

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 function $f : \mathbb{R}^3 \Rightarrow \mathbb{R}^3$. 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 a velocity vector of the fluid. Flow fields are used in several different fields of science and engineering. They can be used to represent storms, air flow during wind tunnel tests, or blood flow. Effectively visualizing flow fields are crucial to their interpretation. An ideal visualization will communicate the structure and physics of the flow field. Unfortunately, identifying meaningful and interesting structures is often computationally difficult due to the size of the flow field data sets.

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. But, 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.

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 densely a set fills a metric space. The box counting dimension of a set S is defined as:

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

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.

Once we have these complexity measurements of streamlines around the flow field, we build a scalar field of complexity values. This scalar field allows us to easily identify which regions of the flow field are complex. Also, using this scalar field, we can apply a variety of visualization techniques to allow the user different insight on the data.

*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.*

2 RELATED WORK

In a previous work by Chaudhuri, he introduces measuring the streamline complexity with fractal dimensions to observe 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 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.

3 BOX COUNTING RATIO

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 measure 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 defined a box counting ratio approximation in a previous work on examining 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:

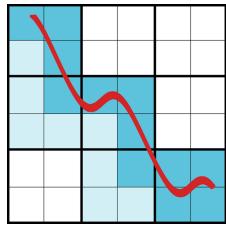
$$\dim_{approx}(S) = \log_2 \frac{N(\frac{\varepsilon}{2})}{N(\varepsilon)} \quad (2)$$

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

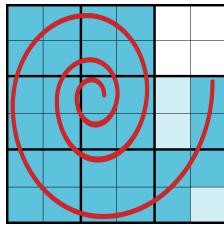
We then measure the fractal dimension of the streamlines by defining a fixed grid, counting the number of boxes that the streamlines intersects at the resolution of $\frac{\varepsilon}{2}$ and ε , and then solving for the final box counting dimension. With this new formula, we expect that the straight 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

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



(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.

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 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 .

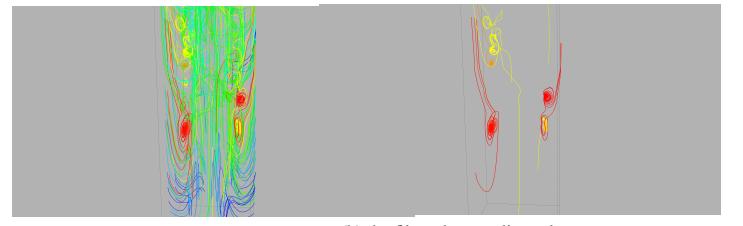
Additionally, a scalar value is associated with each streamline representing the streamline's complexity. For a single streamline ζ_v , its local box counting dimension is measured at each voxel that it intersects. The highest of these local box counting dimension calculations is stored as the complexity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

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 between 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.

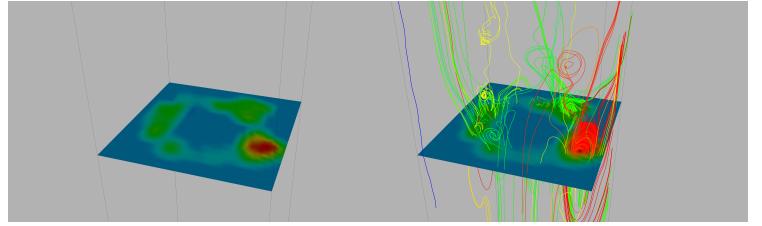
Local complexity maximums: A significant amount of clutter in the streamline display will remain if additional filtering methods are not considered. Several streamlines in the visualization will be visually similar or provide redundant information. We are able to only show the local maximum streamlines to filter streamlines that all represent a single region or feature of the flow. Local maximum filtering



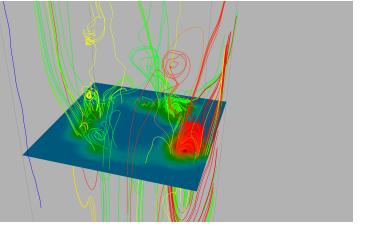
(a) the cluttered view of all of the streamlines in the dataset

(b) the filtered streamlines above a certain complexity value

Fig. 2: Example of the streamline filtering.



(a) the plane indicating the regions of high complexity by color



(b) the high complexity lines near the defined plane.

Fig. 3: Example visualizations using the colored plane.

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.

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 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 identify 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.

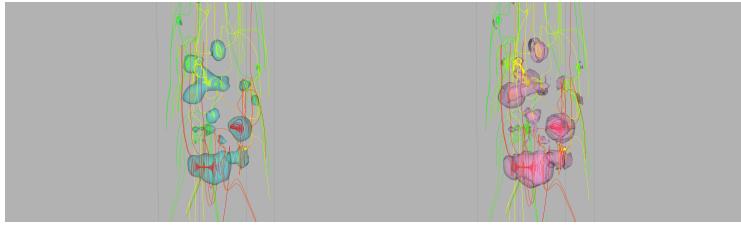
Complexity samplings The streamlines seeded from the region of the colored plane can be viewed in various samplings to show the different levels of complexity around the plane. The scalar values on the colored planes are partitioned into two sets, the one-third highest scalar values and the two-thirds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

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

$$\{x | \phi_w(x) = \sigma\} \quad (3)$$

where σ is the isovalue and ϕ_w is the streamline complexity field, will separate 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.

Streamline gradient magnitudes: The gradient magnitudes of the streamline complexity grid can be calculated to create a new gradi-



(a) the scalar value isosurface enclosing high complexity regions (b) the gradient isosurface enclosing regions of high complexity change.

Fig. 4: Example visualizations using the colored plane.

ent 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

7 EVALUATION

8 SUMMARY AND FUTURE WORK