Stereographic Visualization of Turbulent Pipe Flows Using Anaglyphs with a Twofold Central Projection

Wolfgang Meßner, Walter Huber

messner, huberw@informatik.tu-muenchen.de

Technische Universität München, Institut für Informatik,
Lehrstuhl für Ingenieuranwendungen in der Informatik,
D-80290 München, Germany

Abstract

The size of velocity-vector fields obtained from numerical solutions of the Navier-Stokes equations for three-dimensional, time varying and turbulent flows is extremely large. Stereographic systems can intuitively depict the small depth movements and relations between fluid particles that standard perspective projections fail to. As expensive high-end-stereographic systems aim at one user one computer and are therefore not suitable for demonstration purposes in lectures, the anaglyphic method is applied for visualizing turbulence in a stereographic movie to a large number of people. A solution to overcome the effect of ghosting, which has been a major drawback in the use of anaglyphs, is introduced. A twofold central projection was found to be superior to standard on-axis projections in graphics hardware allowing an easy representation of objects in front of and behind the projection plane and thus enhancing the stereoscopic effect. The visualization system deals with the small fluctuations of the velocity about its mean values (i.e. the so called rms values) to depict real turbulence in a pipe flow at reynolds number $Re \approx 7000$.

1 Introduction

Even for experts to the field of flow simulation the data obtained by numerical simulations (see paragraph 1.1) are difficult to understand. For a graphical visualization of these data special techniques (see [van Wijk 93] for example) can be used – even within a virtual environment (see [Bryson 92, 93]). For introducing non-experts on the other hand to the field of turbulence a completely different approach is needed. The applied visualization techniques must be as intuitive as possible (see paragraph 1.2) and the animation should be practicable for conferences and lectures as well as for use on a single workstation. For combining those demands with a stereographic environment we have been using anaglyphs. In paragraph 2.2 we will be discussing techniques for overcoming the effect of ghosting and for computing both eye views. Paragraph 3.2 describes a technique for depicting real turbulence in a pipe flow at reynolds number 7000 by means of the rms values. Our visualization system has been implemented on a Silicon Graphics Indy Workstation with 64 MB main memory using C as a programming language together with the graphics library GL.

1.1 Numerical Background

In the middle of the last century C.-L. Navier and G.G. Stokes developed the mathematical and physical theory for describing turbulent and laminar flows. But only since the 1960's scientists accomplish to model the complexity of turbulent, three-dimensional and time varying pipe flows with the help of supercomputers. The *Navier-Stokes equations* (2) for z, φ and r-directions together with the *continuity equation* (1) describe laminar as well as turbulent flows. Their differential form is given here using cylinder coordinates ([Unger 94]).

$$\frac{\partial u_z}{\partial z} + \frac{1}{r}\frac{\partial u_\varphi}{\partial \varphi} + \frac{1}{r}\frac{\partial}{\partial r}(u_r r) = 0 \tag{1}$$

$$\begin{aligned}
&\frac{\partial u_z}{\partial t} + u_z\frac{\partial u_z}{\partial z} + \frac{1}{r}u_\varphi\frac{\partial u_z}{\partial \varphi} + u_r\frac{\partial u_z}{\partial r} = -\frac{\partial p}{\partial z} + \frac{1}{Re}\Delta u_z \\
&\frac{\partial u_\varphi}{\partial t} + u_z\frac{\partial u_\varphi}{\partial z} + \frac{1}{r}\left(u_\varphi\frac{\partial u_\varphi}{\partial \varphi} + u_r u_\varphi\right) + u_r\frac{\partial u_\varphi}{\partial r} \\
&\quad = -\frac{1}{r}\frac{\partial p}{\partial \varphi} + \frac{1}{Re}\left[\Delta u_\varphi - \frac{1}{r^2}\left(u_\varphi - 2\frac{\partial u_r}{\partial \varphi}\right)\right] \\
&\frac{\partial u_r}{\partial t} + u_z\frac{\partial u_r}{\partial z} + \frac{1}{r}\left(u_\varphi\frac{\partial u_r}{\partial \varphi} - u_\varphi^2\right) + u_r\frac{\partial u_r}{\partial r} \\
&\quad = -\frac{\partial p}{\partial r} + \frac{1}{Re}\left[\Delta u_r - \frac{1}{r^2}\left(u_r + 2\frac{\partial u_\varphi}{\partial \varphi}\right)\right].
\end{aligned} \tag{2}$$

On a Cray Y-MP it takes for example 32 Megawords to solve these equations for a pipe resolution of 3 million grid points. Scientific visualization systems have to store velocity-vector fields up to some Gigabytes ([Woodward 92]). As our intention was a visualization system for educational and demonstrational purposes making use of no additional expensive hardware our data has about 40 Megabytes in size for each timestep. A projection onto a smaller grid resolution and the need of three consecutive timesteps during the computation process causes the difference between the Cray's 32 Megawords and our 40 Megabytes. Usually three timesteps are enough for one animation session. In paragraph 3.1 a method for reducing the size of storage required by a factor of four without dramatically changing the flow structure is presented that can be applied before using compression techniques.

1.2 Visualization

As there is no natural visual representation for 3D vector fields and as our visual system is only trained to recognize geometric objects, color and texture, a strategy well-known from daily life is used: insert particles in the flow and see what happens ([van Wijk 93]). For computing such a path, swifting the appropriate velocity-vector fields from disk to main memory has to take place every time step as the velocity-vector fields contain far too much data (see paragraph 1.1) to fit into the graphic workstation's main memory simultaneously. For a continuos real-time animation *post-processing* is the only suitable visualization technique in this case, i.e. the velocity-vector fields are used as input to the visualization system and the data (i.e. the pathes) are computed. At a later time this data is fed into the visualization system again for display limiting the interactive control to the context of the visualization system. An advantage of post-processing is that the data can be re-examined repeatedly using different views and methods ([Marshall 90]).

A special strategy for visualizing the underlying rms values (root-mean-square values, i.e. the small fluctuations of the velocity about its mean value) of turbulent flows will be discussed in paragraph 3.

Unfortunately common perspective projections from the 3D-scene onto the 2D-screen fail to intuitively visualize those small, but important depth movements and relations between the fluid particles. Standard methods as turning the pipe around all axes do not help when viewing the flow "in action". The fluid particles are simply moving too fast.

Some high-end graphic computers offer the possibility of viewing TRUE-3D-pictures through expensive shutter glasses. These shutters are not available everywhere and are limited to the computer screen with usually just one user per computer. Therefore another cheaper and more flexible approach is needed.

Going back in the history of stereography one can find a method called *anaglyphs* ([Mucke 70]) that is only based on red-green-filter glasses – available for one US-Dollar each. This method has been successfully used for the visualization system described in this paper. Possibilities that overcome historical drawbacks such as ghost pictures and that enhance the stereographic effect by using a twofold central projection will be discussed in the following.

2 Stereo Computer Graphics and Anaglyphs

2.1 Introduction to Stereoscopy

When an observer looks at a natural scene, the views of the left and right eye differ because of the horizontal separation of the eyes. In stereoscopic displays two views of a scene are generated and displayed so that only the right eye sees the right-eye view and only the left eye sees the left-eye view. As a result, the observer sees a virtual TRUE-3D picture ([Hodges 92]). Accommodation is fixed onto the projection plane, but the eyes' axes converge as if portions of the image are at different distances. Viewing a stereoscopic image therefore disrupts the natural relationship between accommodation and convergence: when the eyes converge on an object, they also focus on the object. People not used to viewing stereoscopic pictures may experience discomfort at the beginning by this departure from the habitual relationship ([Meßner 94], [Southard 92]).

For viewing stereoscopic images on a computer-display system two basic categories can be distinguished: *time-sequential* and *time-parallel.*

Time-sequential systems present the stereoscopic image by alternating the right- and left-eye views of a scenery. The computer screen is synchronized with shutter glasses that make every eye see its appropriate view only. Obviously these systems can be used only on high-performance-graphic workstations, rely on the availability of such shutters and screens and the number of viewers is limited to the number of available shutters, which are fairly expensive.

Time-parallel systems simultaneously display the left- and right-eye view. An optical apparatus is required to separate those two views. Examples for such systems include viewer displays, polarized

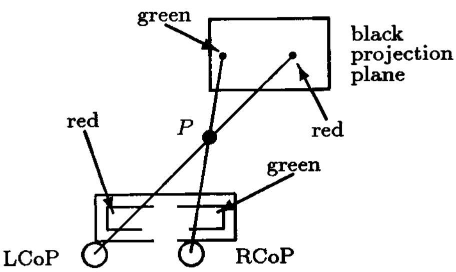

Figure 1: Anaglyphs – Object P in front of projection plane.

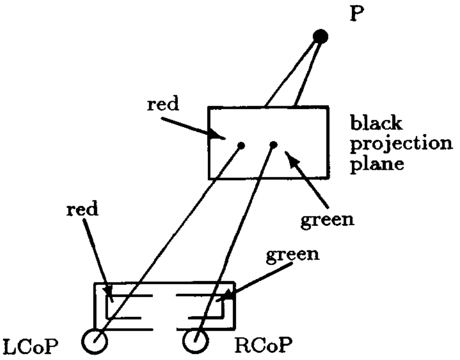

Figure 2: Anaglyphs – Object P behind projection plane.

displays, anaglyph displays and head-mounted displays ([Hodges 92]). Among these only anaglyphs fulfill our needs: cheap, several people are able to view the image at the same time, the possibility of displaying anaglyphs on various media such as TV-screens or slides. Hence a presentation in conferences or in lectures is possible.

2.2 Anaglyphs

In the 1960's anaglyphs have been widely used in books about geometry, linear algebra or vector calculus. Anaglyphs are based on the two eye views colored differently. Assuming a black background, which has proved to be superior to a white one, the right-eye view is painted red and the left-eye view green. The color of the overlapping areas is yellow – the complementary color of red and green. As an optical apparatus glasses with red and green filters are used. Usually the red filters are for the left eye. Wearing those filters the left eye (LCoP, Left Center of Projection) notices through the red filter only the red lines of the right picture as bright lines. The right eye (RCoP, Right Center of Projection) on the other hand does not see the red lines, but only the green lines. Figure 1 and 2 shows this for an object P in front of and behind the projection plane respectively ([Mucke 70]).

The major drawback in the use of anaglyphs is the limitation to only black and white pictures and some inconvenience caused by distortions of the virtual image. Because of technical reasons the green filter cannot be adjusted so that the opposite eye view is completely invisible. This sort of distortion is referred to as *leakage*, *ghosting* or *cross talk* ([McAllister 93], [Mucke 70]). This phenomenon might confuse the unexperienced viewer. In [Mucke 70] a technique is described to overcome this phenomenon by changing the brightness of the background, thus the bright ghosting should have the same color as the background. However, in our experiments this did not produce any positive effects on the screen. As there was no problem with the red color at all, we reasoned that the green color was just too bright. Therefore we reduced the RGB-value of the green color substantially to about 50% and adjusted the complementary value according to the RGB-cube (see figure 3). Reducing an RGB-color along one of the three axes of the RGB-cube can be compared to adding some more black particles to the color. This helps to reduce the brightness of the ghosting thus making it invisible. The green value should not be less than 30% of the brightest possible

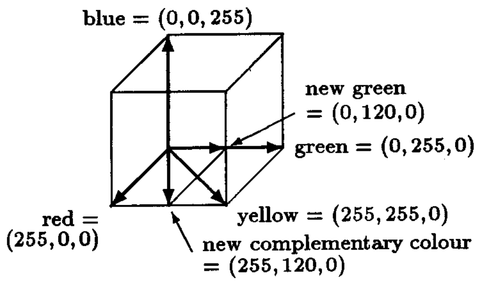

Figure 3: Cross-Talk Elimination.

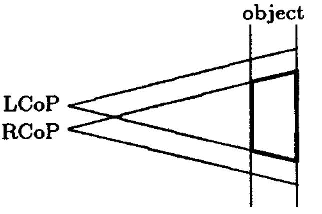

Figure 4: On-Axis Projection.

one as otherwise the red image would be too dominant. When viewing the animation on a TV screen we found it necessary to reduce the brightness of both colors once again to about 50% of the appropriate computer-screen value.

2.3 Stereographic Projections

We have been comparing the performance and features of two different projections for the use with anaglyphs and managed to point out several advantages of the *twofold central projection* compared to an *on-axis projection* implemented in graphics hardware.

2.3.1 On-Axis Projection in Graphics Hardware

The *planar projection* of a 3D object on a 2D projection plane is defined by *projectors* emanating from a center of projection, passing through each point of the object, and intersecting the projection plane thereby drawing the projection of the object. These planar projections can be divided into two basic classes: *perspective* and *parallel*. If the distance from the center of projection to the projection plane is finite, then the projection is perspective, otherwise it is parallel ([Foley 90]). As the visual effect of *perspective foreshortening* of a perspective projection is similar to that of the human visual system, perspective projections are used with stereographics.

For simulating the human viewing process, the on-axis projection makes use of two perspective projections for both LCoP and RCoP (see figure 4) thus producing two different views of the same object.

A *perspective projection* can be either specified by the angles of the rectangular viewing frustum w in y-direction, the aspect ratio a of w to the angle in x-direction and the distances of the near- and far-clipping planes n and f

```
void perspective (float w, float a, float n, float f);
```

or by its coordinates left, right, bottom and top

```
void window (float l, float r, float b, float t, float n, float f);
```

Both representations are equal with

$$\xi := \frac{w}{2 \cdot 180} \qquad (3)$$

$$l = -\tan(\xi \cdot a) \cdot n \qquad r = \tan(\xi \cdot a) \cdot n \tag{4}$$

$$t = \tan(\xi) \cdot n \qquad b = -t. \tag{5}$$

For computing a stereographic picture the distance of the projection plane d and the horizontal eye distance e have also to be taken into consideration. The constant k

$$k = -\frac{e \cdot n}{2 \cdot d} \tag{6}$$

adds to the l and r values from equation (4) thus changing the angles of the rectangular viewing frustum (taken from the Silicon Graphic stereo module). After that the view direction has to be set from the origin along the z-axis and the scene has to be translated for the left- and right-eye view at $\mp\frac{e}{2}$. This can be combined into two procedures for the left- and right-eye view, called

```
void stereoperspLCoP (float w, float a, float n, float f, float d, float e);
void stereoperspRCoP (float w, float a, float n, float f, float d, float e);
```

The method makes use of the standard perspective projections that are found implemented in hardware in some graphic machines. All shadowing, texturing, z-buffering and everything else from graphics libraries can easily be used with them. But this is not necessary for our visualization system. Anaglyphs limit us to black and white images anyway, the demand to produce a real-time animation does not allow expensive rendering.

But there is still a greater drawback: the demand to depict a scenery having portions in front of and behind the screen cannot be put into practice with the on-axis projections implemented in graphics hardware and GL or OpenGL. As shown in figure 4 a perspective projection clips an object with its near- and far-clipping planes and projects everything that falls within the viewing volume toward the apex of the pyramid, i.e. towards the center of projection ([Neider 93]). As can be seen from the above procedures, GL and OpenGL do not allow to change the position of the projection plane explicitly. Even equation (6) only changes the field of view. For non-stereographic views the result equals an explicit movement of the projection plane. But for stereographic pictures only the distance of the object from the projection plane changes, but the projection plane cannot be forced to intersect the object as the order of the left- and right-view images on the projection plane remains the same and thus the desired effect cannot be realised.

In the following we circumvent this restriction by proposing a software implementation of a twofold central projection.

2.3.2 Twofold Central Projection

In [Mucke 70] a twofold central projection onto the same horizontal projection plane for a geometric construction of anaglyphs is proposed. This is shown in figure 1 and 2 and in more abstract form in figure 5. More than 20 years later [Hodges 93] proposed about the same model because other approaches like the rotation method or on-axis projection (see [Hodges 93] and [Southard 92]) failed to represent the stereographic picture without geometric distortions.

The formula for computing the 2D-coordinates on the projection plane are easy to obtain by intersecting the line of projection with the projection plane (see [Hodges 93] and [Meßner 94]). The

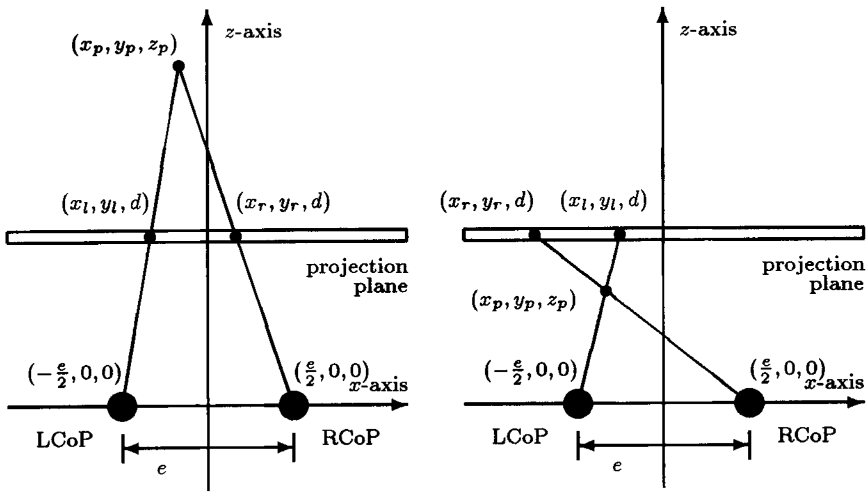

Figure 5: Twofold Central Projection.

coordinates for the left-eye view (x_l, y_l) and (x_r, y_r) for the right-eye view are

$$x_l = \frac{dx_p}{z_p} + \frac{de}{2z_p} - \frac{e}{2} \qquad y_l = \frac{dy_p}{z_p} \tag{7}$$

$$x_r = \frac{dx_0}{z_p} - \frac{de}{2z_p} + \frac{e}{2} \qquad y_r = y_l = \frac{dy_p}{z_p} \tag{8}$$

The y-values for both views are identical thus producing no obsolete vertical parallax like earlier rotation techniques (see [Hodges 93]). Both eye views can be efficiently computed with only 5 multiplications and 4 additions altogether. This is an overhead of only 2 additions per point compared to only computing one eye view. Furthermore in contrast to the on-axis projections (see paragraph 2.3.1), where both views have to be drawn strictly sequentially, similarities of both eye views can be used in the modelling phase to maximize performance. With heavy optimizations this software implementation was running on a Silicon Graphics Indigo Extreme at a speed of approximately 85% compared to the on-axis projections implemented in graphics hardware. Without a machine supporting graphical projections in hardware the twofold central projection should prove to be superior in speed.

The ability of the twofold central projection to present a virtual picture in front of and behind the projection plane (i.e. the screen) has not yet been discussed. The matrices of the on-axis projections (see 2.3.1) do not allow this feature. As can be seen in figure 1 and 2 points in front of and behind the projection plane are distinguished by the relative positions of the red and green points on the projection plane. The horizontal parallax, i.e. the difference H between the left- and right-eye view from equations (8) and (9)

$$H = x_l - x_r = e\left(1 - \frac{d}{z_p}\right) \tag{9}$$

represents the absolute z-distance of the point P from the projection plane. As this distance increases, the absolute horizontal parallax increases proportional. The horizontal parallax for points far away from the viewer and the projection plane is limited by the eye-distance. For those points with $z_p = \infty$ the projection lines become perpendicular to the x-axis. But for points between the projection plane and the viewer the horizontal parallax increases without limits. The human being is not able to converge views with too great a horizontal parallax (see [Hodges 93], [Meßner 94], [Southard 92] for example).

3 Animation of Turbulence

3.1 Data-Size Reduction

As has already been mentioned in the first paragraph, one difficulty in managing data of time varying fluid flows is the enormous size of velocity-vector fields. For an exact path calculation in those vector fields the data would have to be stored with double precision or at least with float precision. But as the trilinear interpolation between grid points in our test cast and the linear interpolation between two consecutive timesteps already propagate the deviation from the exact path, we reasoned that storing the velocities in an even less precise data format would not cause too much a breakdown of the typical flow structure. This is adequate for a visualization system since the human eyes cannot perceive those small deviations. Within a visualization system the human being cannot distinguish between small deviations. We type-converted the original float data to an unsigned-char format offering 256 discrete values. The original data is scanned for its lowest and highest values becoming the first and last of the unsigned-char values thus increasing accuracy. The double or float values are then projected onto the available palette of discrete unsigned-char values. Thus we managed to reduce the size of the velocity-vector fields to a fifth or fourth the size of the original data depending on the values stored in text or binary format respectively. The savings would be twice as much for converting from double to unsigned-char. However, this method is not intended to substitute common compression techniques but should be used additionally.

3.2 Visualizing Turbulence

In [Duvenbeck 88] a good introduction to the most fundamental methods based on Lagrange's and Euler's fluid-flow descriptions is given. In contrast to recent research dealing either with graphical tools for visualizing flows or with flows around airplanes, we have been explicitly working on the visualization of turbulence in pipes at a reynolds number of $Re \approx 7000$ at the most.

The pressure drop down the pipe for a given flow rate may be up to a hundred times larger than with laminar flows ([Bradshaw 71]). Concerning the visualization the tracer particles leave the pipe too quickly without actually having done anything "exciting". In contrast to visualizing flows around jets simply tracing the particles does not provide much information.

As we are using the data one of us (Walter Huber, see [Bungartz 94]) has computed, we know about the probability distribution of the velocity at a given point with the time average called mean velocity being the simplest statistical property (see figure 6). The fluctuations of the velocity about the mean value are typically fairly small – just about 10% of the mean velocity for points near the z-axis of a pipe ([Bradshaw 71]). Pathes of tracer particles show lines almost parallel to the z-axis.

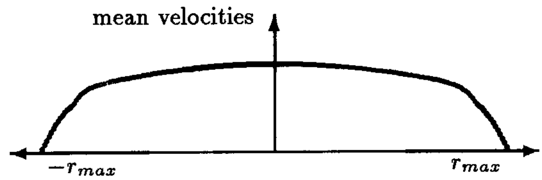

Figure 6: Mean velocities in the pipe.

Subtracting the mean z-velocities causes only those small fluctuations of velocity to remain, i.e. we are visualizing the rms values (root-mean-square values). Tracer particles are no longer following physical pathes but their movements are representing *real turbulence*.

4 Summary

We successfully managed to apply the old fashioned anaglyphic stereoscopic method to the new field of visualizing turbulence in pipe flows. We showed a possibility to overcome the obsolete effect of ghosting.

A twofold central projection as a geometric construction for anaglyphs has been transfered for computer use and shown to be superior to common on-axis projections thus allowing the representation of a virtual scenery in front of and behind the projection plane.

Anaglyphs as a time-parallel stereoscopic system do not depend on the screen and special graphics hardware, but can as well be represented on different other media such as television, photos or slides making demonstrations in conferences and lectures easier.

We attempted to reduce the size of the velocity-vector fields by type-converting to a less precise format.

For visualizing turbulence we proposed subtracting the appropriate mean velocities so that the movement of a tracer particle depicts real turbulence.

References

[Bradshaw 71] Peter Bradshaw. *An Introduction to Turbulence and its Measurement*. Pergamon Press, Oxford, 1971. Reprint 1994.

[Bryson 92] Steve Bryson, Creon Levit. *The Virtual Wind Tunnel*. IEEE Computer Graphics & Applications, pp. 25–34, 1992.

[Bryson 93] Steve Bryson. *Virtual Reality in Scientific Visualization*. Comput. & Graphics, Vol. 17, No. 6, pp. 679–685, 1993.

[Bungartz 94] Hans Bungartz, Walter Huber. *First Experiments with Turbulence Simulation on Workstation Networks using Sparse Grid Methods*. in: E. H. Hirschel (ed.). *Flow Simulation with High-Performance Computers II*. Vieweg Verlag. Braunschweig. *will be published*.

[Duvenbeck 88] Harald Duvenbeck, Alfred Schmidt. *Darstellung zwei- und dreidimensionaler Strömungen*. Institut für Angewandte Mathematik, SFB 256, Universität Bonn. in: H. Jürgens, D. Saupe (eds.). *Visualisierung in Mathematik und Naturwissenschaften*. Bremer Computergraphik-Tage 1988. Springer-Verlag, 1988.

[Foley 90] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes. *Computer Graphics – Principles and Practice*. Addison-Wesley, Inc., 2nd ed., 1990.

[McAllister 93] David F. McAllister. *Introduction*. in: David F. McAllister (ed.). *Stereo Computer Graphics and Other True 3d Technologies*. pp. 119–151. Princeton University Press, Princeton, New Jersey, 1993.

[Hodges 92] Larry F. Hodges. *Time-Multiplexed Stereoscopic Computer Graphics*. IEEE Computer Graphics & Applications, pp 20–30, March 1992.

[Hodges 93] Larry F. Hodges, David F. McAllister. *Computing Stereoscopic Views*. in: David F. McAllister (ed.). *Stereo Computer Graphics and Other True 3d Technologies*. pp 71–89. Princeton University Press, Princeton, New Jersey, 1993.

[Krause 94] E. Krause. *High Performance Computing in Fluid Mechanics*. in: Michael Griebel, Christoph Zenger (eds.). *Numerical Simulation in Science and Engineering*. Proceedings of the FORTWIHR Symposium on High Performance Scientific Computing. München, 1993. Vieweg-Verlag. Braunschweig, 1994.

[Marshall 90] Robert Marshall, Jill Kempf, Scott Dyer, Chieh-Cheng Yen. *Visualization Methods and Simulation Steering for a 3D Turbulence Model of Lake Erie*. The Ohio Supercomputer Graphics Project, The Ohio Supercomputer Center, The Ohio State University. ACM, 1990

[Meßner 94] Wolfgang Meßner. *Stereographische Visualisierung von dreidimensionalen, turbulenten und instationären Rohrströmungen mit Hilfe von Anaglyphen*. Diplomarbeit. Institut für Informatik, Lehrstuhl für Ingenieuranwendungen in der Informatik, Technische Universität München, 1994.

[Mucke 70] Helmut Mucke. *Anaglyphen – Raumzeichnungen – Eine Anleitung zum Konstruieren von Raumbildern*. BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1970.

[Neider 93] Jackie Neider, Tom Davis, Mason Woo. *OpenGL Programming Guide*. Addison-Wesley, 1993.

[Southard 92] David A. Southard. *Transformations for Stereoscopic Visual Simulation*. Comput. & Graphics Vol. 16, No. 4, pp 401-410, 1992.

[Unger 94] Friedemann Unger. *Numerische Simulation turbulenter Rohrströmungen*. Dissertation. Lehrstuhl für Fluidmechanik, Technische Universität München, 1994.

[van Wijk 93] Jarke J. van Wijk. *Flow Visualization with Surface Particles*. IEEE Computer Graphics & Applications, July 1993.

[Woodward 92] Paul R. Woodward. *Scientific Visualization of Complex Fluid Flow*. Pacifico Yokohama (ed.). Supercomputing Japan 92, Conference Proceedings. Tokyo, 1992.