Visualizing flow fields using fractal dimensions

Ross Vasko, Han-Wei Shen and Rephael Wenger

Abstract— Streamlines are a popular way of visualizing flow in vector fields. A major challenge in streamline visualization is selecting the streamlines for visualization. Rendering too many streamlines clutters the visualization and makes features of the field difficult to identify. Rending too few streamlines causes viewers to completely miss features of the flow field not rendered.

The fractal dimension of a streamline represents its space-filling properties. To identify complex or interesting streamlines, we build a regular grid of scalar values which represent the fractal dimension of streamlines around each grid vertex. High fractal dimension indicates vortices or turbulent regions. We use this scalar grid both to filter streamlines by fractal dimension and to identify and visualize regions containing vortices and turbulence. We describe an interactive tool which allows for quick streamline selection and visualization of regions containing vortices and turbulence.

Index Terms—Streamlines, fractal dimension.

1 Introduction

A vector field in 3D is a map $f : \mathbb{R}^3 \Rightarrow \mathbb{R}^3$. A flow field is a specific type of vector field that represents the flow of some fluid. Each point in a 3D flow field represents the direction and rate of mass transport of a flow. Flow fields are used in several different fields of science and engineering. They can be used to represent weather patterns, air flow during wind tunnel tests, or blood flow. Flow field data sets contain several different features, such as vortices, and have regions with different types of flow behavior. Visualizing these different regions and features is crucial to understanding the flow data.

A common way to visualize flow fields is with streamlines. A streamline is a curve that is tangent to the velocity vector of the flow field at each point of the curve. Streamlines are computationally inexpensive to generate and allow a viewer to see the behavior of a region of the flow field. A large challenge in using streamlines to visualize flow fields is choosing which streamlines to display. If too many streamlines are displayed, the visualization quickly becomes cluttered and interesting features or behavior of the flow field become hard to identify. If an arbitrary sampling is chosen to display less streamlines, some interesting features of the flow field could potentially not be displayed at all.

The geometric characteristics of streamlines are directly representative of the properties of the underlying flow field. To search for various flow field properties, it is possible to search for streamlines that would have the corresponding geometric characteristics. However, the strict defintion of such characteristics is often difficult and can make the search for streamlines too restrictive or misleading. We attempt to extract streamline characteristics with a measurement that does not seek specific geometric characteristics. Rather, we only attempt to classify the complexity of the streamline by how densely it fills a space using the previously defined *box-counting ratio*. Under the box-counting ratio measurement in 3D, the measurements can range from zero to three. The closer to three that a measurement is, the more densely the object fills the 3D space. Simple features and regions such as a laminar flow will generate streamlines with a box-counting ratio near one, due to how straight line does not fill a space densely. As the flow field displays more turbulent behaviors and generates streamlines that fill a space more densely, the box-counting ratio will increase towards three. For streamlines that densely fill some 3D space, such as vortices, the box-counting ratio can increase past two.

With this measurement, we are then able to categorize streamlines based on their complexity without forcing a restrictive definition. Such a measurement allows for filtering to remove clutter while still retaining important or defining flow field features. Additionally, this measurement allows for not only complexity measurement to be assigned to the streamlines, but complexity measurements can also be assigned to the flow field itself. A scalar grid can be constructed with values that are representative of the complexity of the streamlines in some nearby neighborhood. This scalar field allows us to easily identify which regions of the flow field are complex. Additionally, a variety of visualization techniques can be applied to the scalar field to allow the user different insight on the data.

The box-counting ratio is further described and defined in Section 3. The steps to then apply the box-counting ratio to measure streamline complexities and compute a scalar grid that reflect the flow field complexity are contained in Section 4. Visualization techniques that are gained using these complexity measurements are described and shown in Section 5. Full examples of flow field visualizations for various data sets are given in Section 6. Evaluations for this filtering and visualization method are given in Section 7.

2 Related Work

There has already been a considerable amount of work done in streamline filtering and flow field feature identification, but many of these methods require prerequiste knowledge about the flow field or depend on restrictive definitions of such features. In one of the first works on 2D streamline seeding, Turk and Banks [1] seed streamlines to meet a required density using energy minimization functions to aid in the streamline generation. This work was extended into 3D in [2] by using similar energy minimization techniques on streamlines embedded in surfaces in 3D. Mebarki et al. [3] use an approach to create similar visualizations by seeding new streamlines a far distance away from already seeded streamlines. While these techniques create clear and evenly spaced streamlines visualizations for surfaces, they are difficult to generally extend into 3D. Additionally, these do not address further filtering techniques to remove uninteresting streamlines from the visualization.

McLoughlin et al. [4] seek to reduce streamline clutter by requiring a predefined rake and then removing streamlines along that rake that exhibit significant amounts of similarity to reduce redundency and clutter. View points of streamlines are evalutated in [5] by analyzing the amount of occlusion present in a viewpoint. Streamlines can then be added or removed depending on the amount of occlusion in an area. Measuring the amount of information conveyed by a streamlines is presented in [6] using concepts of entropy. With this measurement, more streamlines are required to be seeded from complex regions such as vortices in order to properly capture the information in the regions. This method is built on in [7] to also use the entropy measurements to evaluate view points of streamlines.

Critical points of the flow field are analyzed in [8] and different seeding strategies are used depending on the behavior near the critical

The Ohio State University. E-mail: vasko.38@osu.edu, shen.94@osu.edu and wenger.4@osu.edu

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 2 November 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

points. In [9], various definitions are constructed for different flow field features. If the flow field meets the defined criterea, it is labeled with a true boolean value as having the defined feature. Similarly, various features are defined in [10] and the vectors of the flow field are examined to see if they meet the requirements of various properties. There is also a significant amount of work done on identifying and extracting vortices in the flow field. In all [11], [12], and [13] features of a vortex are defined and then searched for in the flow field. While all of these methods can produce reasonable results in feature extraction, this search for such defined features can be misleading or limiting. We wish to instead use a method of feature identification that does not depend on such a strict definition. [14] does segment the flow field generally, but this method does not supply a way of quantifying the complexity of flow field regions.

We attempt to avoid any strict definition of flow field features to prevent restricting our algorithm to only find specific features. Instead, we treat the complexity of a streamline as how it fills a space. In a previous work by Chaudhuri et al. [15], he introduces measuring the streamline complexity with fractal dimensions to oberserve behavior at different scales. The motivation for moving to such a measurement was to focus solely on the geometry of the line and to prevent limiting oneself to only searching for some certain fluid property. By setting parameters in the box counting ratio, the fractal dimension measurements are used in this work to be able to the identify streamline features at different scales. Features in the flow field can then be organized by their complexity as well as what scale the feature appears in. Using the box counting ratio, we are able to both have a general definition of complexity and provide scalar values that are representative of the flow field complexity.

3 Box Counting Ratio

To address this challenge, we look for a way to identify interesting or complex streamlines or regions of the flow field. The box-counting dimension measures how a densely a set fills a metric space. The box counting dimension of a set S is defined as:

$$dim(S) = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log(\varepsilon^{-1})} \tag{1}$$

Where $N(\varepsilon)$ is the minimum number of boxes needed to cover the set. Using a previously proposed modified definition of the box counting dimensions, we will measure how streamlines fill a space to determine their complexity.

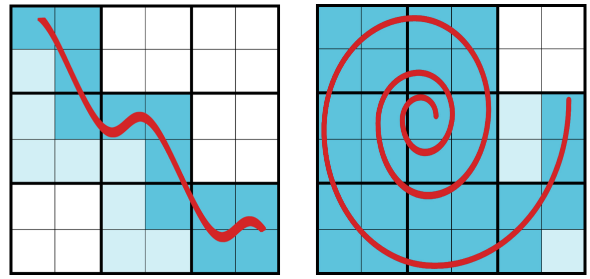
The box counting ratio does not calculate an integer dimension for a set. Rather, depending on how the set fills a space, the dimension is some real number between 0 and 3 for a space filling fractal in 3D. But, because our streamlines generated from a simulation are simply a series of connected straight line segments, this strict definition would meausre each streamline as having a fractal dimension as 1. Regardless, calculating this limit exactly becomes infeasible to do on a computer for each streamline. Instead, we use an alternative definition that allows us to simply approximate the box counting ratio.

Khoury and Wenger [16] defined a box counting ratio approximation in a previous work on examing the properties of isosurfaces using fractal dimensions. The idea of this new approximation is to sample an object at two different resolutions and then estimate the box counting ratio using these two measurements. The new approximated box counting ratio of a set S is defined as:

$$dim_{approx}(S) = \log_2 \frac{N(\frac{\varepsilon}{2})}{N(\varepsilon)} \tag{2}$$

where $N(\varepsilon)$ in this case is simply the number of boxes that the set will intersect on a fixed grid.

We then measure the fractal dimension of the streamlines by defining a fixed grid, counting the number of boxes that the streamlines intersects at the resolution of $\frac{\varepsilon}{2}$ and ε, and then solving for the final box counting dimension. With this new formula, we expect that the straight and simple streamlines will have a fractal dimension near 1, as they do not have any space filling properties. As the streamline becomes more complex and fills a 3D region more densely, we expect the fractal dimension to increase exceed 2. These measurements are illustrated in Fig. 1



(a) simple line with box counting ratio of $\log_2 \frac{12}{5} = 1.26$ (b) complex line with box counting ratio of $\log_2 \frac{29}{8} = 1.86$

Fig. 1: Example of box counting ratio calculations on streamlines.

4 Streamline Complexity Grid

For each regular point p in a vector field $f : \mathbb{R}^3 \Rightarrow \mathbb{R}^3$, let ζ_p be the unique streamline passing through p. Define $\phi_w(p)$ as the local box counting ratio of ζ_p in a $w \times w \times w$ region around p, where boxes have edge lengths 1 and 2. Function ϕ_w defines a scalar field on the regular points of f. The scalar $\phi_w(p)$ represents the complexity of a streamline through p in a $w \times w \times w$ neighborhood of p. We call the scalar field ϕ_w, the streamline complexity field of f.

To compute the set of streamlines, we first seed a streamline from each regular point voxel v in our vector field f. We then calculate the box counting dimensions of each of these streamlines using the previously described box counting dimension algorithm. If the streamline intersects more than a predefined cutoff of c_s boxes of edge length 2, then the streamline is stored. If the streamline does not intersect more than c_s boxes, it is discarded.

To compute the local box counting ratio of ζ_v in a $w \times w \times w$ region centered at v, we take a subset of the streamline, ζ_s, that precisely includes the portion of ζ_v that is within the $w \times w \times w$ region. The box counting dimension of ζ_s is then computed using the previously described box counting dimension algorithm. Similarly to the streamline generation step, it is also ensured that ζ_s more than c_v boxes of edge length 2. If ζ_s does meet the cutoff requirement, the box counting dimension is considered a value estimate, if not, the value is discarded.

To construct a scalar grid representing the streamline complex field of f, a set of streamlines are generated as previously described. For each streamline ζ_v that intersects voxel v, we compute the local box counting dimension of ζ_v. If ζ_v meets the cutoff requirement, the resulting box counting dimension is stored in a list with all other box counting dimensions calculated for v. Once the local box counting dimension of all of the streamlines that intersect v have been calculated, we take the P^{th} percentile of the local box counting dimension list and store this value as v's complexity value. Let the streamline whose local box counting dimension was measured as the P^{th} percentile local box counting dimension for voxel v be denoted as v_ζ. We call the resulting scalar grid, the streamline complexity grid of f.

Additionally, a scalar value is associated with each streamline representing the streamline's complexity. For a single streamline ζ_v, its local box counting dimension is measured at each voxel that it intersects. The highest of these local box counting dimension calculations is stored as the complexity value of ζ_v .

5 Visualization Techniques

The streamline complexity computations can be used in a variety of ways to filter streamlines of varying complexities or to provide a method of visualization for the flow field.

Fig. 2: Example of the streamline filtering.

Fig. 3: Example visualizations using the colored plane.

Fig. 4: Example visualizations using the colored plane.

5.1 Streamline filtering by value

The user is able to choose two values, a and b where $a < b$, and only display streamlines with a complexity inbetween the chosen values. Specifically, the streamline ζ_p will be displayed only if $a \leq \phi_w(p) \leq b$. The constants allow the user to choose the level of streamline complexity that they will view. By choosing constants of values near 1, streamlines with a low box counting ratio and smooth flow will be displayed. By choosing constants of values above 2, streamlines with a relatively high box counting ratio and turbulent or complex flow will be displayed. Filtering by complexity value is shown in Fig. 2

5.2 Local complexity maximums

A significant amount of clutter in the streamline display will remain if additional filtering methods are not considered. Several streamlines in the visualization will be visually similar or provide redundant information. We are able to only show the local maximum streamlines to filter streamlines that all represent a single region or feature of the flow. Local maximum filtering will show streamline v_ζ only if for all of the 8 points q that directly neighbor v on the complexity grid, $\phi_w(v) > \phi_w(q)$. This method allows for single streamlines representatives to be shown for each feature or region rather than several, cluttered streamlines.

5.3 Colored plane

A colored plane can be used to allow the user to visualize the scalar complexity grid, ϕ_p, directly. A color gradient from blue to green to red is able to be defined and mapped to values in the range 0 to 3, for each of the possible box counting dimensions ratios. Low scalar values will be displayed as blue colors, while high scalar values will be displayed as red colors. A plane is then defined on the scalar complexity grid and each point on the plane is colored from this defined color gradient. The user is able to control the plane through the scalar complexity grid to identfiy regions varying complexity in the grid. Once regions of interest are identified through the color plane, the user can display streamlines near that region to understand its behavior.

5.4 Complexity samplings

The streamlines seeded from the region of the colored plane can be viewed in various samplings to show the different levels of complexity around the plane. The scalar values on the colored planes are partitioned into two sets, the one-third highest scalar values and the two-thirds lowest scalar values. Two sets of streamlines are also constructed. For the voxels that the plane intersects, $S_{top} = \{v_\zeta | \phi_w(v)$ is in the top one-third scalar values$\}$ and $S_{bottom} = \{v_\zeta | \phi_w(v)$ is in the bottom two-thirds scalar values $\}$. Two values are then defined by the user to determine the percentages of S_{top} and S_{bottom} displayed. An example of the plane visualizations are shown in Fig. 3.

5.5 Isosurfaces:

An isosurface can be used to highlight regions of the flow field with a high complexity. An isosurface, which is defined by

$$\{x \mid \phi_w(x) = \sigma\} \tag{3}$$

where σ is the isovalue and ϕ_w is the streamline complexity field, will seperate all scalar values above σ from the values below σ on the streamlines complexity. This isosurface will enclose the regions of streamlines with a fractal dimension higher than the σ value and provide a simple way to identify regions of a defined complexity. At particularly high σ values, the isosurface will only enclose vortices and turbulent features of the flow field that the user may have otherwise missed.

5.6 Streamline gradient magnitudes:

The gradient magnitudes of the streamline complexity grid can be calculated to create a new gradient magnitude scalar grid ϕ_g. The scalar $\phi_g(x)$ is given by $\|\nabla\phi_p(x)\|$. Another isosurface can be used to visualization the function ϕ_g to identify regions of high change of turbulence or complexity. Vortices in the flow field tend to have high complexity values recorded near their centers, with values quickly decreasing towards their boundaries. When a high isovalue is chosen for the gradient magnitude isosurface's isovalue, the isosurface will highlight these isolated regions of turbulence or turbulent regions that quickly become smooth. Example of isosurfaces of the scalar complexity values and gradient magnitudes are shown in Fig. 4.

6 Examples (or Results?)

In this section we will demonstrate how the previous described techiques can enhance the visualization of large flow field data sets. The previously decribed algorithms and techniques are applied to two different data sets, a Solar Plume data set and a Hurricane Isabel data set. Explanations and figures are given to show the advantages and features given by our algorithms.

The algorithm was implented in C++ using The Visualization Toolkit (VTK) across two seperate programs. The first program has been created to generate streamlines, computer complexity measurements, and compute the streamline complexity grid and then output this information into files. Then another program takes the output files as input and allows the user to interact with the streamlines and the streamline complexity grid. The second program allows for a realtime interactivity with the flow data set.

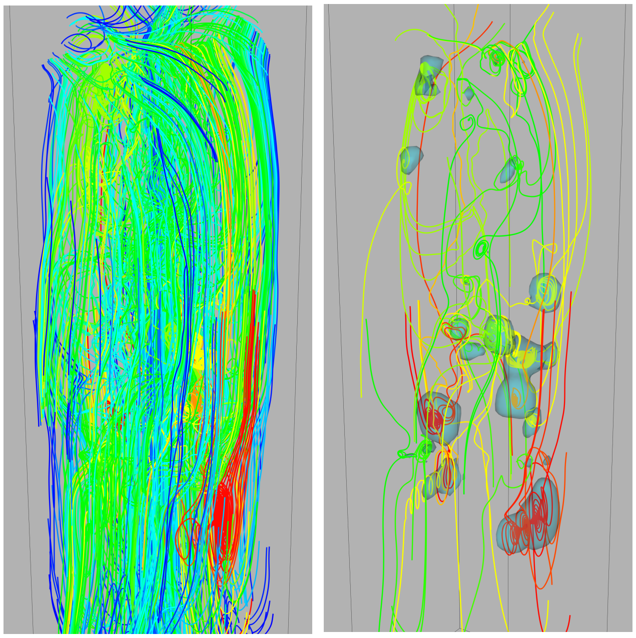
6.1 Solar Plume

The Solar Plume data is a $126 \times 126 \times 512$ vector field data set that was provided by (who?). This data set has many different regions of varying flow complexity. Central regions of the flow tend to be more laminar with vortices near the boundaries of the data set. These regions are able to be identified and highlighted by our algorithm.

A regular sampling of streamlines from the Solar Plume data set is displayed in Fig. 5.a. From the figure it is apparent that due to the size of this data set that a regular sampling is not suitable for this data set. The streamlines cause severe occlusion and none of the strucutre of the flow field can be understood. A filtering of the streamlines produced by considering the complexity measurements of the streamline given by our algorithm is displayed in Fig. 5.b. The filtering drastically reduces the clutter in the visualization and makes the regions of interest much more clear to the user. To filter the streamlines in the plume data set, we are using the local maximum filtering as well as only showing the streamlines above a complexity value near 1.6. The streamlines are also colored by complexity, with the red streamlines being the most complex and oftern contain vortices. The green and less complex streamlines only contain mild turbulence through the line. The isosurface is in this rendering is set to a value near 1.6 to enclose the complex regions of the line.

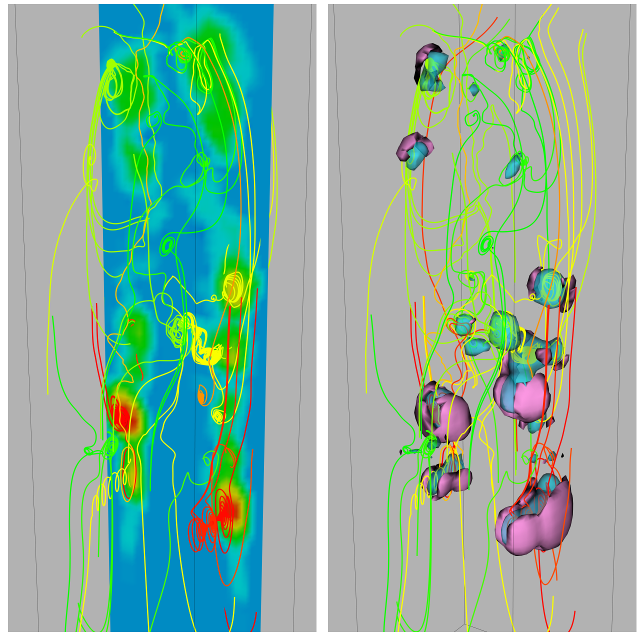
Additionally, other visualization techniques are able to be applied to the plume data set. The color plane allows the viewer to see that the Solar Plume plume consists of a mostly smooth behavior towards its center and then becomes more turbulent near to edges. Rather than manually attempting to identify which regions of the flow field exhibit specific types of behavior, this visualization allows the viewer to identify these regions quickly. A plane can be rendered to show the complexity values of the scalar field and the gradient isosurfaces can be rendered to show the areas of high change of turbulence. These additional feautres for the Solar Plume data set are displayed in Fig 6

A histogram with streamline complexity values are shown in Fig. 7. The histogram shows further examples from the Solar Plume data set of streamlines at different complexity values. As the complexity



(a) Streamlines seeded regularly through the Solar Plume data set. (b) The Solar Plume streamlines filtered by complexity mesasurements.

Fig. 5: Example of how filtering increases the visibility of flow field features in the Solar Plume data set. In (a), the severe occlusion prevents any the viewer from seeing any structure in the flow field. (b) contains the same data set of streamlines as in (a), but with the filtering by complexity values applied. The number of streamlines has been significantly reduced and regions of high complexity are now clear to the viewer. Additionally, vortices in the flow field are enclosed by isosurfaces in (b).



(a) High complexity areas highlighted by the colored plane. (b) Regions of high complexity change highlighted by an isosurface.

Fig. 6: Example of additional filtering techniques. In (a), makes a distinction between the low complexity regions and high complexity regions. The gradient magnitude isosurfaces are displayed in (b). The isosurfaces form a boundary around the voritces in the plume data set.

value increases, the streamline becomes much less common and more difficult to find without a strategic method of filtering.

6.2 Hurricane Isabel

The Hurricane Isabel data set is a $500 \times 500 \times 100$ and used as a data set in a 2005 IEEE Visualization contest. The data set consists of a mostly laminar flow along with several different isolated vortices within the flow. Again, our algorithm is able to seperate these regions and highlight the significant sections to a view.

A regular sampling of streamlines from the Hurricane Isabel data set is displayed in Fig. 8.

7 Evaluation

8 Summary and Future Work

References

[1] G. Turk and D. Banks, "Image-guided streamline placement," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 453–460. [Online]. Available: http://doi.acm.org/10.1145/237170.237285

[2] X. Mao, Y. Hatanaka, H. Higashida, and A. Imamiya, "Image-guided streamline placement on curvilinear grid surfaces," in *Visualization '98. Proceedings*, Oct 1998, pp. 135–142.

[3] A. Mebarki, P. Alliez, and O. Devillers, "Farthest point seeding for efficient placement of streamlines," in *Visualization, 2005. VIS 05. IEEE*, Oct 2005, pp. 479–486.

[4] T. McLoughlin, M. Jones, R. Laramee, R. Malki, I. Masters, and C. Hansen, "Similarity measures for enhancing interactive streamline seeding," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 19, no. 8, pp. 1342–1353, 2013. [Online]. Available: http://www.sci.utah.edu/publications/Mcl2013a/McLoughlin_TVCG2013.pdf

[5] S. Marchesin, C.-K. Chen, C. Ho, and K.-L. Ma, "View-dependent streamlines for 3d vector fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1578–1586, Nov 2010.

[6] L. Xu, T.-Y. Lee, and H.-W. Shen, "An information-theoretic framework for flow visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1216–1224, Nov 2010.

[7] T.-Y. Lee, O. Mishchenko, H.-W. Shen, and R. Crawfis, "View point evaluation and streamline filtering for flow visualization," in *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, March 2011, pp. 83–90.

[8] X. Ye, D. Kao, and A. Pang, "Strategy for seeding 3d streamlines," in *Visualization, 2005. VIS 05. IEEE*, Oct 2005, pp. 471–478.

[9] T. Salzbrunn and G. Scheuermann, "Streamline predicates," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 6, pp. 1601–1612, Nov 2006.

[10] E. Heiberg, T. Ebbers, L. Wigstrom, and M. Karlsson, "Three-dimensional flow characterization using vector pattern matching," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 3, pp. 313–319, July 2003.

[11] I. A. Sadarjoen and F. H. Post, "Geometric methods for vortex extraction," in *Data Visualization99*. Springer, 1999, pp. 53–62.

[12] I. Sadarjoen, F. Post, B. Ma, D. Banks, and H.-G. Pagendarm, "Selective visualization of vortices in hydrodynamic flows," in *Visualization '98. Proceedings*, Oct 1998, pp. 419–422.

[13] J. Zhong, T. Huang, and R. Adrian, "Extracting 3d vortices in turbulent fluid flow," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 2, pp. 193–199, Feb 1998.

[14] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. Joy, "Topological segmentation in three-dimensional vector fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 10, no. 2, pp. 198–205, March 2004.

[15] A. Chaudhuri, T.-Y. Lee, H.-W. Shen, and R. Wenger, "Exploring flow fields using space-filling analysis of streamlines," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 10, pp. 1392–1404, Oct 2014.

[16] M. Khoury and R. Wenger, "On the fractal dimension of isosurfaces," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1198–1205, Nov 2010.

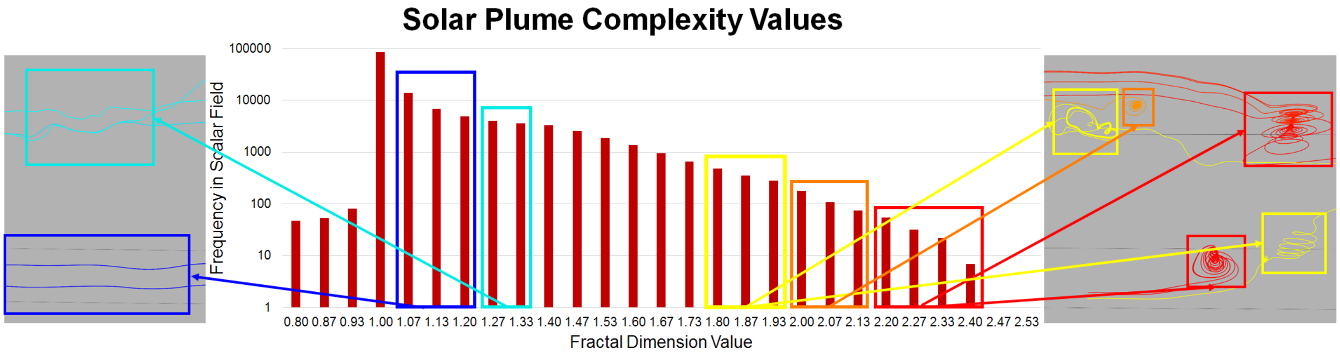
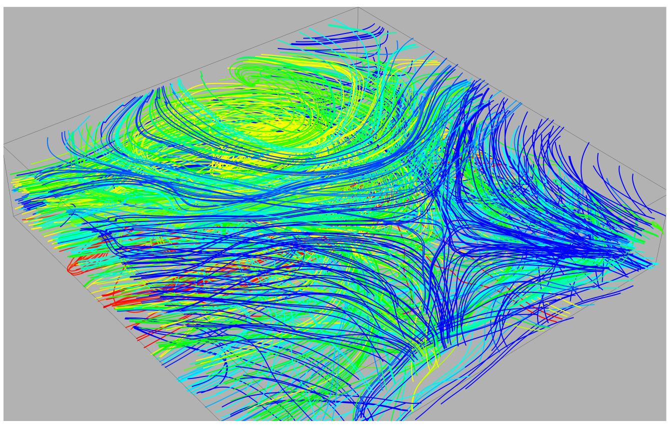
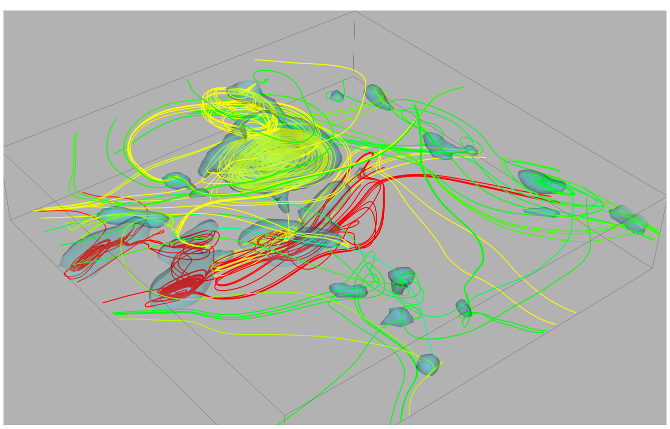


Fig. 7: Histogram of different complexity values and examples of streamlines at those different complexity values.



(a) Streamlines seeded regularly through the Hurricane Isabel data set.



(b) The Hurricane Isabel streamlines filtered by complexity mesasurements.

Fig. 8: Example of how filtering increases the visibility of flow field features in the Hurricane Isabel data set. In (a), the visualization is taken over by the simple, laminar streamlines. (b) contains the same data set of streamlines as in (a), but with the filtering by complexity values applied. The streamlines have now been significantly filtered so that the vortices in the data set are highlighted. These vortices are highlighted by isosurfaces.