Visualization of Time-Dependent Velocity Fields by Texture Transport

Joachim Becker[1] and Martin Rumpf[2]

[1] Institute for Applied Mathematics , Freiburg University , Germany
[2] Institute for Applied Mathematics , Bonn University , Germany

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, which should be closely related to the physical interpretation. A new tool, the texture transport method is presented, which especially applies to time-dependent velocity fields. It is based on an accurate numerical scheme for convection equations, which is used to compute Lagrangian coordinates in space time. These coordinates are then used as texture coordinates referring to some prescribed texture in the Lagrangian reference space. The method is combined with a reliability indicator. This indicator influences the final appearance of the texture and thereby leads to reliable visual information. At first the method applies to 2D problems. It can be generalized to 3D.

1 Introduction

The visualization of field data, especially of velocity fields from CFD computations is one of the fundamental task 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 receptable method. Furthermore it should be closely related to the mathematical meaning of field data, which is mainly expresses 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, and for simplicity we at first assume that $v \cdot \nu = 0$ where ν is the outer normal on $\partial\Omega$ (lateron we will define corresponding in- and outflow conditions in a Lagrangian frame) 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 $\phi(x,0) = x$.
The spot noise method proposed by van Wijk [22] 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 [6], where they use higher order polynomial deformation of the spots in areas of significant vorticity. The Line Integral Convolution (LIC) approach of Cabral and Leedom [2] 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 streamline 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. [12] proposed a similar method on surfaces. Max and Becker [13] present a method for visualizing 2D and 3D flows by animating textures. Turk [21] discusses an approach by which a certain number of streamlines is automaticly equally distributed on the computational domain.

Especially for 3D velocity fields particle tracing is a very popular tool. But a few particle integrations released by the user can hardly scope with the complexity of 3D vector fields. Zöckler et al. [20] use pseudo randomly distributed and illuminated and transparent streamlines to give a denser and receptable representation, which shows the overall structure and enhances important details.
An effective method to calculate stream surfaces in 3D, which nicely depicts separation phenomena has been presented by Hultquist [9]. Van Wijk [23] proposed the implicit stream surface method. For a stationary flow field the transport equations $v \cdot \nabla\phi = 0$ are solved for given v and certain inflow and outflow boundary conditions in a precomputing step. Then isosurfaces of the resulting function ϕ are streamsurfaces and can efficiently be extracted with interactive frame rates even for larger data sets.

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 time-dependent 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. Therefore we ask for an approach using texture based methods as well-suited tools to ensure a overall representation of field data, which avoids the above drawback in the non stationary case. We adopt the idea of the implicit streamsurfaces and discuss the corresponding transport problem for time-dependent data, solve it numerically for certain boundary and initial conditions and use the result to generate an appropriate texture mapping.
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 shares 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 particle lines in terms of visible texture correlation. This representation continuously depends on time and we can easily animate the evolution. In 3D we proceed similar as in the implicit streamsurface method proposed by van Wijk and texture the resulting streamsurfaces analogously.

The paper is organized as follows. In Section 2 we will in detail explain the continuous transport problem and the related coordinate systems. The numerical scheme and especially its improvement by higher order shape functions is discussed in section 4, whereas in Section 3 we deal with the question of reliability and propose a method to represent this adequately in the resulting images. Furthermore in Section 6 we briefly give first results in 3D. Finally we draw conclusion and outline future research directions.

2 Lagrangian coordinates and transport equations

Velocity fields in numerical simulations are mostly given in the spatially fixed Eulerian coordinate system, whereas its physical meaning in terms of moving fluid particles is more closely related to the Lagrangian frame. This observation is the starting point of various visualization techniques. The method we propose here displays Lagrangian coordinates using a texture mapping, which map a certain pattern from a Lagrangian coordinates system to the Eulerian frame. To start with, let us assume $\Omega \subset I\!R^2$ 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 $v : \Omega \times [0, \hat{T}] \to I\!R^2$ 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 4).
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

$$\begin{aligned} \partial_t \rho + v \cdot \nabla \rho = 0 \quad & \text{in } \Omega\,, \\ \rho = \rho_\Gamma \quad & \text{on } \Gamma^+\,, \end{aligned} \tag{1}$$

thereby we obtain $\rho = X$ for $\rho_{\Gamma} = X$ on Γ^+, respectively $\rho = T$ for $\rho_{\Gamma} = T$ on Γ^+. At 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 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

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

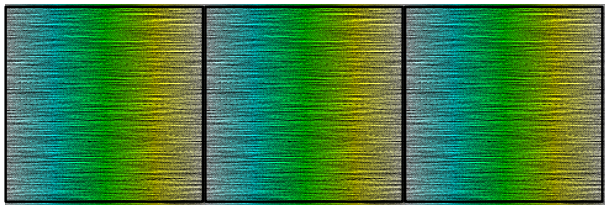

Fig. 1. LIC convolution along the T component.

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 well–posed problem by prescribing suitable initial conditions. If every particle paths 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 $\Omega \subset \mathbb{R}^+ \times \mathbb{R}$ and $\Gamma_+ \subset 0 \times \mathbb{R}$ we choose $\tilde{X}(x_1, x_2) = (0, x_1)$, $\tilde{T}(x_1, x_2) = 0$.
In Section 4 we will discuss a numerical algorithm to compute an approximation of the transport solution and thereby of the Lagrangian coordinates.
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 scaleability property. These requirements are fulfilled by the following construction:

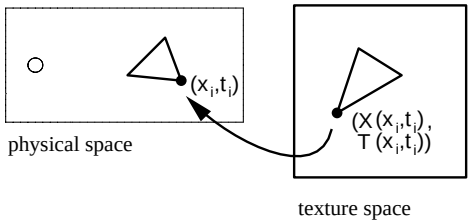

Fig. 2. A sketch of the applied mapping from texture space into physical domain Ω.

- Choose some white noise on a rastered domain $[0,1]^2$, those coordinates are denoted x, t corresponding to the Lagrangian coordinates X, T and duplicate this domain three times shifting it in the t-direction by -1,0 and 1 (compare Fig. 1).
- Then use the LIC type convolution along the T component with a filter length smaller than 1 (compare Fig. 1).
- If a smoothing in x is intended repeat the same duplication and convolution in x direction with a second filter length. This mollification scale should be significantly smaller in order not to destroy the correct perception of the flow direction.
- Thereby we obtain a texture on the original domain $[0,1]^2$ with a previously fixed rasterization. By periodic shifting in both directions we finally obtain a 2 periodic texture.
- 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 $\mathbb{R}^2$ with the fundamental cell $[0,1]^2$ which covers $\{\lambda(X(x,t), T(x,t)) \,|\, x \in \Omega,\, t \in [0, \hat{T}]\}$ (compare Fig. 1).

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.
Finally one degree of freedom is still left in the generation of an image. We can code by coloring a second scalar physical quantity, i. e. pressure. Alternatively color can be used to accentuate the motion on the streakline pattern. Therefore in addition a time periodic coloring of the greyscale texture is applied. In that case the T component of the Lagrangian coordinates is represented twice, by the periodic structure of the texture in T direction and by the coloring. If we disclaim the first we can abandon the T component of the texture to a reliability quantification of the numerical transport results. We will focus on this important aspect in the next paragraph.

3 Texture visualization and reliability

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 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 [10, 18, 11] 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. 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 possible makes no sense.

Therefore we first create a texture π with a smooth transition from clearly visible pattern with a high signal bandwidth to an uniform grey valued texture. As already explained, if we code the Lagrangian coordinate T solely by color, the corresponding t texture component is no longer needed. Then we are able to parameterize the above transition over $t \in [0,1]$. Let us suppose that the current one dimensional texture $\pi(\cdot, 0)$ is periodic with its fundamental cell $[0,1]$, i. e. a white noise $\alpha(\cdot)$ on $[0,1]$ periodicly expanded on $\mathbb{R}$ and finally convoluted by some block filter kernel $\chi_\epsilon(\cdot)$ with support ϵ. Here χ_ϵ denotes the characteristic function on $[-\epsilon, \epsilon]$. Then we have two methods at hand to define the required transition.

- We can expand the support of the filter kernel from ϵ at $t = 0$ to 1 at $t = 0$. In detail we define the texture at $t \in [0,1]$ by $\pi(x,t) := \chi_{\epsilon(t)} * \alpha$ where $\epsilon(t)$ is a monotone function on $[0,1]$ with $\epsilon(0) = \epsilon$ and $\epsilon(1) = 1$.
- Alternatively we can successively decrease the texture signal's amplitude. I. e. for given $\pi(\cdot, 0)$ and $\bar{\pi} = \int_0^1 \pi(x,0)dx$ define

$$\pi(x,t) = (1 - \beta(t))\pi(x,0) + \beta(t)\bar{\pi}$$

  where β is a monotone increasing function on $[0,1]$ with $\beta(0) = 0$ and $\beta(1) = 1$. In particular a spline β with vanishing derivatives at 0 and 1 has proved useful in the applications.

Finally these two methods can be combined by concatenation of the two operators (cf. Fig. 3 for the resulting texture).
With this new parameter family of one dimensional texture spaces at hand we now consider the implications 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 closed to 1 large errors and therefore small reliability of the computational results. Then we take $(\lambda X, \eta)$ 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.

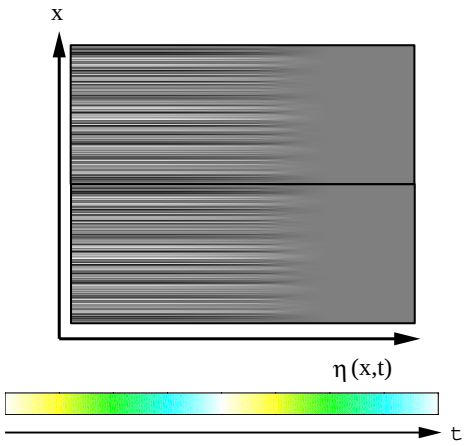

Fig. 3. Fundamental cell of the texture space with error dependent blurring and the periodic color ramp for the coding of time.

4 Higher order numerical transport scheme

Numerical schemes for hyperbolic conservation laws are accompanied by some numerical viscosity, which leads to a significant data mollification and a "smearing out" of the solution structure. This phenomena is well-known for shock propagation in CFD, but it already appears in case of linear transport problems. There is a trade off between the amount of this numerical viscosity and the occurring of oscillations. Especially in the current application to much numerical viscosity would destroy the evolution of interesting flow patterns represented in the numerical solution of our flow problem. Therefore, after some first testing we reject the usage of standard first order Finite Volume schemes and choose the 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 unstructured mesh covering the computational domain Ω and consisting of regular Elements E_i for i in some index set $I_{\mathcal{M}}$. On this grid we introduce the space $\mathcal{V}$ of piecewise polynomial function, which are not required to be continuous on element faces. Then we consider the transport equation (1), 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 $E \in \mathcal{M}$. Thereby we obtain

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

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 [7] 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. [3–5]. In our implementation we approximate $\rho = T, X$ on each volume E by a linear function. Let us emphasize that we obtain standard first order Finite Volume schemes if we take into account piecewise constant shape functions in space and a forward Euler scheme in time.

5 Examples in 2D

At first we considered an incompressible flow around a circular obstacle in a rectangular channel. At moderate Reynolds numbers we expect the Kármán vortex flow pattern. Here the numerical velocity v is calculated by a mixed Finite Element method with quadratic shape functions for v [1]. To resolve the

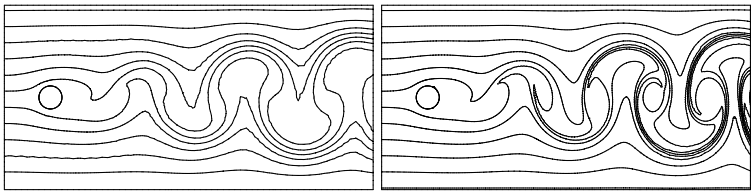

Fig. 4. A comparison between first and higher order method for the numerical transport of the X component using isolines on the physical domain Ω at a certain time.

approximation quality also in the numerical solver for the transport problem we refine the triangular Finite Element mesh, subdividing each triangle into 16 smaller triangles, and then start the second order Discontinuous Galerkin method to calculate the X, T coordinates. Several figures above already reflect the obtained results. Fig. 6 depicts several timesteps from the evolution of flow in time and Fig. 7 underlines the scalability of the texture for several different magnification factors. The zooming region is outlined in black in the original full image.
Finally we compute and display the texture transport for a compressible velocity field given by the numerical solution of the 2D Euler equations. Two obstacles are placed in a channel and we increase the prescribed inflow velocity successively in time. Fig. 8 compares the induced flow pattern at different times.

6 Vector fields in 3D

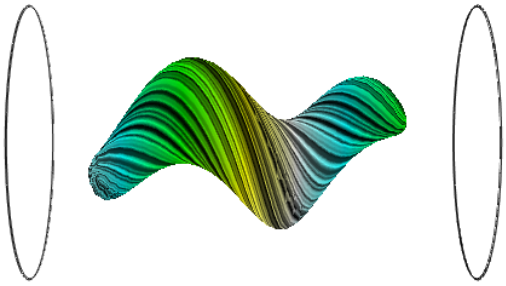

Fig. 5. 3D texture transport

The method of the Lagrangian coordinate transport can obviously be transfered to the three dimensional case. Thereby we especially compute the transport of two dimensional inlet coordinates $X \in \Gamma^+$. For a visualization of the results in terms of texture rendering, we pick up the implicit streamsurface idea presented by J. Wijk [23]. Consider implicitly defined curves $\gamma = \{X \in \Gamma^+ \,|\, g(X) = 0\}$ for

some regular function g on Γ^+. Let us denote by $s : [0,1] \to \Gamma^+$ a parameterization of γ, which is supposed to be periodic if γ is closed. Then the image $X(\gamma, \cdot)$ of γ under the coordinate map X is a streamsurface. This surface can be extracted on the discrete grid by any discrete isosurface algorithm. With respect to the parameterization s it can be textured over the same texture space, already used in 2D applications. If we furthermore consider a family of implicit parameterized curves, we also obtain a continuous transition in the texture images concerning continuous modifications of this parameter in an interactive exploration. Instead of implicit curves on Γ^+ we can also ask for the images of implicit surfaces on $\Gamma^+ \times [0, \hat{T}]$ and texture them correspondingly. Fig. 5 shows a first picture of such a surface deformation by the Lagrangian coordinate mapping. Therefore an ellipsoid has been prescribed on $\Gamma^+ \times [0, \hat{T}]$. The underlying velocity is a test data set on a cylindrical domain.

7 Conclusions

A new method for the visualization of vector field data has been presented. It applies to stationary and time dependent data in 2D and combined with the implicit streamsurface method of van Wijk it has strong provisions for the three dimensional case as well. Based on the numerical solution of the transport equation for the Lagrangian coordinates (related to the inflow boundary) texture coordinates are calculated which map a pattern in the Lagrangian coordinate space onto the computational domain. The resulting pattern shows a strong alignment in the direction of particle paths and can be animated in time. The method is computationally expensive concerning the numerical solution of the transport problem, which may run in parallel to the actual numerical flow simulation or afterwards in a preparatory step for the post processing. Compared to this the actual post processing is fast and interactive especially on machines with hardware texturing. Future research will be on the distributed calculation of transport and the efficient extraction of well suited texture patterns in 3D. Here we will combine the proposed method with multilevel visualization techniques [14, 16, 15]. The authors thank E. Bänsch for providing the numerical data of the von Kármán vortex street.
This paper is a part of the PhD-thesis one author (J. Becker) is working on.

References

1. Bänsch, E.: Simulation of instationary, incompressible flows, Submitted to Acta Math. Univ. Comenianae.
2. Cabral, B.; Leedom, L.: Imaging Vector Field Using Line Integral Convolution, Computer Graphics Proceedings, Annual Conference Series 1993.
3. Chavent, G.; Cockburn, B.: The Local Projection P0-P1-Discontinuous-Galerkin Finite Element Method For Scalar Conservation Laws, Mathematical Modelling and Numerical Analysis Vol.23,N 4,1989,p.565-592.

4. Cockburn, B.; Shu, C.-W.: TVB Runge-Kutta Local Projection Discontinuous-Galerkin Finite Element Method For Conservation Laws II: General Framework, Mathematics of Computation, Vol.52,Nu.186,1989,p.411-435.
5. Cockburn, B.; Hou, S.; Shu, C.-W.: TVB Runge-Kutta Local Projection Discontinuous-Galerkin Finite Element Method For Conservation Laws IV: The Multidimensional Case, Mathematics of Computation, Vol.54,Nu.190,1990,p.545-581.
6. de Leeuw, W. C.; van Wijk, J. J.: Enhanced Spot Noise for Vector Field Visualization, presented at Visualization '95,Atlanta.
7. Engquist, B., Osher, S.: One sided difference approximations for nonlinear conservation laws. Math. of Comp. 36 (1981), 321-351.
8. Forssell, L. K.: Visualizing Flow Over Curvilinear Grid Surfaces Using Line Integral Convolution, IEEE Visualization '94,240–246, 1995.
9. Hultquist,J. P. M.: Interactive Numerical Flow Visualization Using Stream Surfaces. Computing Systems in Engineering, Vol.1, No.2-4, 1990, pp.349–353.
10. Kröner, D.: Numerical Schemes for Conservation Laws, Wiley Teubner.
11. Kröner, D.; Ohlberger, M.: A-posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions. Preprint, Mathematische Fakultät, Albert-Ludwigs-Universität Freiburg. 1998.
12. Max, N.; Crawfis, R.; Grant, C.: Visualizing 3D Velocity Fields Near Contour Surfaces, IEEE Visualization '94,248–254, 1994.
13. Max, N.; Becker, B.: Flow Visualization using Moving Textures, Proceedings of the ICASE/LaRC Symposium on Time Varying Data, NASA Conference Publication 3321, D. C. Banks, T. W. Crocket, and K. Stacy, editors,(1996) pp. 77 - 87.
14. Neubauer, R.; Ohlberger, M.; Rumpf, M.; Schwörer, R.: Efficient Visualization of Large–Scale Data on Hierarchical Meshes. Lefer, W. and Grave, M., Visualization in Scientific Computing, 1997, Springer.
15. Ohlberger, M.; Rumpf, M.: Adaptive Projection Methods in Multiresolutional Scientific Visualization. Report 20, Sonderforschungsbereich 256, Bonn, 1998.
16. Ohlberger, M.; Rumpf, M.: Hierarchical and Adaptive Visualization on Nested Grids. Computing. Vol.59 (4), pp.269-285, 1997.
17. SFB 256, University of Bonn: GRAPE manual, http://www.iam.uni-bonn.de/main.html, Bonn 1995.
18. Sonar, T.; Süli, E.: A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. Oxford University,Report 94/9 , 1994.
19. Stalling, D.; Hege C.: Fast and Resolution Independent Line Integral Convolution, Proceedings SIGGRAPH '95, 1995.
20. Stalling, D.; Zöckler; M.; Hege, H.-C.: Fast Display of Illuminated Field Lines. IEEE Transactions on Visualization and Computer Graphics, No. 2 Vol.3 1997.
21. Turk, G.: Re-tiling polygonal surfaces. Computer Graphics (SIGGRAPH '92 Proceedings) Vol.26 1992,55–64.
22. van Wijk, J. J.: Spot Noise – Texture Synthesis for Data Visualization, Computer Graphics, Volume 25, Number 4, 1991.
23. van Wijk, J. J.: Implicit Stream Surfaces, IEEE Visualization '93,245–252, 1993.
24. van Wijk, J. J.: Flow Visualization with Surface Particles. IEEE Computer Graphics and Applications Vol. 13, No.4, 1993 , pp.18–24.

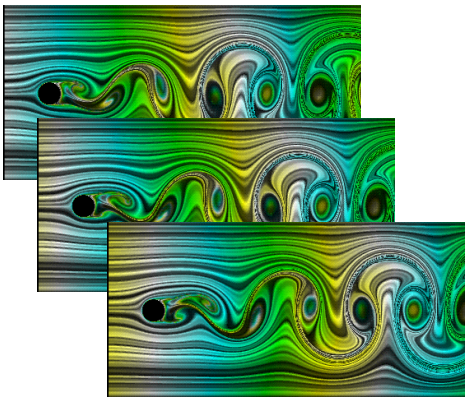

Fig. 6. Texture transport in the von Kármán vortex street.

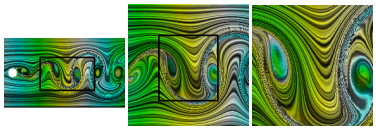

Fig. 7. Several intermediate steps in a continuous zoom into the physical space Ω.

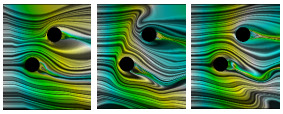

Fig. 8. Texture transport applied to a compressible flow arround two cylinders.