above ODE forward and backward in time at every pixelized point in the domain, convolves a white noise along these particle paths with some Gaussian type filter kernel, and takes the resulting value as an intensity value for the corresponding pixel. According to the strong correlation of this intensity along the streamlines and the lack of any correlation in the orthogonal direction the resulting texturing of the domain shows dense streamline filaments of varying intensity. Hege and Stalling [19] increased the performance of this method especially by reusing portions of the convolution integral already computed on points along the streamline. Max et al. [16] proposed a similar method on surfaces. Max and Becker [15] present a method for visualizing 2D and 3D flows by animating textures.
Shen and Kao [18] have applied a LIC type method to unsteady flow fields and Interrante and Grosch [11] generalized line integral convolution to 3D in terms of volume rendering of line filaments.
In [20] Turk discusses an approach which selects a certain number of streamlines. They are automatically equally distributed all over the computational domain to characterize in a sketch type representation the significant aspects of the flow. An energy minimizing process is used to generate the actual distribution of streamlines.
Most of these methods are designed and implemented on flow fields, which are constant in time. If we for instance apply line integral convolution in the timedependent case, successive images of a time sequence are in general not correlated. Grey level values at grid points change very rapidly because the streamlines at time t and $t + \delta t$ on which the convolution is performed have almost no overlap even for very small δt.
Furthermore, if we ask for a finer or coarser scale of the resulting pattern, the line integral convolution method has to be restarted with a coarser initial image. In case of spot noise larger spots have to be selected and their stretching along the field has to be increased. The methods to be presented here will incorporate different approaches to a straightforward scalability.
As already mentioned in the introduction our method of field aligned nonlinear anisotropic diffusion to visualize vector fields is derived from well known image processing methodology. Discrete diffusion type methods are known for a long time. Perona and Malik [17] have introduced a continuous diffusion model which allows the denoising of images together with the enhancing of edges. Alvarez, Guichard, Lions and Morel [1] have established a rigorous axiomatic theory of diffusive scale space methods. Kawohl and Kutev [2] investigate a qualitative analysis of the Perona and Malik model. Recovering of lower dimensional structures in images is analysed by Weikart [23], who introduced an anisotropic nonlinear diffusion method where the diffusion matrix depends on the so called structure tensor of the image. A Finite Element discretization and its convergence properties have been studied by Kačur and Mikula [12].

3 Vector Field Aligned Nonlinear Diffusion

Let us now discuss a first PDE based method. Here, nonlinear anisotropic diffusion applied to some initial random noisy image will enable an intuitive and scalable visualization of complicated flow fields. Therefore, we pick up the idea of line integral convolution, where a strong correlation in the image intensity along streamlines is achieved by convolution of an initial white noise along the streamlines. As proposed already by Cabral and Leedom [5] a suitable choice for the convolution kernel is a Gaussian kernel. On the other hand an appropriately scaled Gaussian kernel is known to be the fundamental solution of the heat equation. Thus, line integral convolution is nothing else than solving the heat equation in 1D on a streamline parametrized with respect to arclength. On pixels which are located on different integral lines the resulting image intensity is not correlated. Hence, the thickness of the resulting image patterns in line integral convolution is of the size of the random initial patterns, in general a single pixel. Increasing this size as it has been proposed by Kiu and Banks [13] leads to broader stripes and unfortunately less sharp transitions across streamline patterns. As described so far, line integral convolution is a discrete pixel based method. It can be regarded as a discretized streamline diffusion process. If we ask for a wellposed continuous diffusion problem with similar properties, we are lead to some anisotropic diffusion, now controlled by a suitable diffusion matrix.
To begin with, let us at first review the basics of the nonlinear diffusion methods in image processing. We consider a function $\rho : \mathbb{R}_0^+ \times \Omega \to \mathbb{R}$ which solves the parabolic problem

$$\begin{aligned} \tfrac{\partial}{\partial t}\rho - \operatorname{div}\left(A(\nabla\rho_\epsilon)\nabla\rho\right) &= f(\rho)\,, \quad \text{in } \mathbb{R}^+ \times \Omega\,, \\ \rho(0,\cdot) &= \rho_0 \quad , \quad \text{on } \Omega\,, \\ \tfrac{\partial}{\partial \nu}\rho &= 0 \quad , \quad \text{on } \mathbb{R}^+ \times \partial\Omega \end{aligned}$$

for given initial density $\rho_0 : \Omega \to [0,1]$. Here $\rho_\epsilon = \chi_\epsilon * \rho$ is a mollification of the current density, which will later on turn out to be necessary for the wellposedness of the above parabolic, boundary and initial value problem. In our setting we interprete the density as an image intensity, a scalar greyscale or – with a slight extension to the vector valued case – as a vector valued color. Thus, the solution $\rho(\cdot)$ can be regarded as a family of images $\{\rho(t)\}_{t\in\mathbb{R}^+}$, where the time t serves as a scaling parameter. Let us remark, that by the trivial choice $A = 1$ and $f(\rho) = 0$ we obtain the standard linear heat equation with its isotropic smoothing effect. In image processing ρ_0 is a given noisy initial image. The diffusion is supposed to be controlled by the gradient of the image intensity. Large gradients mark edges in the image, which should be enhanced. Whereas small gradients indicate area of approximately equal intensity. Here denoising, i. e. intensity diffusion is considered. For that purpose we prescribe a diffusion coefficient

$$A = G(\|\nabla\rho_\epsilon\|)$$

where $G : \mathbb{R}_0^+ \to \mathbb{R}^+$ is a monotone decreasing function with $\lim_{d\to\infty} G(d) = 0$ and $G(0) = \beta$ where $\beta \in \mathbb{R}^+$ is constant (cf. Fig. 1), e. g. $G(d) = \beta\,(1 + \|d\|^2)^{-1}$.
If we would replace the mollified gradient $\nabla\rho_\epsilon$ as argument of G by the true gradient

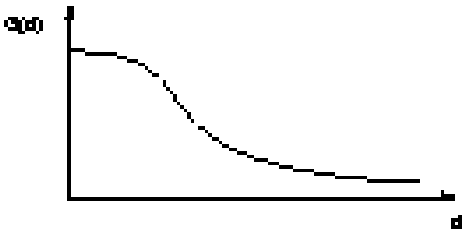

Figure 1: The shape $G(\cdot)$ which applied to the gradient of the mollified image intensity serves as a diffusion coefficient in image processing.

Figure 2: The noisy image on the left is successively smoothed by nonlinear diffusion. On the right the resulting smoothed image with enhanced edges is shown.

$\nabla\rho$, which leads to the original Perona Malik model, we would in general obtain a backward parabolic problem in areas of high gradients, which is no longer wellposed [2]. The invoked mollification avoids this shortcoming and comes along with a desirable presmoothing effect. Nevertheless the enhancing of steep gradients and thereby edges in the image, known from backward diffusion is retained if we adjust the mollification carefully. A suitable choice [12] for this mollification is a convolution with the heat equation kernel for small times, that is solving the standard diffusion problem for a short period of time. Figure 2 gives an example of such an image smoothing and edge enhancement by nonlinear diffusion. The function $f(\cdot)$ may serve as a penalty which forces the scale of images to stay close to the initial image, e. g. choosing $f(\rho) = \gamma(\rho_0 - \rho)$ where γ is a positive constant.

Now we incorporate anisotropic diffusion. For a given vector field $v : \Omega \to I\!R^n$ we consider linear diffusion in the direction of the vector field and a Perona Malik type diffusion orthogonal to the field. Let us suppose that v is continuous and $v \neq 0$ on Ω. Then there exists an family of continuous orthogonal mappings $B(v) : \Omega \to SO(n)$ such that $B(v)v = e_0$, where $\{e_i\}_{i=0,\cdots,n-1}$ is the standard base in $I\!R^n$ (cf. Fig. 3).

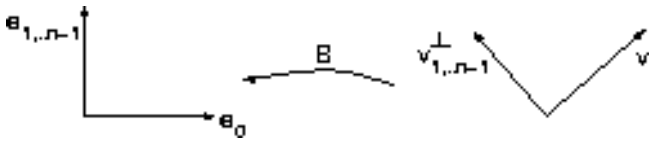

Figure 3: The coordinate transformation B(v).

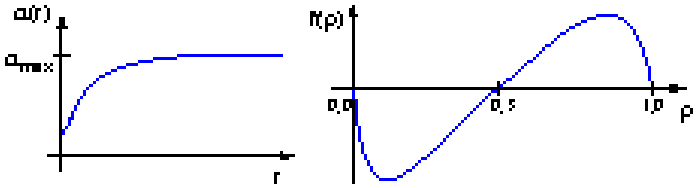

Figure 4: The graphs of the velocity dependent linear diffusion $\alpha(\cdot)$, respectively the scalar contrast enhancing right hand side $f(\cdot)$.

We consider a diffusion matrix $A = A(v, \nabla\rho_\epsilon)$ and define

$$A(v,d) = B(v)^T \begin{pmatrix} \alpha(\|v\|) & \\ & G(d) \end{pmatrix} B(v)$$

where $\alpha : I\!R^+ \to I\!R^+$ controls the linear diffusion in vector field direction, i. e. along streamlines, and the above introduced edge enhancing diffusion coefficient $G(\cdot)$ acts in the orthogonal directions. We may either choose a linear function α or in case of a velocity field, which spatially varies over several orders of magnitude, we select a monotone function α (cf. Fig. 4) with

$\alpha(0) > 0$ and

$\lim_{s\to\infty} \alpha(s) = \alpha_{\max}$.

In general it does not make sense to consider a certain initial image. As initial data ρ_0 we thus choose some random noise of an appropriate frequency range. This can for instance be generated running a linear isotropic diffusion simulation on a discrete white noise for a short time. Hence pattern will grow upstream and downstream, whereas the edges tangential to these patterns are successively enhanced. Still there is some diffusion perpendicular to the field which supplies us for evolving time with a scale of progressively coarser representation of the flow field. If we run the evolution for vanishing right hand side f the image contrast will unfortunately decrease due to the diffusion along streamlines. The asymptotic limit turns out to be a averaged grey value. Therefore, we strengthen the image contrast during the evolution, selecting an appropriate function $f : [0,1] \to I\!R^+$ (cf. Fig. 4) with

$f(0) = f(1) = 0$,

$f > 0$ on $(0.5, 1)$, and $f < 0$ on $(0, 0.5)$.

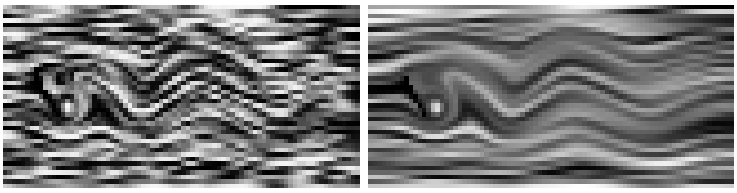

Figure 5: A single timestep is depicted from the nonlinear diffusion method applied to the vector field describing the flow around an obstacle at a fixed time. A discrete white noise is considered as initial data. We run the evolution on the left for a small and on the right for a large constant diffusion coefficient α.

Finally we end up with the method of nonlinear anisotropic diffusion to visualize complex vector fields. Thereby we solve the nonlinear parabolic problem

$$\frac{\partial}{\partial t}\rho - \operatorname{div}\left(A(v, \nabla\rho_\epsilon)\nabla\rho\right) = f(\rho)$$

starting from some random initial image ρ_0 and obtain a scale of images representing the vector field in an intuitive way (cf. Fig. 5).

If we ask for pointwise asymptotic limits of the evolution, we expect an almost everywhere convergence to $\rho(\infty, \cdot) \in \{0, 1\}$ due to the choice of the contrast enhancing function $f(\cdot)$. The space of asymptotic limits significantly influences the richness of the developing vector field aligned structures. We may ask how to further on enrich the pattern which is settled by anisotropic diffusion. This turns out to be possible by increasing the set of asymptotic states. We no longer restrict ourselves to a scalar density ρ but consider a vector valued $\rho : \Omega \to [0, 1]^m$ for some $m \geq 1$ and a corresponding system of parabolic equations. The coupling is given by the nonlinear diffusion coefficient $G(\cdot)$ which now depends on the norm $\|\nabla\rho\|$ of the Jacobian of the vector valued density $\nabla\rho$ and the right hand side $f(\cdot)$. We define

$$f(\rho) = h(\|\rho\|)\rho$$

with $h(s) = \tilde{f}(s)/s$ for $s \neq 0$, where $\tilde{f}$ is the old right hand side from the scalar case, and $h(0) = 0$. Furthermore we select an initial density which is now a discrete white noise with values in $B_1(0) \cap [0, 1]^m$. Thus the contrast enhancing now pushes the point wise vector density ρ either to the 0 or to some value on the sphere sector $S^{m-1} \cap [0, 1]^m$ in $\mathbb{R}^m$ (cf. Fig. 6). Figure 7 shows an example for the application of the vector valued anisotropic diffusion method applied to a convective flow field. An incompressible Bénard convection is simulated in a rectangular box with heating from below and cooling from above. The formation of convection rolls will lead to an exchange of temperature. Finally Figure 8 shows results of this method applied to several timesteps of the same convective flow. We recognize that the presented method is able to nicely depict the global structure of the flow field, including its saddle points, vortices, and stagnation points on the boundary. Furthermore Fig. 9 shows a different application to a porous media flow field.

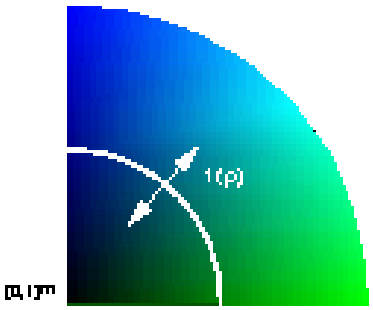

Figure 6: A sketch of the vector valued contrast enhancing function f which leads to asymptotic states $\rho(\infty, \cdot) \in \{0\} \cup (S^2 \cap [0,1]^2)$. Here, the components of the density are interpreted as blue respectively green color values. The arrows indicate the direction of contrast enhancement.

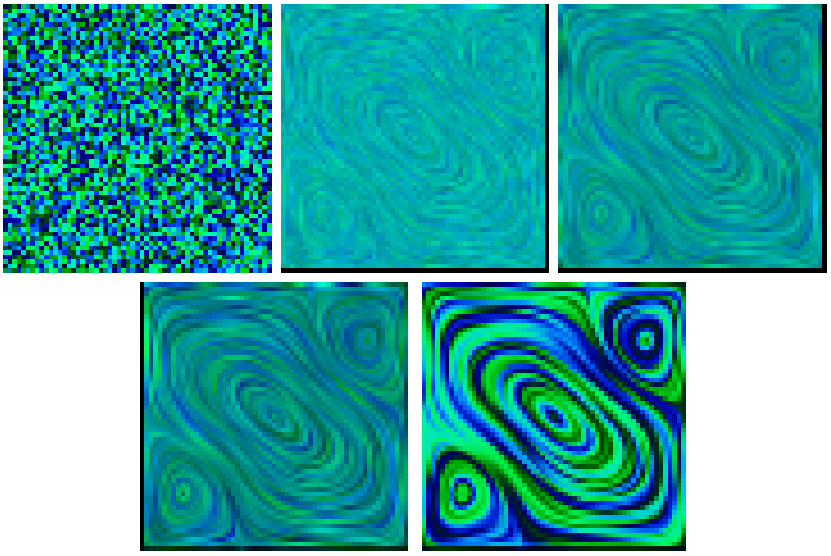

Figure 7: Several diffusion timesteps are depicted from the vector valued nonlinear anisotropic diffusion method applied to a convective flow field in a 2D box.

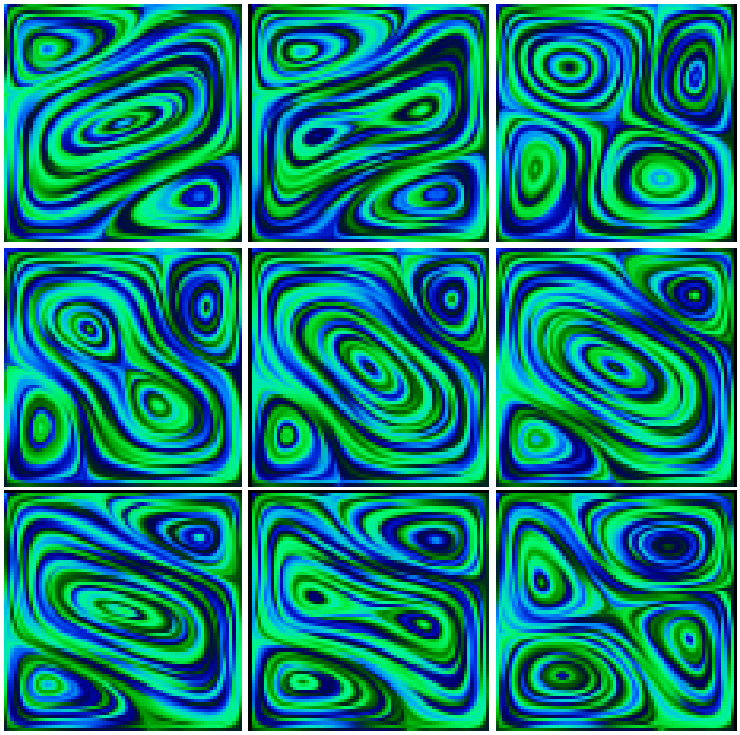

Figure 8: Convective patterns in a 2D flow field are displayed and emphasized by the method of field aligned anisotropic diffusion. The images show the velocity field of the flow at different timesteps. Thereby the resulting alignment is with respect to streamlines of this timedependent flow.

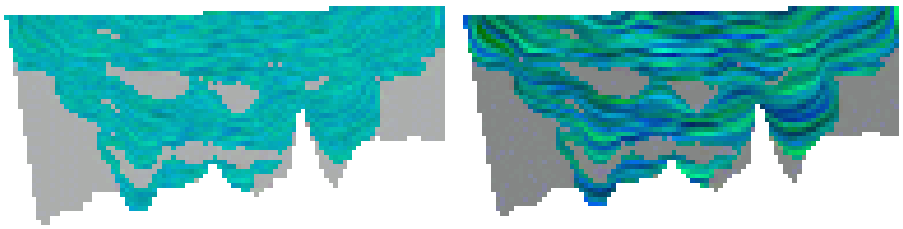

Figure 9: Field aligned diffusion clearly outlines the principle features of a porous media flow in the vicinity of a salt dome. Lenses of lower permeability force the flow to pass through narrow bridges. We depict two timesteps of the diffusion process.

4 Texture Transport Method

The second method to be presented will be based on a numerical calculation of the flux ϕ itself. We adopt the idea of the implicit streamsurfaces and discuss the corresponding transport problem for timedependent data, solve it numerically for certain boundary and initial conditions and use the result to generate an appropriate texture mapping.
At first, let us give a brief outline. At the inlet of a fluid container we prescribe inflow boundary conditions, which are the inflow coordinates, respectively the inflow time. Furthermore outflow boundary conditions are given at the outlet and slip conditions on the remaining part of the boundary. In the interior the linear transport equations with respect to the prescribed velocity $v(x,t)$ describe the fluid motion, i.e. the transport of the inflow time and inflow coordinates. The set of points in space and time which share a specific inflow coordinate coincides with a particle line, whereas the set of points with the same inflow time and inflow coordinates on a bounded surface respectively line on the inlet, describe the movement of the corresponding surface or line in time. Therefore in 2D we take the space spanned by the inflow time and the inflow coordinates as texture space and prescribe a texture with strong correlation in the direction of time. Then using the numerical results of the transport calculation, in explicit the numerical inflow time and inflow coordinates as texture coordinates, we obtain a dense representation of streak lines in terms of visible texture correlation. This representation continuously depends on time and we can easily animate the evolution. In 3D we may proceed similar as in the implicit streamsurface method proposed by van Wijk and texture the resulting streamsurfaces analogously.

Inflow time and inflow coordinates may be regarded as a Lagrangian frame. The method we propose here displays Lagrangian coordinates by texture mapping, which map a certain pattern from a Lagrangian coordinates system, i. e. from texture space, to the Eulerian frame. In more detail, let us assume $\Omega \subset I\!R^n$ to be a domain describing a fluid container with an inlet boundary $\Gamma^+ \subset \partial\Omega$ and an outlet boundary $\Gamma^- \subset \partial\Omega$. Furthermore we suppose the fluid velocity $v : \Omega \times [0, \hat{T}] \to I\!R^n$ to be given for a fixed time $\hat{T}$. In the application this velocity will be delivered by a numerical simulation, which runs simultaneously or has stored its results in files on disk. This numerical simulation is based on an additional computational grid. Therefore, to avoid some sampling procedure with its obvious drawbacks, the post processing method has to be based on the same grid (cf. Section 5).
Let us now interpret the coordinates X on the inlet boundary Γ^+, respectively the inflow time T as depending variables, which are transported with the fluid. Then they are described by the following transport equation for a density ρ

$$\begin{aligned} \partial_t \rho + v \cdot \nabla \rho &= 0 \quad && \text{in } \Omega\,, \\ \rho &= \rho_\Gamma \quad && \text{on } \Gamma^+\,, \end{aligned}$$

thereby we obtain $\rho = X$ for $\rho_\Gamma = X$ on Γ^+, respectively $\rho = T$ for $\rho_\Gamma = T$ on Γ^+. On the outlet Γ^- no boundary condition has to be described if $v \cdot \nu \geq 0$ for all times, where ν is the outer normal of the domain Ω. This transport can be interpreted as a

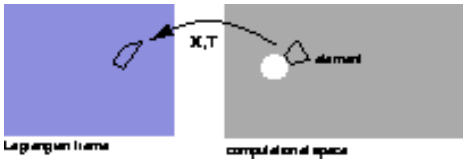

Figure 10: A sketch of the applied mapping from texture space into physical domain Ω.

simultaneous and global particle tracing. On a particle path $x(t)$ the solution ρ of the above transport equation is constant, because $\dot{x}(t) = v(x(t), t)$ and

$$\begin{aligned} \frac{d}{dt}\rho(x(t),t) &= \partial_t \rho(x(t),t) + \dot{x}(t) \cdot \nabla \rho(x(t),t) \\ &= 0\,. \end{aligned}$$

Therefore points of constant X value are located on the particle line starting at position X on Γ^+. Analogously a constant T value indicates points on a surface which is the image of a corresponding surface on the inlet under the flow $\phi(\cdot, T)$. In this sense X, T as functions on $\Omega \times [0, \hat{T}]$ can be regarded as Lagrangian coordinates describing the motion of particles which pass through Γ^+. Particles which have earlier entered the fluid container are not considered so far.
The transport equation becomes a wellposed problem by prescribing suitable initial conditions. If every particle path starting at a position in Ω has left the domain, the solution ρ no longer depends on these initial conditions. For moderate values of $\hat{T}$ this might not be the case and for certain applications especially the initial phase of the physical simulation is of great importance. Therefore we suppose that $\tilde{X}$ and $\tilde{T}$ are extensions of $X|_{\Gamma^+}$ respectively 0 on Ω and choose them as initial conditions for the two transport problems. E. g. if $\Omega \subset I\!R^+ \times I\!R$ and $\Gamma_+ \subset 0 \times I\!R$ we choose $\tilde{X}(x_1, x_2) = (0, x_1), \tilde{T}(x_1, x_2) = 0$.

Next we have to define an appropriate pattern in the texture space $\Gamma^+ \times [0, \hat{T}]$. There are several desirable features which should be realized by the textural representation of the Lagrangian coordinates. It should simultaneously code time and inlet coordinates. Furthermore to enable long time animation of moving fluids the pattern in the texture space should be periodic in T and the zooming into detailed areas has to be supported by a scalability property. Thus, we use a periodic color coding of T and a periodic scalable 1D texture for X. This 1D texture can be generated by a convolution of a white noise. For details we refer to [3].
Depending on the projection from world to screen coordinates we scale the computed Lagrangian coordinates X and T by some factor λ. If λ_0 is an initial scale which especially depends on the size of the domain $\Omega \times [0, \hat{T}]$ and $s = (\det P)^{\frac{1}{3}}$ where P is the 3×3 projection matrix describing the linear part of the affine mapping from world to image space, then $\lambda := \lambda_0\, s$ is an appropriate choice for this scaling factor. Finally we obtain as texture coordinates $\lambda X, \lambda T$ mapping points in Ω into the 2 periodic texture space $I\!R^2$ with the fundamental cell $[0, 1]^2$ which covers $\{\lambda(X(x,t), T(x,t)) \,|\, x \in$

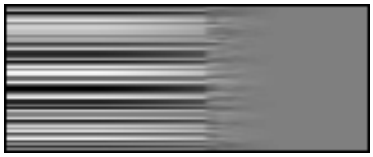

Figure 11: Fundamental cell of the texture space with error dependent blurring.

Ω, $t \in [0, \hat{T}]\}$.
Due to this construction the resulting texture on Ω at time $t \in [0, \hat{T}]$ continuously depends on t and the scaling from world space into image. Furthermore the resulting pattern is independent of this scaling. This avoids aliasing effects as long as the filter length in x direction is large enough.
Although we use a higher order Finite Volume method to solve the transport equation for a given velocity v numerically, there are unavoidable error sources. In general, especially for CFD applications, v itself is computed by some numerical algorithms, which implies approximation errors compared to the true fluid velocity in the physical application and leads to errors in the data v which we plug into the numerical transport scheme. Furthermore due to the still considerable numerical viscosity and the approximation restriction of the shape functions we obtain additional important errors contributions. Let us suppose that, by some error estimator [14] or a weaker error indicator we can measure local in space and time the resulting accumulated error. We will denote this measure $\eta(x,t)$ with $x \in \Omega$, $t \in \hat{T}$ and regard it as a function in the linear Finite Element space.
Our intention is now to use η information in the generation of the vector field images, i. e. to modify the texture correspondingly. In areas where η is small, the numerical solution of the transport equation and thereby the texturing of the domain Ω is reliable, whereas in regions with large η–values, the actual meaning of the texture is unclear and possibly makes no sense.
Therefore we first create a 2D texture $\pi(\cdot,\cdot)$ on $[0,1]^2$ with a smooth transition from $\pi(\cdot,0)$ with a clearly visible pattern - an 1D original texture for X - and a high signal bandwidth to an uniform grey valued texture $\pi(\cdot,1)$ (cf. Fig. 11 for the resulting texture).
We now consider an implication of the error indicator on the choice of the actual texture coordinates. Let us suppose that η is a function with values in $[0,1]$, where small values indicate small error bounds and values close to 1 indicates large errors and therefore small reliability of the computational results. Then we take $(\eta, \lambda X)$ as texture coordinates which map the latter introduced texture onto the computational domain. Again this texture is scalable and continuously depends on time. The following examples for different applications all use this texture for the Lagrangian X coordinate and color for the corresponding T coordinate. Thereby a simple error indicator which measures local gradients has been applied. Let us finally present some applications of the texture transport method. Figure 12 shows several timesteps of a von Kármán vortex formation in an incompressible flow behind an obstacle. In Figure 13 we pick up the formation of convection roles in a two dimensional box. We define an interior line L as an artificial

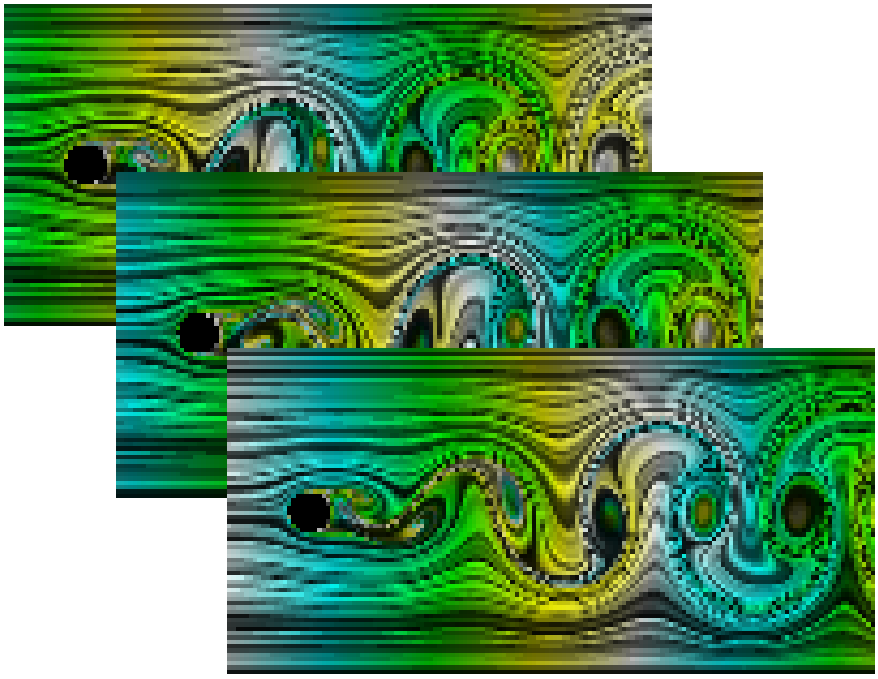

Figure 12: Texture transport in the von Kármán vortex street

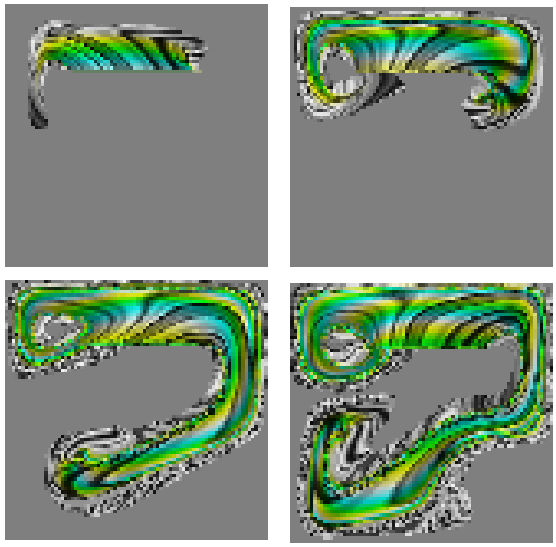

Figure 13: Texture transport applied to convection roles in an incompressible flow

boundary for the transport problem, where depending on the sign of $v \cdot \nu$ we assume outflow and inflow conditions on both sides of the line segment. Texture blurring is applied to indicate the region of reliability concerning the numerical transport solution. Furthermore Figure 14 shows the application of this method to several timesteps of a compressible flow with sub and super sonic regimes around two obstacles.
In Figure 15 we visualize the vortex formation at the contact discontinuity of a compressible flow over a forward facing step.
Finally to compare the texture transport method with the nonlinear diffusion method presented in the last section, we have depicted the visualization of a porous media flow in Figure 16.

5 Discretizations and Numerical Solution

In what follows we discuss the discretization and implementation of the two PDE based methods. The texture transport approach relies on the solution of linear hyperbolic equations, where the damping of numerical viscosity plays an essential role. The field aligned diffusion requires the solution of a system of nonlinear parabolic equations.

Texture Transport

We have already mentioned that numerical schemes for hyperbolic conservation laws are accompanied by some unavoidable numerical viscosity, which leads to a significant data mollification and a “smearing out” of the solution structure. This phenomenon is well known for shock propagation in CFD, but it already appears in case of linear trans-

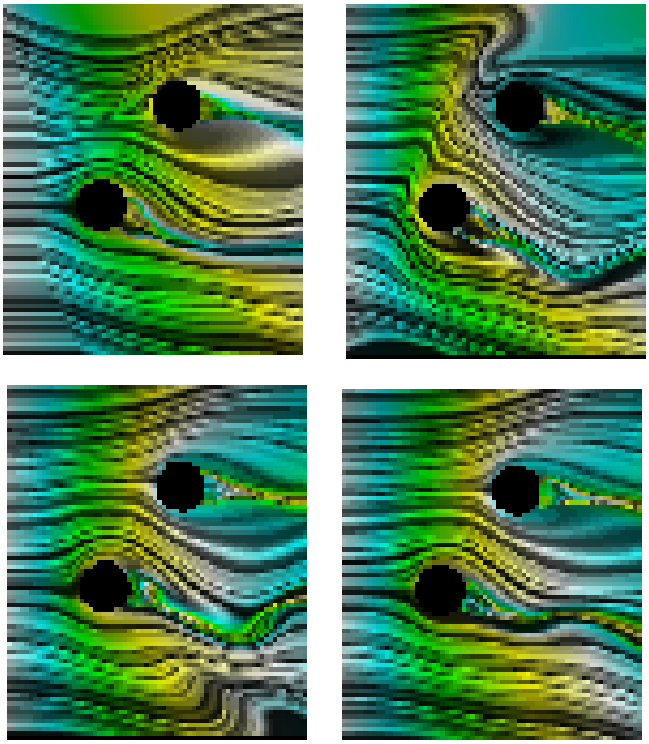

Figure 14: Texture transport applied to a compressible flow around two cylinders.

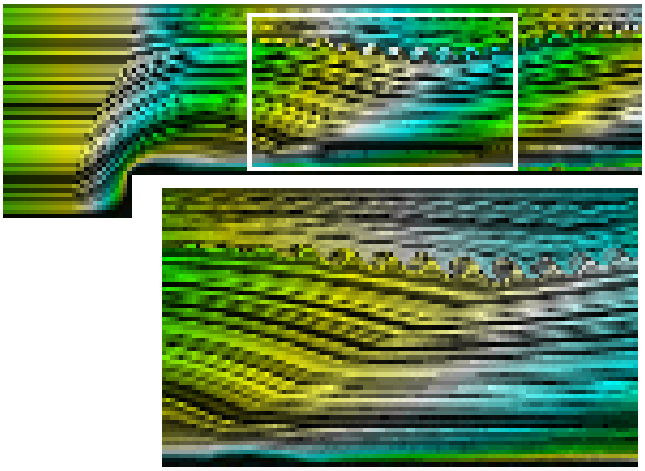

Figure 15: Texture transport is used to visualize a vortex phenomenon for a compressible flow.

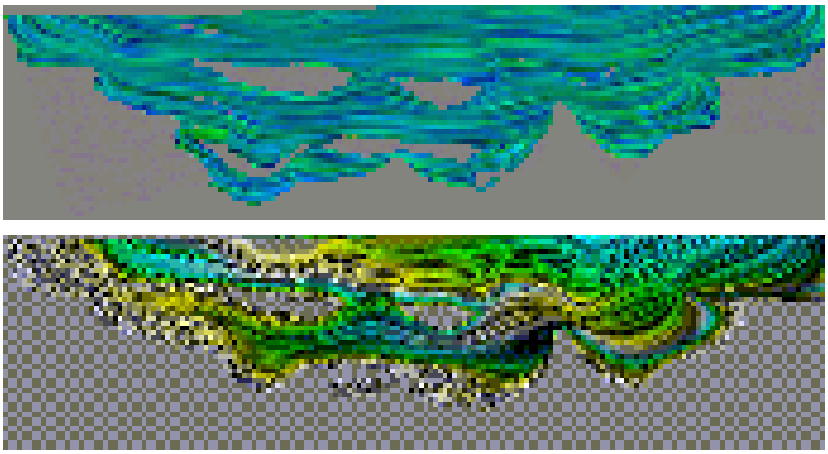

Figure 16: We compare the visualization of a porous media flow using anisotropic diffusion (on the top) and the texture transport method (on the bottom) cf. Fig. 9.

port problems. There is a trade off between the amount of this numerical viscosity and the occurance of oscillations. Especially in the current application too much numerical viscosity would destroy the evolution of interesting flow patterns presented in the numerical solution of our flow problem. Therefore, we select a higher order Discontinuous Galerkin method as an appropriate solver, with considerable smaller numerical viscosity.
The oscillations, which are well known for any type of higher order finite volume scheme, are avoided by invoking a limiting process.
Let us suppose $\mathcal{M}$ to be some structured or unstructured mesh covering the computational domain Ω and consisting of regular Elements E. On this grid we introduce the space $\mathcal{V}$ of piecewise polynomial functions, which are not required to be continuous on element faces. Then we consider the transport equation, written in conservation form

$$\frac{\partial}{\partial t}\rho + \mathrm{div} f(\rho) = \rho\,\mathrm{div} v$$

where $f : I\!R \to I\!R^n$ and $f_i(\rho) := v_i\rho$, multiply it with some $\psi \in \mathcal{V}$ and integrate over $E \in \mathcal{M}$. Applying integration by parts we obtain

$$\frac{\partial}{\partial t}\int_E \rho\,\psi + \int_{\partial E} f(\rho)\cdot\nu\,\psi = \int_E \rho\,\mathrm{div} v\,\psi + f(\rho)\nabla\psi$$

If we now require $\rho \in V \times [0,T]$ and replace the flux term $f(\rho)\cdot\nu$, which describes the flow over the faces of E by some numerical flux $g(\rho^-,\rho^+)$, where ρ^- and ρ^+ denote the piecewise polynomial, but discontinuous function ρ in E, respectively in the adjacent cells $\tilde{E}$ at the faces of E, with

$$\begin{aligned} g(\rho,\rho) &= f(\rho)\cdot\nu\,, \\ g(\rho^-,\rho^+) &= -g(\rho^+,\rho^-)\,, \end{aligned}$$

we obtain the semidiscrete Discontinuous Galerkin method. The Engquist-Osher flux [10] is used in the current texture transport algorithm. Finally we discretize this by some Runge Kutta scheme in time and to avoid oscillations combine the resulting algorithm with a limiter which cut off local extrema after each Runge Kutta iteration step. For a detailed discussion on the Discontinuous Galerkin Method we refer to Cockburn et. al. [6, 8, 7]. In our implementation we approximate $\rho = T, X$ on each volume E by a linear function. Let us emphasize that we obtain a standard first order Finite Volume scheme, if we take into account piecewise constant shape functions in space and a forward Euler scheme in time.

Field Aligned Diffusion

The method of field aligned diffusion is implemented based on a Finite Element discretization in space and a semi implicit backward Euler scheme in time. We have restricted ourselves to regular grids generating by recursive subdivision. On these grids we consider the bilinear, respectively trilinear Finite Element spaces. Numerical integration is based on the simplest quadrature rule, which evaluates the integrant solely at the element's center point. These integration formulas are applied to compute the local stiffness matrixes. The semi implicit character of our scheme means, that the nonlinearity $G(\nabla \rho_\epsilon)$ is evaluated at the old time. In each timestep the computation of ρ_ϵ is based on a single short implicit timestep for the corresponding heat equation with respect to initial data ρ. We take into account mass lumping to calculate the local mass matrixes. The regular grids are procedurally interpreted as quadtree, respectively octtree. Finally no matrix is explicitly stored. The necessary matrix multiplications in the applied iterative CG solver is performed in successive tree traversals. Hierarchical BPX type [4] preconditioning is applied to accelerate the convergence of the linear solver. A single timestep on a 256^2 grids performed on a Silicon Graphics workstation with an R10000 processor at 195 MHz requires ca. 1.2 secs. The code is prepared to incorporate spatial grid adaptivity if possible.

6 Comparison to Other Methods

So far we have introduced the two methods of texture transport and field aligned diffusion to ensure an intuitive understanding of complex flow fields. We have discussed a variety of important properties and advantages. Let us now rank them among other visualization methods and compare them with different techniques. Here we especially pick up the line integral convolution method and the spot noise approach.
Given a possible timedependent vector field $v(x,t)$ there are two basic and well known tools in flow visualization. For a fixed point $x \in \bar{\Omega}$ we can consider streamlines, that is for fixed time t we consider a curve $\{x(s) | s \in I\!R\}$ with

$$\frac{\partial}{\partial s} x(s) = v(x(s), t)\,, \quad x(0) = x\,.$$

Alternatively we may examine a family of curves $\{x_s(t)|t \in I\!R\}$ with

$$\frac{\partial}{\partial t} x_s(t) = v(x_s(t), t)\,, \quad x_s(s) = x\,.$$

For fixed t the curve $\{x_s(t)|s \in I\!R\}$ is a streakline. In an obvious sense the method of field aligned diffusion comes up with a dense coverage of the domain Ω with streamlines, whereas the texture transport method results in a dense coverage of Ω with streaklines. If we fix the texture in the Lagrangian coordinates for the texture transport, respectively an initial intensity with moderate frequency range for the field aligned diffusion, both methods will lead to a vector field representation which is continuous in time. I. e. resulting images of close timesteps will strongly correlate.
For stationary flow fields we obtain similar results by both methods. Thin flow aligned patterns are generated. Line integral convolution leads to comparable results with the essential difference that the PDE based methods carry a nice scale space property. I. e. either selecting a coarser the texture in the Lagrangian frame or evolving a longer time in the anisotropic diffusion method we obtain a successive coarsening of the resulting flow representation.
Furthermore in a restricted sense, line integral convolution (LIC) and spot noise can be regarded as special cases of the field aligned diffusion method. LIC with Gaussian filter kernel can be identified as the asymptotic limit of the latter method of a concentration of the edge enhancing function $G(\cdot)$ at 0. Other filter kernel shapes correspond to different, in general non linear diffusion processes along streamlines. Further on, generating a single deformed spot on the computational domain like proposed in [9] can be regarded as an early timestep in the diffusion starting with initial data, that is a characteristic function of a circular disk. If we release a bunch of such disks as initial data in such a way that the evolving patterns do not overlap, then the resulting image is comparable to spot noise. Thus, spot noise can be regarded as a parallel version of short time diffusive vector field visualization.

7 Conclusions

We have introduced two novel method for the post processing of flow data. From a mathematical point of view one of their major advantages is, that they are based on physically intuitive continuous models, i. .e transport and diffusion. Most of their properties can be discussed on this level. Finally they are discretized in an appropriate way making use of recent and efficient numerical algorithms. This clearly indicates the strong interdependence of scientific computing and scientific visualization methods. The intention of this paper is especially to outline one interesting facet of this interplay. From the author's point of view exciting future research directions are further investigations of flow visualization in 3D by means of texture transport and field aligned diffusion. Furthermore, a visualization approach based on anisotropic diffusion and applicable for timedependent vector fields is a challenging topic.
Further results and the algorithm running on an $n \times m$ 2D vector array is avaliable as a source code at the URL:

http://www.iam.uni-bonn.de/SFB/visual/nonlinear.diffusion

Acknowledgement

The authors would like to acknowledge Karol Mikula and Jarke van Wijk for inspiring discussions and many useful comments on flow visualization and image processing. Furthermore they thank Frank Koster from Bonn University, Eberhard Bänsch from Bremen University and Klaus Johannsen from Stuttgart University for providing the incompressible flow data sets, respectively the porous media simulation data.

References

[1] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Arch. Ration. Mech. Anal.*, 123 (3):199–257, 1993.

[2] B. Kawohl and N. Kutev. Maximum and comparison principle for one-dimensional anisotropic diffusion. *Math. Ann.*, 311 (1):107–123, 1998.

[3] J. Becker and M. Rumpf. Visualization of time-dependent velocity fields by texture transport. In *Proceedings of the Eurographics Scientific Visualization Workshop '98*. Springer, 1998.

[4] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55:1–22, 1990.

[5] B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 263–272, Aug. 1993.

[6] G. Chavent and B. Cockburn. The local projection p0-p1-discontinuous-galerkin finite element method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 23:565–592, 1989.

[7] B. Cockburn and C.-W. Hou, S.and Shu. Tvb runge-kutta local projection discontinuous-galerkin finite element method for conservation laws iv: The multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.

[8] B. Cockburn and C.-W. Shu. Tvb runge-kutta local projection discontinuous-galerkin finite element method for conservation laws ii: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[9] W. C. de Leeuw and J. J. van Wijk. Enhanced spot noise for vector field visualization. In *Proceedings Visualization '95*, 1995.

[10] B. Engquist and S. Osher. One sided difference approximations for nonlinear conservation laws. *Math. of Comp.*, 36:321–351, 1981.

[11] V. Interrante and C. Grosch. Stragegies for effectively visualizing 3d flow with volume lic. In *Proceedings Visualization '97*, pages 285–292, 1997.

[12] J. Kačur and K. Mikula. Solution of nonlinear diffusion appearing in image smoothing and edge detection. *Appl. Numer. Math.*, 17 (1):47–59, 1995.

[13] M.-H. Kiu and D. C. Banks. Multi-frequency noise for lic. In *Proceedings Visualization '96*, 1996.

[14] D. Kröner and M. Ohlberger. A-posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions. *Preprint, Mathematische Fakultät, Albert-Ludwigs-Universität Freiburg*, 1998.

[15] N. Max and B. Becker. Flow visualization using moving textures. In *Proceedings of the ICASE/LaRC Symposium on Time Varying Data, NASA Conference Publication 3321*, pages 77–87, 1996.

[16] N. Max, R. Crawfis, and C. Grant. Visualizing 3D Velocity Fields Near Contour Surface. In *Proceedings of IEEE Visualization '94*, pages 248–254, 1994.

[17] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. 1987.

[18] H.-W. Shen and D. L. Kao. Uflic: A line integral convolution algorithm for visualizing unsteady flows. In *Proceedings Visualization '97*, pages 317–322, 1997.

[19] D. Stalling and H.-C. Hege. Fast and resolution independent line integral convolution. In *SIGGRAPH 95 Conference Proceedings*, pages 249–256. ACM SIGGRAPH, Addison Wesley, Aug. 1995.

[20] G. Turk and D. Banks. Image-guided streamline placement. In *Proc. 23rd annual conference on Computer graphics, August 4 - 9, 1996, New Orleans, LA USA*. ACM Press, 1996.

[21] J. J. van Wijk. Spot noise-texture synthesis for data visualization. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 309–318, July 1991.

[22] J. J. van Wijk. Flow visualization with surface particles. *IEEE Computer Graphics and Applications*, 13(4):18–24, July 1993.

[23] J. Weickert. *Anisotropic diffusion in image processing*. Teubner, 1998.