

Case Study: Visualizing Ocean Flow Vertical Motions using Lagrangian–Eulerian Time Surfaces

Josh Grant¹ Gordon Erlebacher² James O’Brien¹

¹Center for Ocean–Atmospheric Prediction Studies,
Florida State University, USA*

²School of Computational Science and Information
Technology, Florida State University, USA[†]

Abstract

Ocean model simulations commonly assume the ocean is hydrostatic, resulting in near zero vertical motion. The vertical motion found is typically associated with the variations of the thermocline depth over time, which is mainly effected by the development and movement of ocean fronts and eddies. A new technique, extended from Lagrangian–Eulerian Advection, is presented to help understand the variation of vertical motion associated with the change in thermocline depth over time. A time surface is correctly deformed in a single direction according to the flow. The evolution of the time surface is computed via a mixture of Eulerian and Lagrangian techniques. The dominant horizontal motion is textured onto the surface using texture advection, while both the horizontal and vertical motions are used to displace the surface. The resulting surface is shaded for enhanced contrast. Timings indicate that the overhead over standard 2D texture advection is no more than 12%.

Keywords: Unsteady vector fields, ocean currents, normal velocity.

1 INTRODUCTION

Although oceanography is a well-developed science and research has been done to visualize its currents [2, 6, 9, 18], little research has been done to visualize the vertical motion found in the dominant features, including ocean fronts and eddies. The vertical motion within ocean fronts and eddies is insignificant when compared to the horizontal motion, but it is still important to study because maritime life depends on it. The majority of marine life inhabits the top 100 to 200 meters below the ocean surface, because sunlight does not penetrate below these depths. The plant life in these regions depends on vertical motion to transport nutrients from the deeper waters to the surface. Without the appropriate nutrients, photosynthesis cannot occur and thus plant life cannot survive [3]. Regions of upwelling (upward vertical motion) often contain an abundance of maritime life and are typically targeted by fisheries.

The currents in the Gulf of Mexico are characterized by what oceanographers call the loop current and by large anticyclonic eddies (Figure 1 left). The loop current is a warm ocean frontal zone bringing warmer water from the Caribbean Sea into the Gulf of Mexico (Figure 1 left). These dominant flow features in the Gulf of Mexico can span hundreds of kilometers in the horizontal directions. However, the ocean depth is measured in thousands of meters, a two to three order of magnitude ratio. In order to study the vertical variations found in the ocean, oceanographers often scale the vertical by some arbitrary amount [8].

*{grant.obrien}@coaps.fsu.edu

[†]erlebach@csit.fsu.edu

Numerical simulations of ocean flow typically assume the ocean is hydrostatic¹, which results in near zero vertical motion. Any vertical motion found is mainly due to the changes in the thermocline² (Figure 1 right) depth over time. The movement of ocean fronts, eddies, and internal waves effect the thermocline depth and oceanographers are interested in the long-term effects of these depth changes. When the thermocline moves up or down, upwelling (upward motion) or downwelling (downward motion) occurs respectively.

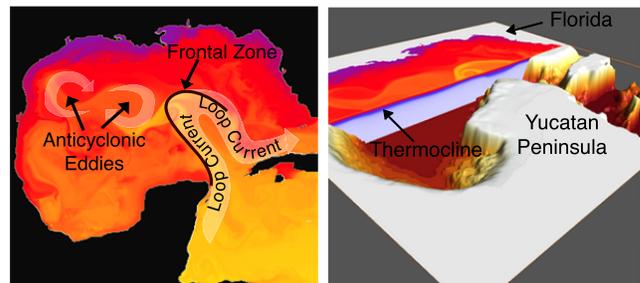


Figure 1: Left: Surface temperature in the Gulf of Mexico with the frontal zone marked and the loop current and anticyclonic eddies superimposed. Right: View of the thermocline in the Gulf of Mexico with the ocean topography drawn as well. The vertical coordinates are scaled by a factor of 20. Warmer water and cooler water is colored by red and blue respectively.

In this case study, we extend the Lagrangian–Eulerian Algorithm (LEA) [5] to take into account the vertical motion of the flow. Particles are uniformly distributed on a horizontal plane. Their color is encoded into a texture that will be advected. When fetching the particle property at the previous time step, its vertical motion is taken into account. This results in a time-dependent surface onto which the advection texture is projected.

2 RELATED WORK

In the last several years, visualization techniques with texture advection [4, 7, 5, 13] have increased. Two advantages of the technique are 1) every texel becomes a seed point and 2) the number of seed points is independent of the underlying vector field and is held fixed. By uniformly covering the domain, the particles identify all the features of the flow field. In addition to basic advection, optional post-processing stages are possible, including LIC [1], velocity masks, and dye injection. Although it can be extended to 3D, texture advection may be best suited for 2D vector fields to prevent information overload. This problem is common to many vector field visualization techniques. Furthermore, the animation frame rate drops dramatically when displaying 3D textures.

Time surfaces (also known as flow surfaces) are a useful tool to investigate the dynamics of a 3D unsteady velocity field [10, 14, 15,

¹Density of the water increases with depth.

²A layer in a large body of water with an abrupt temperature gradient.

17]. They are a 3D version of timelines and require an initial collection of particles on a surface. The trajectories of the particles are then updated and subsequent surfaces are formed. When time surfaces are created using standard particle tracing, the surfaces can self-intersect and become difficult to interpret. Another approach implemented by Westermann *et al.* [14] creates time surfaces implicitly to prevent self-intersection.

3 EXAMPLE

Consider a 3D flow, uniform in x , a zero y component, and a sinusoidal variation in z :

$$\mathbf{v}(x, y, z) = (u_0, 0, \varepsilon \sin(kx - \omega t)) \quad (1)$$

Starting from $(x_0, y_0, 0)$, a particle follows the trajectory $x(t) = u_0 t + x_0$, $y(t) = 0$, and

$$z(t) = \frac{-2\varepsilon}{\omega - ku_0} \sin\left(\frac{k(u_0 t + 2x_0) - \omega t}{2}\right) \sin\left(\frac{(\omega - ku_0)t}{2}\right) \quad (2)$$

If particles are uniformly distributed along the infinite surface $z = 0$ at $t = 0$, the time variation of this surface is given by equation (2). This surface is the product of a wave propagating toward positive x and an oscillatory wave that returns to zero with period $T = (2\pi)/(\omega - ku_0)$. Figure 2 displays a finite domain $(x, y) \in [0, 2\pi]$ of the velocity field defined in equation (1) by showing a time surface and a height field of the vertical velocity. The horizontal motion is textured onto the surfaces, a technique similar to [11]. The velocity-based height field shows the wave speed associated with the normal velocity profile, namely ω/k , while the time surface, consistent with particle tracking, shows the correct wave speed of $(\omega - 2ku_0)/(ku_0)$. The time surface also captures the periodic flattening of the manifold. However, most vector fields are not defined over an infinite domain.

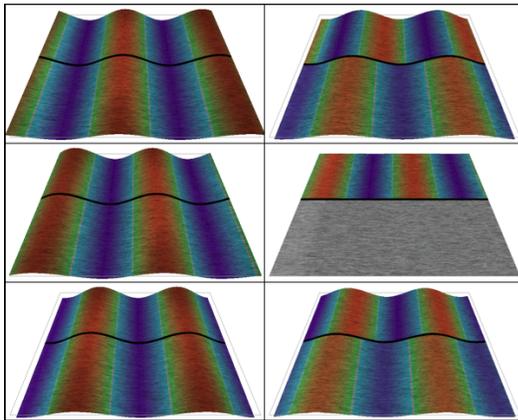


Figure 2: Three frames from an animation of the solution to Eq. (1). The height field is $w(x, y, t)$ (left) and $z(x, y, t)$ (right). Time increases from top to bottom. The half of images are color coded with w , the bottom half with z .

4 LAGRANGIAN-EULERIAN ADVECTION

It is necessary to understand the concept behind Lagrangian-Eulerian Advection (LEA) [5] before explaining the new approach to time surfaces. The two are similar and the only major difference is the property advected. In this section, we provide enough detail about LEA to follow the proposed extensions.

Let $\mathbf{v}^t(\mathbf{x})$ denote an unsteady vector field at time t and position \mathbf{x} . Let p denote an arbitrary particle and let $\mathbf{x}^\tau(p)$ refer to its trajectory,

defined by

$$\frac{d\mathbf{x}^\tau(p)}{d\tau} = \mathbf{v}^{t+\tau}(\mathbf{x}^\tau(p)) \quad (3)$$

with initial condition $\mathbf{x}^0(p) = (i, j)$, where (i, j) refers to a texel in the advected noise texture. The Lagrangian phase of the algorithm retrieves the particle property at the previous discrete time $t - h$ by integrating equation (3) backward in time. At time $t - h$, the particle is located at

$$\mathbf{x}^{-h}(p) = \mathbf{x}^0(p) + \int_0^{-h} \mathbf{v}^{t+\tau}(\mathbf{x}^\tau(p)) d\tau \quad (4)$$

Interpolating the particle property at \mathbf{x}^{-h} , the advected noise texture, \mathbf{N} , advects according to $\mathbf{N}^t(\mathbf{x}^0) = \mathbf{N}^{t-h}(\mathbf{x}^{-h})$, which constitutes the Eulerian phase of the algorithm. The superscript on \mathbf{x} always lies in the range $[-h, 0]$ to reflect the fact that the array of particle coordinates are reinitialized after every iteration. The implementation of LEA is based on a first order approximation to equation (4):

$$\mathbf{x}^{-h}(p) = \mathbf{x}^0(p) - h\mathbf{v}^t(\mathbf{x}^0(p)) \quad (5)$$

which states that the value of any property of the particle at time t is retrieved from its value at known particle position \mathbf{x}^{-h} at time $t - h$. Additional processing is required to correctly handle general inflow conditions, to avoid a decrease in the noise frequency due to multiple particles originating from the same texel, and to minimize aliasing effects when h is too large [5]. Furthermore, \mathbf{x} and \mathbf{v} can refer to either a 2D or a 3D velocity, but only the 2D case is implemented in [5].

5 ALGORITHM

The desired result is the evolution of a time surface that is correctly deformed in a single direction. For example, the height of the surface is maintained in a fixed 2D array. The height of a particle on the surface at time t at position (i, j) is estimated from the particle's location at time $t - h$. Therefore, we do not continually track a set of particles as is done in standard time surface techniques [10, 12, 15, 17]. At each time step the surface is created using a different set of particles that have identical positions in the x and y directions.

The plane is spanned by the x and y directions, and the vertical direction is along z . $(\mathbf{v}^t(\mathbf{x}, z), w^t(\mathbf{x}, z))$ is an unsteady 3D velocity field at time t , where $\mathbf{x} = (x, y)$ and z define a position within the physical domain.

Figure 3 shows the time surface S^{t-h} at time $t - h$, and the resulting surface S^t after a time interval h . When S^t is computed, a particle p located at $(\mathbf{x}^0(p), z^t(p))$ at time t was located at $(\mathbf{x}^{-h}(p), z^{t-h}(p))$ at time $t - h$. The horizontal motion of the particle satisfies equation (3), taking into account the vertical displacement of the particle

$$\frac{d\mathbf{x}^\tau(p)}{d\tau} = \mathbf{v}^{t+\tau}(\mathbf{x}^\tau(p), z^{t+\tau}(p)) \quad (6)$$

with initial condition $\mathbf{x}^0(p) = (i, j)$. Similarly, the change in the particle height over the time interval $[t - h, t]$ satisfies

$$\frac{dz^{t+\tau}(p)}{d\tau} = w^{t+\tau}(\mathbf{x}^\tau(p), z^{t+\tau}(p)) \quad (7)$$

where $\tau \in [-h, 0]$. Since the particle height is not reinitialized at every iteration, its superscript increases monotonically. However, $\mathbf{x}(p)$ is reinitialized at each iteration to ensure that the surface never self-intersects.

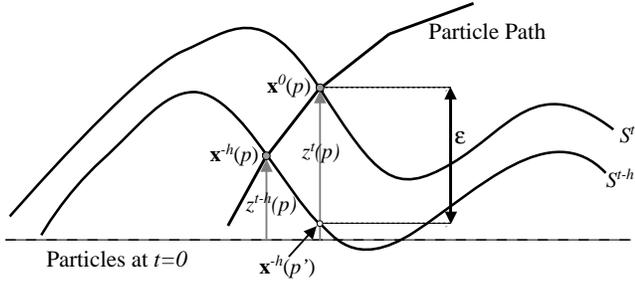


Figure 3: Profile of a surface at two different time steps.

The solution of equations (6-7) over a time interval h satisfies the system of integral equations

$$\mathbf{x}^{-h}(p) = \mathbf{x}^0(p) + \int_0^{-h} \mathbf{v}^{t+\tau}(\mathbf{x}^\tau(p), z^{t+\tau}(p)) d\tau \quad (8)$$

$$z^t(p) = z^{t-h}(p) + \int_{-h}^0 w^{t+\tau}(\mathbf{x}^\tau(p), z^{t+\tau}(p)) d\tau \quad (9)$$

To summarize, S^t is computed given particles on a 2D manifold S^{t-h} . The Lagrangian part of the algorithm is based on two steps. First, particle heights at $\mathbf{x}^0(p)$ are retrieved from the previous time step at $(\mathbf{x}^{-h}(p), z^{t-h}(p))$. Then the vertical coordinate of the particle is computed from equation (7) and integrated forward in time to properly account for its displacement between successive iterations. The term Eulerian refers to the updating of heights at fixed x and y locations, and discarding references to specific particles.

Discrete solutions to Equations (8-9) are found with a first order approximation:

$$\mathbf{x}^{-h}(p) = \mathbf{x}^0(p) - h\mathbf{v}^t(\mathbf{x}^0(p), z^t(p)) \quad (10)$$

$$z^t(p) = z^{t-h}(p) + hw^{t-h}(\mathbf{x}^{-h}(p), z^{t-h}(p)) \quad (11)$$

Unfortunately, these equations are coupled and an explicit solution cannot be found. Equation (10) requires $z^t(p)$ before it is even computed in equation (11). In ocean flows however, changes in the horizontal velocity with respect to height are negligible, and therefore, $z^t(p)$ can be replaced by $z^{t-h}(p')$ without introducing noticeable error. The amount of error introduced is directly proportional to the size of ϵ in Figure 3 and the amount of horizontal motion change. Additional error is introduced at regions of inflow when the backwards integration requires referencing of nonexistent heights. The extent of this error is not known and solutions are not given in this case study.

6 RESULTS

The ocean model data used in this study comes from the Navy Coastal Ocean Model (NCOM) in the Gulf of Mexico. The NCOCM is a three dimensional time-dependent simulation of temperature, salinity, and velocity field. The horizontal resolution of the full model is on a regular grid of 352×320 or $1/20$ of a degree. The vertical spacing is stacked and contains 60 slices, extending from 2.5 meters to 4000 meters below the ocean surface. The top 20 slices are uniformly spaced five meters apart, and the remaining 40 slices are spaced using a logarithmic scale where deeper slices are spaced further apart [8]. By convention the vertical axis z points toward the sky, and $z = 0$ is the ocean surface.

To date, the NCOCM has seven years of simulation completed, and a five month period beginning in April of the seventh model year is used. A region just north of the Yucatan Peninsula extending to Louisiana and eastward to Florida (Figure 4) is focused on. The

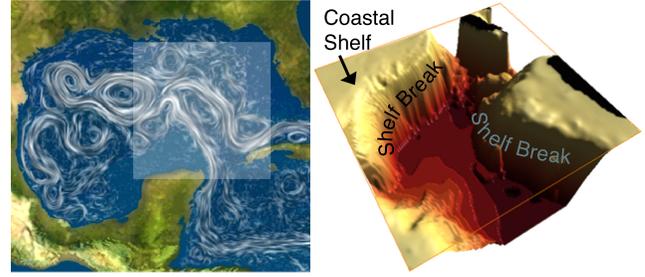


Figure 4: Left: Surface flow of the NCOCM displayed with LIC and region of interest highlighted. The flow is faded according to velocity magnitude (zero magnitude is not shown) and a texture of the surrounding land masses is positioned in the background. Right: Plot of the topography in the selected region marking the coastal shelf and the shelf break. The vertical is scaled by a factor of 100 and the surface is colored by depth. Camera is positioned looking from the northwest.

isolation results in a subset of the NCOCM containing 77 forty eight hour time steps each with a resolution of $161 \times 162 \times 60$.

The vertical structure of the loop current and its associated eddies are displayed by computing time surfaces at 200, 400 and 800 meters below the surface. In each figure, a light is positioned to the left. This produces shadows on the right side of hill and the left side of a dip. Note that a static view from the above is difficult, if not impossible, to distinguish dips from hills on the surface. Real time interaction with the data is required.

The mesh size of the surface in each figure is that of the velocity (161×162), while the texture has a resolution of 350×352 . The discrete time step of the data is 48 hours. To avoid flow structures from moving too quickly, an interpolation time step $h = 6$ hours is used. However, this still causes aliasing effects when using LEA, remedied through the application of LIC to the LEA texture at each time step. In addition, the display is scaled by a factor of 200 in the z direction.

Figure 5 compares the 200, 400, and 800 meter depths after running the simulation for 120 time steps (30 days). These particular images help study the large anticyclonic eddy associated with the loop current. The vertical structure and the life cycle of the eddy is visualized by showing the raising and sinking of the thermocline associated with the eddy genesis and decay. Not much effort has been expended by oceanographers to better understand these vertical structures, possibly due to a lack of adequate visualization tools. The hope is that LETS will help oceanographers formulate a new set of questions, which will eventually lead to a new and better model of the ocean. Dr. Steve Morey at the Center for Ocean-Atmospheric Prediction Studies (COAPS) says that he and other oceanographers cannot always explain all features evident on a given time surface, stating that visualization is not used for just answering questions, but also for finding new questions to investigate.

7 Conclusions

A new technique called Lagrangian–Eulerian Time Surfaces (LETS) has been introduced to help track the evolution of features caused by depth variations of the thermocline. A set of particles are initially placed on a horizontal plane, which is then tracked in time using a hybrid Eulerian–Lagrangian algorithm. A texture computed from the horizontal velocity is mapped to the evolving surface. When added as a post-processing option of Lagrangian–Eulerian Advection (LEA) the amount of time increases by only 12%.

Future improvements to LETS include

- Properly handling inflow regions.
- Incorporate dye advection to detect origin of particles.

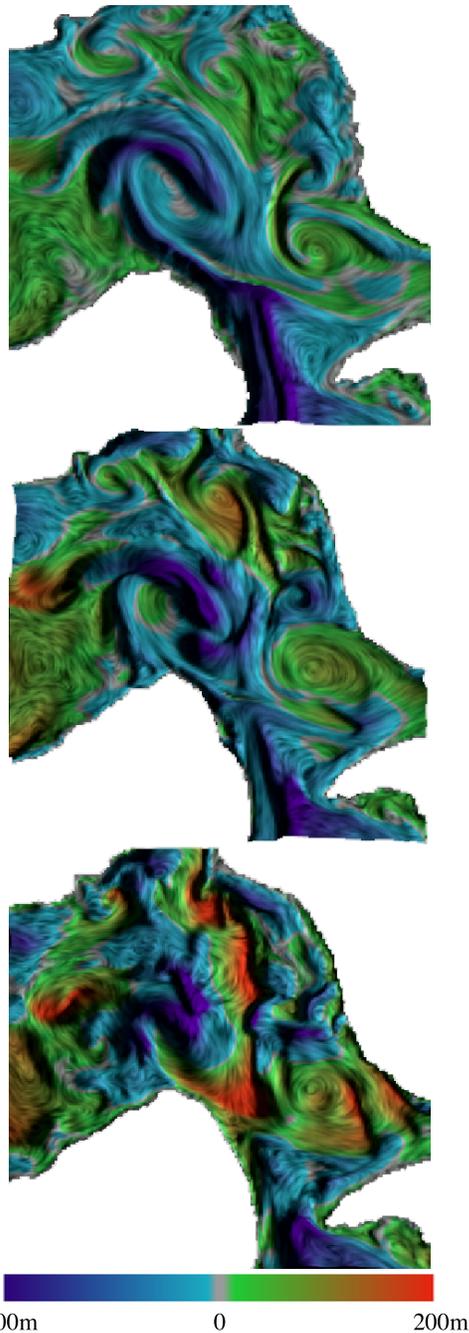


Figure 5: Time surfaces initialized at 200 (top), 400 (middle), and 800 (bottom) meters below the surface and computed after 120 time steps (30 days).

- Hardware implementation of the algorithm for improved frame rates.
- Using the image based flow visualization [16] instead of LEA for improved frame rates.

LETS is not suited for all types of 3D velocity fields. Velocity fields with vertical velocities comparable to the horizontal velocities will produce distorted surfaces very quickly over the time scale of interest, during which the particles cover a sizable fraction of the physical domain. Although scaling along z could be applied to the resulting surface, these velocity fields may be better visualized using other techniques. LETS is meant primarily for velocity

fields where the vertical velocities are at least two to three orders of magnitude smaller than the horizontal velocities.

Acknowledgments

The Center for Ocean-Atmospheric Prediction Studies (COAPS) receives the base of its funding from the National Oceanic and Atmospheric Administration (NOAA) Office of Global Programs. Additional funding for this work was provided by NSF Grant No. 0083792.

References

- [1] B. Cabral and L. Leedom. Imaging Vector Fields Using Line Integral Convolution. In *Proc. of SIGGRAPH-93: Computer Graphics*, pages 263–270, Anaheim, CA, 1993.
- [2] Kelly Gaither and R.J. Moorhead. Visualizing Vector Information in Ocean Environments. In *Oceans '95 MTS/IEEE Proceedings*, pages 1907–1914, San Diego, CA, October 1995.
- [3] M. Grant Gross and Elizabeth Gross. *Oceanography: A View of Earth*, page 207. Prentice Hall, seventh edition, 1996.
- [4] W. Heidrich, R. Westermann, H.-P. Seidel, and T. Ertl. Applications of Pixel Textures in Visualization and Realistic Image Synthesis. In *ACM Symposium on Interactive 3D Graphics*, pages 127–134, ACM, April 1999.
- [5] B. Jobard, G. Erlebacher, and M. Y. Hussaini. Lagrangian Eulerian Advection for Unsteady Flow Visualization. In *Proceedings of IEEE Visualization '01*, pages 53–60, October 2001.
- [6] Andreas Johannsen and Robert J. Moorhead II. AGP: Ocean Model Flow Visualization. *IEEE Computer Graphics & Applications*, 15(4):28–33, July 1995.
- [7] N. Max and Becker B. Flow visualization using moving textures. In *Proceedings of the ICASW/LaRC Symposium on Visualizing Time-Varying Data*, pages 77–87, 1995.
- [8] S.L. Morey, J. Zavala-Hidalgo, and J.J. O'Brien. High-Resolution Ocean Modeling of the Gulf of Mexico. In *Research Activities in Atmospheric and Ocean Modeling*, page In Press. CAS/JSC Working Group on Numerical Experimentation, 2001.
- [9] Scott Nations, Robert Moorhead, Kelly Gaither, Steve Aukstakalnis, Rhonda Vickery, Jr. Warren C. Couvillion, Daniel N. Fox, Peter Flynn, Alan Wallcraft, Patrick Hogan, and Ole Martin Smedstad. Interactive Visualization of Ocean Circulation Models. In Roni Yagel and Gregory M. Nielson, editors, *IEEE Visualization '96*, pages 429–432, 1996.
- [10] F.H. Post and T. van Walsum. Fluid Flow Visualization. In G.M. Nielson H. Hagen, H. Miller, editor, *Focus on Scientific Visualization*, pages 1–40. Springer Verlag, Berlin, 1993.
- [11] Gerik Scheuermann, Holger Burbach, and Hans Hagen. Visualizing Planar Vector Fields with Normal Component Using Line Integral Convolution. In *IEEE Visualization '99*, pages 255–262, San Francisco, 1999. IEEE.
- [12] C. Silva, L. Hong, and A. Kaufman. Flow Surface Probes for Vector Field Visualization, 1997.
- [13] Daniel Weiskopf, Matthias Hopf, and Thomas Ertl. Hardware-Accelerated Visualization of Time-Varying 2D and 3D Vector Fields by Texture Advection via Programmable Per-Pixel Operations. In T. Ertl, B. Girod, G. Greiner, H. Niermann, and H.-P. Seidel, editors, *Vision, Modeling, and Visualization VMV '01 Conference Proceedings*, pages 439–446, 2001.
- [14] R. Westermann, C. Johnson, and T. Ertl. A Level-Set Method For Flow Visualization. In *Proceedings IEEE Visualization '00*, pages 147–154, 2000.
- [15] J.J. van Wijk. Flow Visualization with Surface Particles. *IEEE Computer Graphics & Applications*, 13(4):18–24, July 1993.
- [16] J.J. van Wijk. Image Based Flow Visualization. In *ACM Computer Graphics (SIGGRAPH 2002 Conference Proceedings)*, July 2002.
- [17] J.J. van Wijk, A.J.S. Hin, W.C. de Leeuw, and F.H. Post. Three Ways to Show 3D Fluid Flows. *IEEE Computer Graphics & Applications*, 14(5):33–39, September 1994.
- [18] Z. Zhu and R. J. Moorhead. Extracting and Visualizing Ocean Eddies in Time-Varying Flow Fields. In *7th International Symposium on Flow Visualization*, pages 206–211, Seattle, WA, September 1995.