

Creative Flow Visualization Using A Physically Accurate Lighting Model

Mark J. Stock

Department of Aerospace Engineering

The University of Michigan

1320 Beal Ave. FXB Building

Ann Arbor, MI, 48109-1278 USA

1-734-936-0539

mstock@umich.edu

ABSTRACT

A method is presented for creative visualization of three-dimensional turbulent flows in a two-dimensional image using a more compact representation of the flowfield and a photometrically accurate rendering step. Three-dimensional fluid flow is compactly represented as vortex lines or sheets, and a computational method called a vortex method, which is uniquely suited for this representation, is used to calculate the evolution of those elements in time and space. The positions of the vortex elements at any moment in time constitute a detailed three-dimensional form of the flow structure. To take advantage of the strengths of the human visual system and effectively convey meaning and understanding of this structure with an image, we used a renderer that handles physical light properties such as shadow and interreflection. The Radiance lighting simulation system, with its accurate and tunable interreflection lighting algorithm, was chosen to render the images. We believe that with this method, greater understanding of complex turbulent flows can be communicated, and more informative and captivating images can be created.

Keywords

rendering, visualization, fluid dynamics, vortex dynamics, turbulence.

1. INTRODUCTION

From before birth, we familiarize ourselves with the indirect effects of fluid motion. We wade in an invisible sea of air and sense its pull on our clothes, our hair, and our car. We gain glimpses of its hidden motions when we pour milk into water, or watch a cloud form or smoke rise. We rarely truly see its internal

motions, though, because the fluid we observe is either transparent, and thus its motions are invisible, or it is opaque and we are unable to see inside of it.

Purposefully injecting smoke or other particles into a flow in order to more closely observe it is probably the oldest, and is still a valid, method for physical visualization of flows. Unless a sequence of images is made, though, no velocity information can be gained, and particulate density information is all that remains.

Computer simulation, though, offers us a window through which to see and gain understanding of flows of all kinds.

Because computers calculate and store all of the information pertaining to a flow, we are able to analyze any aspect of it, choosing from any number of visualization methods.

The first real advances in two-dimensional flow visualization, beyond grid arrows and individual streamlines, stemmed from the creation of spot noise textures [21] and its successor, the line integral convolution (LIC) method [4]. LIC has since been improved [17, 22] and applied to pseudo-3D flows [16], as 2D textures on 3D objects [15], and as a three-dimensional texture [4, 7]. The effectiveness of LIC is due, in part, to the obvious similarity with 2D illuminated particle visualization.

To display more data than simply velocity, LIC can be combined with multiple layers of layers of glyphs [8] or data-driven spots [18], each representing other flow variables such as strain, vorticity, or turbulence intensity. These methods often create confusing images because they are cluttered with too many icons and colors that individually may have little intuitive meaning. However, to a trained observer, they are able to communicate large amounts of information, including complex relations between variables.

The primary problems with flow visualization in three dimensions are the occlusion of data behind other data and the lack of directional and depth hints. One solution is to allow interactive manipulation of the three-dimensional dataset, allowing the user to view occluded regions or details by rotating them into view or bringing them closer to the virtual viewpoint. Another solution is to define a reduced set of glyphs or markers that can similarly represent the complex fluid flow pattern. This

First published at COSIGN-2002,
02 – 04 September 2002, University of Augsburg,
Lehrstuhl für Multimedia-Konzepte und Anwendungen,
Germany

avoids the problem, common in large flow datasets, of visually overlapping geometry. Methods to extract volumes of high vorticity [25], individual vortex lines [24], or tubes [1] from velocity field datasets have met with some success. Other methods aim to reduce the large amount of data by extracting salient features from streamlines in the velocity field [19], or by finding topological critical points and regions in 3D velocity data [2, 6].

In this paper, we assemble a visualization method that allows us to look at a three-dimensional turbulent flow by representing it as parcels of rotating fluid, each rendered as a solid object.

Even this representation can become visually confusing, so to enhance understanding, we turned to proven visual cues such as lighting, shadow, and interreflection. Recent research in vision and perception indicates that accounting for these illumination components during rendering leads to more effective communication of 3D shape. New methods and techniques in computer graphics can recreate these effects, and are thus able to create more realistic images of 3D objects than have been possible before.

Regardless of the technique used for flow visualization, the source material—the flow itself—is characterized by forms and motions that are universal to fluid flow, and are at times both consciously and unconsciously familiar. A by-product of this universality is that these images speak a common language to all observers. We seek to use this language to create insight and appreciation for the invisible and complex motions of turbulence, and to possibly communicate more than the sum of its parts.

2. FLUID DYNAMICS

2.1 Governing Equations

The modern era of fluid dynamics arguably began in the first half of the nineteenth century when Claude Navier and George Stokes first wrote the equations that now bear their names. The Navier-Stokes equations, the fluid equivalent of Newton's $F = ma$, define the three components of fluid vector acceleration $\delta\mathbf{u}/\delta t$ in terms of the vector-valued velocity \mathbf{u} and gravity \mathbf{g} and the scalar pressure p , density ρ , and coefficient of kinematic viscosity ν . Together with the constraint equation for conservation of mass (1), the vector form of these equations constitutes a good starting point for the study of incompressible fluid dynamics.

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{g} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad (2)$$

The last term in (2) is an approximation of the friction forces present in the flow, and scales in magnitude with the velocity and the coefficient of viscosity ν . When this term dominates the equation, as is the case with flow at very small scales or in fluids with high viscosity such as molasses, the flow is said to be laminar. Laminar flows are typically smooth and non-fluctuating. In contrast, when the friction term is small enough that inertia no longer succumbs to the stabilizing effects of viscosity, the flow's

motions change drastically and the flow becomes turbulent. A turbulent flow is characterized by large and rapid velocity fluctuations of a seemingly random nature. Both laminar and turbulent flow can be observed in Fig. 1 [20].

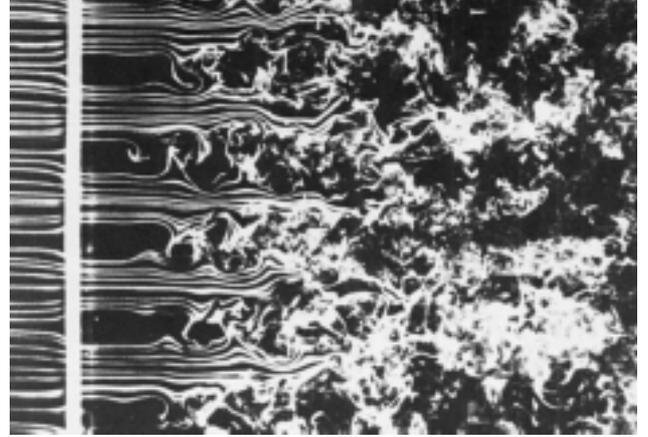


Figure 1: Smoke visualization of uniform laminar stream owing through a perforated plate, showing laminar, transitional, and turbulent flow, photograph by Thomas Corke and Hassan Nagib.

To date, exact solutions of the Navier-Stokes equations exist for about 80 particular cases, most of them for laminar flow in special geometries.

2.2 Computational Fluid Dynamics

Modern fluid dynamics research deals very frequently with turbulent flows and other flows with complex unsteady motions for which analytic solutions do not exist. Computational fluid dynamics (CFD) provides a way to calculate an approximate solution to the Navier-Stokes, or any similar fluid dynamic equation set, for a wide variety of flow regimes or boundary condition. CFD works by breaking one large complicated problem into a very large number of much smaller and simpler interrelated problems and concurrently solving them on a computer.

There are two major types of CFD formulations: Eulerian, in which flow properties are computed and stored on a fixed grid, and Lagrangian, in which flow properties are tied to particles that move about in a gridless domain. An extensive body of literature is available for each of these formulations, and each has advantages and disadvantages.

Eulerian methods have dominated industrial CFD and thus created a need for visualization tools that can efficiently deal with large volumes of grid data. Current visualization methods for large field datasets either must employ high data throughput rates or rely on data reduction to emphasize meaningful features.

Lagrangian methods, being a more natural way to represent a physical system, usually incorporate computational elements that, by definition, represent important features of the system such as discrete packets of fluid, fluid interfaces, or solid particles. Computation time is thus not wasted in areas where nothing interesting is occurring. Often, though, a greater density of

Lagrangian particles is required to maintain simulation accuracy than would be required for an equivalent Eulerian method.

A Lagrangian vortex method was used to create the datasets from which images appearing in this paper are rendered.

More details of this method appear in section 2.4.

2.3 Vorticity

Flow properties typically include velocity, density, pressure, temperature, or scalar fraction, though many other properties can be computed from this basic set. Depending on the problem at hand, an alternate property, called vorticity, can be far more informative than those listed above.

Vorticity is the curl of the velocity ($\omega = \nabla \times \mathbf{u}$) and is a vector quantity which points along the axis of fluid rotation and has a magnitude proportional to the rate of rotation of the fluid around that axis. Vorticity is most often created when fluid is in moving contact with a solid boundary, though it can be created in other special situations. Additionally, it can be shown that knowing the vorticity field and boundary conditions of a flow is equivalent to knowing the velocity field.

In this paper, because we deal primarily with incompressible, constant-temperature, single-phase turbulent flows, we can make several simplifications to the equations of motion, not the least of which is that we can rewrite the Navier-Stokes relations in terms of the vorticity. Taking the curl of the Navier-Stokes equation gives the vorticity transport equation.

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \omega \quad (3)$$

One of the benefits of this formulation is the absence of the pressure term. The equation is now only dependent on vorticity and velocity. This greatly simplifies some numerical methods designed to solve the Navier-Stokes equations.

In flows with minimal viscous diffusion, another advantage of the vorticity formulation manifests. In these flows, the volume of fluid with significant vorticity magnitude is typically a small fraction of the total flow volume. This means that the flow can be represented in a more compact form by vorticity than is possible with velocity. This fact lends support to computational methods in vorticity variables. This compactness is surprisingly illustrated in the results of Banks and Singer [1]. They achieved compression ratios of up to 4000:1 by representing a turbulent flow field with vortex tubes, though no tests were performed to compare the reconstructed velocity field to the original velocity field.

A disadvantage of using vorticity to represent flow fields is that, without animation of the flow or appropriate training, vortex lines themselves give no indication of flow velocity.

This could be because visual representations of fluid vorticity itself are far less common in nature than those of velocity. Fortunately, because vortex lines flow with the velocity field, we are still able to perceive fluid-like shapes and motions in the vorticity field.

2.4 Vortex Methods

A vortex method is a computational method in which the flow is represented by a collection of Lagrangian particles of vorticity moving under the self-influence of one another. This motion is

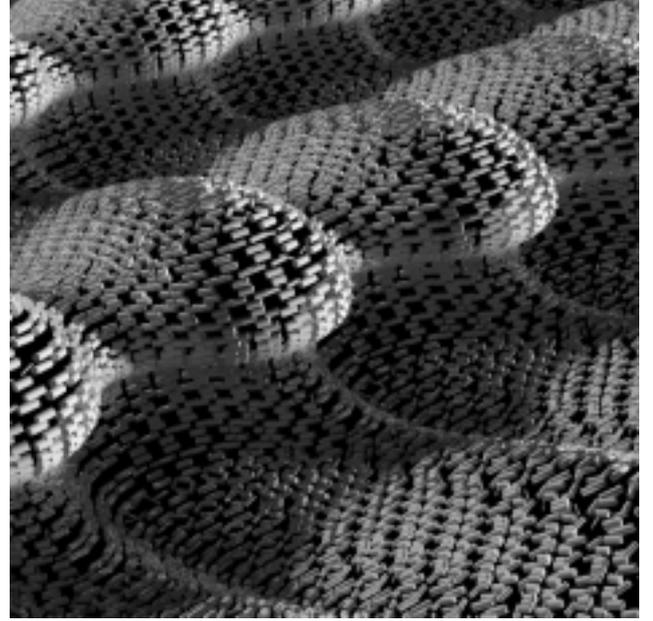


Figure 2: Development of three-dimensional shear layer instability, the element vorticity is represented by cylinder volume and direction; image rendered with global illumination calculation

quantified by the Biot-Savart law (4), which determines the velocity at a point in space given a complete definition of the vorticity field.

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{4\pi} \int \frac{\omega(\mathbf{x}', t) \times (\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}' \quad (4)$$

Using this formula, a vortex particle's velocity can be computed from the vorticity and location of every other particle. Each particle is then advected according to its local velocity, and has its vorticity modified to account for vortex stretching and viscous effects. Fig. 2 shows a deforming shear layer defined by vortex particles of various strengths and directions.

It is obvious that if the Biot-Savart law were used to calculate the velocity of each of the N particles in a simulation, then the calculation of a single time step would involve $O(N^2)$ evaluations. This is clearly inappropriate for large values of N , as would be necessary for simulations of three-dimensional turbulence. The simulation results presented here were performed with a Vortex-In-Cell (VIC) method, which uses a temporary grid to solve for the velocity field instead of evaluating the Biot-Savart kernel for every vortex element pair.

The discretization and remeshing routines are unique to this method, and govern the final shape of the rendered geometries. The simulation starts with a single vortex ring or other similar closed vortex loop, defined as a series of nodes connected

together with elements. The vorticity is represented as circulations on the edges of triangular elements. The resulting vortex line, through the VIC method, defines a specific velocity field. That velocity field is then used to advect the nodes that make up the line, changing the geometry of the line. When the distance



Figure 3: Rendering of a collection of vortex segments from a wholly contrived flow inside of a cube, with global illumination calculation. Note the illusion of depth despite the lack of perspective.

between two nodes exceeds a preset threshold, a new node is inserted between the two. Likewise, when two nodes approach to within a set distance, they are merged into one node. The objects rendered in this work are the result of this constant stretching and refining of the initial vortex line.

More complete descriptions of this and other types of vortex methods are available in the literature [11, 5].

3. VISUALIZATION

Richard W. Hamming, mathematician and computer scientist, best emphasized the need for good data visualization when he penned the following phrase:

The purpose of computation is insight, not numbers.

Rendering, in some manner, is a requirement for visualization of three-dimensional data. The rendering method can be as simple as an isometric line drawing, as complex as a costly separate calculation to realistically simulate the behavior of light in a scene, or any combination of methods in between. Fig. 3 illustrates a mix of methods: an unrealistic parallel view projection is used with a realistic interreflection calculation in the rendering step. It is important that the final choice of

rendering method support, and not detract from, the message to be visually conveyed.

Additionally, effective visualization relies on choosing the data that most succinctly communicate the desired message. For the same reasons that a vortex method was chosen to simulate the flow, so shall vorticity be chosen as the primary data variable.

3.1 Direct Lighting and Shadow

There are essentially three components of illumination on any object; the first is classical—or direct—shading, which depends on the angle and distance between the surface and the illumination. The second is that of shadowing, which depends on the visibility of the light source from the surface patch. Lastly, surface interreflections account for the specular and diffuse reflections of incoming light on a surface. The first two of these components are related, as collimated light from direct sources not only accounts for classical shading, its occlusion is what creates a shadow. Interreflection will be covered separately.

The most common method for simulating direct lighting and shadows is called raytracing. Any raytracing program will treat direct lighting and shadows properly, and thereby be able to portray a three-dimensional scene in a two-dimensional image. In order to effectively communicate the three-dimensional shape using visual cues in the image, though, the scene must conform to the human visual system's assumptions of lighting conditions, viewpoint, and object.

Experiments conducted to isolate human perceptual cues used in determining an object's shape from its shading have shown that not only does the human visual system use information from shadows to resolve ambiguities [3], but it makes prior assumptions, too. These assumptions are that the illumination is from above [13], the viewpoint is from above [14], and the shape itself is globally convex [10]. Most visualization programs that simulate direct lighting give control over these parameters.

Some visualization tools are designed to support only classical shading, thus ignoring many of the cues that our visual system relies on. The popularity of those tools, though, lies in their rendering speed. For example, the image in Fig. 4 was rendered using a general OpenGL triangle viewer at near-interactive rates, though it does not render shadows or interreflection.

3.2 Interreflection

Interreflection is the transport of light through a scene via successive specular and diffuse reflections off of surface patches.

It can manifest as a very subtle change in brightness or color in an image or it can change the character of an image substantially. It is also difficult to simulate computationally, often increasing rendering time by an order of magnitude or more.

Pure diffuse lighting, which is a simple way to mimic the effects of even interreflection, is shown to be just as effective an aid for distinguishing the local qualitative shape of surfaces as direct lighting and shadows combined [9]. A study by Madison [12] showed that interreflection and shadow could be equally important for human visual perception of spatial layout. A combination of both, in their experiment, was more effective than either individual method for communicating virtual object contact.

This importance of interreflection is easily demonstrated with the two images in Fig. 5. An identical scene was rendered twice, once without and once with interreflection. It is obvious that the

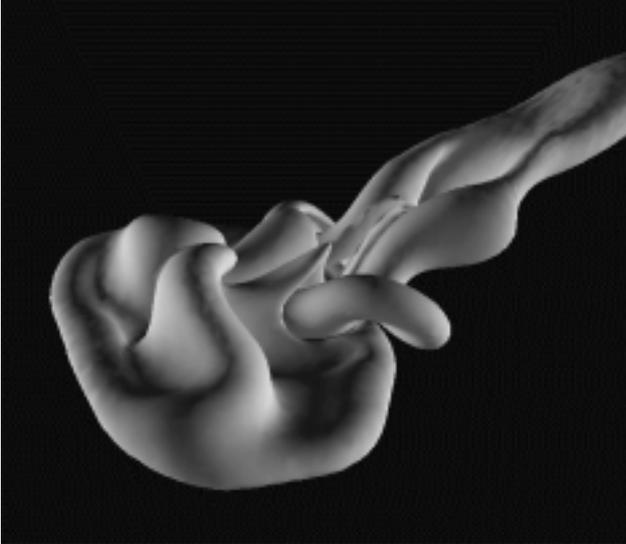


Figure 4: Advancing front of a buoyant jet, triangular elements rendered in OpenGL with direct lighting only

interreflection calculation adds significant visual depth to the image. It can even negate the loss of perspective, another valuable depth cue, illustrated in Fig. 3.

Interreflection can clearly enhance visual communication, though it has yet to gain wide acceptance. Most immersive media and data visualization techniques use only perspective and interactivity to communicate meaning.

3.3 Depth Cueing and Fog

In addition to the three primary components of scene illumination, there exist other methods to communicate depth or distance in a flat image. These are depth cueing and participating media.

Aerial perspective, sometimes called depth cueing, is the use of subtle changes in the color or intensity of light as it travels long distances through an atmosphere. Faraway objects take on more of the color of the horizon the closer they are to it. This is a commonly used trick borrowed from painting that can easily be incorporated into a computational method.

When fog is thick enough to scatter light away from its original direction, though, it begins to participate in the rendering calculation and is then referred to as participating media. This adds another depth-creating effect, as participating media is capable of giving visual volume to empty space itself. As light passes through a volume of fog, a certain amount scatters away from its original direction. Some of this light scatters toward the viewer, creating a sunbeam-like effect. These sunbeams not only signify a lack of solid structure within their volume, they hint at the shape of the opening that lets the light through. In Fig. 6, this effect is prominent enough to steal attention from the solid material in the scene. Fog is yet another visual cueing effect that manifests during realistic rendering.



Figure 5a: Late-stage development of a perturbed vortex ring, rendered without global illumination



Figure 5b: Late-stage development of a perturbed vortex ring, rendered with global illumination

3.4 Radiance

Realistic rendering, it seems, is the chief vehicle for communication of three-dimensional shape via a static two-dimensional image. With this as the criterion, Ward Larson's Radiance [23] was chosen for final visualization of most of the images appearing in this paper.



Figure 6: Repeating volume of periodic turbulence, rendered with fog and interreflection

Ward-Larson describes Radiance as a lighting simulation and rendering system. Radiance includes a photometrically accurate interreflection calculation algorithm and contains a large number and variety of modeling and analysis tools. Its renderer is capable of calculating all of the aforementioned components of illumination, all in scenes containing nearly any level of complexity. More accurate rendering calculations or larger scenes necessarily take longer to render, though Radiance tackles scenes impossible with other renderers. An example of the type of scene that Radiance excels at is shown in Fig. 7. The scene contains over one quarter million primitives and the ambient interreflection file contains 120 MB of information.



Figure 7: Vortex lines in fully developed turbulence, rendered with global illumination

Radiance versions 3.1 and 3.4 for UNIX were used for this work. Special compile options were used to allow very large ambient interreflection data files and to activate true Monte Carlo sampling. Images were rendered at two to four times presentation

resolution and downsampled with a radial Gaussian filter. Additionally, some images were post-processed with a multi-step filter that mimics human visual response.

4. CONCLUSION

Though turbulent fluid flow is an extraordinarily common occurrence in daily life, its effective visualization remains elusive. One possible solution is to represent a complicated turbulent flow as a collection of connected vortexes, and to simulate its development with a computational method utilizing those vortex elements. In doing this, previously invisible structures are uncovered. The vortex lines, particles, and surfaces making up those structures are the most basic continuum fluid dynamic elements, and their organization is what defines the turbulent flow. When those elements are represented as solid objects, they form a mathematical sculpture of the turbulence, but one that would never be able to support itself as a physical structure.

Evidence supports the proposal that the best way to visualize a virtual object, for the purposes of communicating meaning, is to use a rendering method that most accurately simulates the true behavior of light in the scene. This simulation, by definition, will produce an image containing all of the same visual cues that real scenes possess, including light, shadow, interreflection, and depth cueing. Visualization software is now capable of computing all of these effects accurately, even for large scenes. This capability opens the doors for better data visualization, and for more creative and expressive visualization of things invisible or wholly impossible.

5. ACKNOWLEDGMENTS

Thanks to ChevronTexaco for supporting the vortex method research that helped produce these datasets, and to Sergey Klibanov of the University of Michigan for his work on an OpenGL viewer. Also thanks to Georg Mischler for his information on Radiance compile options, and to Greg Ward Larson for his suggestions and encouragement to continue learning Radiance. Lastly, thanks to Werner J. A. Dahm of the University of Michigan, and Grétar Tryggvason of Worcester Polytechnic Institute, for their advice and guidance.

6. REFERENCES

- [1] D. C. Banks and B. A. Singer. Vortex tubes in turbulent flows: identification, representation, reconstruction. In Proceedings, IEEE Conference on Visualization '94. IEEE Computer Society Press, 1994.
- [2] R. Batra and L. Hesselink. Feature comparisons of 3-D vector fields using earth mover's distance. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 105{114. IEEE Computer Society Press, 1999.
- [3] K. Berbaum, T. Bever, and C. S. Chung. Extending the perception of shape from known to unknown shading. Perception, 13:479 - 488, 1984.

- [4] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 263{270. ACM Press, 1993.
- [5] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge Univ. Press, Cambridge, UK, 1999.
- [6] W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 349 - 354. IEEE Computer Society Press, 1999.
- [7] V. Interrante and C. Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In Proceedings of the conference on Visualization '97, pages 421- 424. ACM Press, 1997.
- [8] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 333{340. IEEE Computer Society Press, 1999.
- [9] M. S. Langer and H. H. Bülthof. Depth discrimination from shading under diffuse lighting. *Perception*, 29:649 - 660, 2000.
- [10] M. S. Langer and H. H. Bülthof. A prior for global convexity in local shape-from-shading. *Perception*, 30:403 - 410, 2001.
- [11] A. Leonard. Vortex methods for flow simulation. *J. Comput. Phys.*, 37:289 - 335, 1980.
- [12] C. Madison, W. Thompson, D. Kersten, P. Shirley, and B. Smits. Use of interreflection and shadow for surface contact. *Perception and Psychophysics*, 63(2): 187 - 194, 2001.
- [13] V. Ramachandran. Perception of shape from shading. *Nature*, 331:163 - 165, 1988.
- [14] F. R. Reichel and J. T. Todd. Perceived depth inversion of smoothly curved surfaces due to image orientation. *Journal of Experimental Psychology: Human Perception and Performance*, 16:653 - 664, 1990.
- [15] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl. Interactive exploration of volume line integral convolution based on 3D-texture mapping. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 233 - 240. IEEE Computer Society Press, 1999.
- [16] G. Scheuermann, H. Burbach, and H. Hagen. Visualizing planar vector fields with normal component using line integral convolution. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 255 - 261. IEEE Computer Society Press, 1999.
- [17] D. Stalling and H.-C. Hege. Fast and resolution independent line integral convolution. In Proceedings of the conference on Visualization '95, pages 249{256. IEEE Computer Society Press, 1995.
- [18] R. Taylor. Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications*, 22(3): 6 - 10, 2002.
- [19] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 35 - 42. IEEE Computer Society Press, 1999.
- [20] M. Van Dyke. *An Album of Fluid Motion*. Parabolic Press, Stanford, CA, 1982.
- [21] J. J. van Wijk. Spot noise texture synthesis for data visualization. *Computer Graphics*, 25(4): 309 - 318, 1991.
- [22] V. Verma, D. Kao, and A. Pang. PLIC: bridging the gap between streamlines and LIC. In Proceedings of the conference on Visualization '99: Celebrating ten years, pages 341{348. IEEE Computer Society Press, 1999.
- [23] G. Ward Larson and R. Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. Morgan Kaufmann, San Francisco, CA, 1998.
- [24] L. A. Yates and G. T. Chapman. Streamlines, vorticity lines, and vortices. In 29th Aerospace Sciences Meeting. AIAA, 1991.
- [25] N. J. Zabusky, O. N. Boratav, R. B. Pelz, M. Gao, D. Silver, and S. P. Cooper. Emergence of coherent patterns of vortex stretching during reconnection: A scattering paradigm. *Phys. Fluids A*, 2(5): 2469, 1991.