Exploring Flow Fields Using Fractal Analysis of Field Lines

Abon Chaudhuri, Teng-Yok Lee, Student Member, IEEE, Han-Wei Shen, Marc Khoury, and Rephael Wenger

Abstract—Streamline based techniques are widely used to visualize flow fields produced in large scale scientific simulations. As the shape of a streamline is often related to the underlying property of the given flow field, it is important to identify streamlines (or streamline segments) that have unique geometric features. The Kolmogorov capacity or box counting dimension is a standard metric for analyzing streamlines in turbulent flow. In this paper, we introduce a related metric, called the box counting ratio, for measuring complexity of discretized approximations to streamlines.

Besides the metric, we propose a novel feature-based visual analytic framework for organizing and representing important segments extracted from large number of streamlines. We utilize the box counting ratio to identify and segment the interesting parts of the streamlines. Then we construct high dimensional feature vectors from segmented streamlines, and finally project the feature vectors to a 2D space. From this space, the user can drive visualization of selected features in the original spatial domain, which is more clutter-prone and less navigation-friendly. In short, our framework enables the user to easily interact with the features otherwise hidden in large data. We strengthen our claims with elaborate examples and case studies using real-world combustion and climate data sets.

1 INTRODUCTION

Large scale simulations in various scientific and engineering disciplines such as climate modeling, computational fluid dynamics and automobile design produce enormous vector fields. Analysis and effective visualization of these vector fields play a critical role in discovery of knowledge from such simulations. However, sheer size of the fields often impedes direct interactive exploration. In such cases, fast parallel programs are employed to compute a large number of streamlines, which can be further analyzed to understand the vector field. Streamline based techniques are widely used for exploring flow fields, because the streamlines’ geometry characterizes the local flow features, such as, vortex, turbulence etc. A streamline segment going through a turbulent region of a flow field should have significantly different shape from a segment passing through regions of uniform or laminar flow. To understand flow fields better, this close relation between a streamline’s geometric features and the corresponding flow features needs to be explored in further depth.

A collection of streamlines which have sampled the flow field densely enough is representative of the field. Given such a collection, the users, who are mostly domain scientists, aim at identifying and selecting out the streamlines (or streamline segments) that represent interesting flow features. When a precise definition of the feature is not known, they look for methods to organize and possibly cluster the streamlines based on their geometric properties. A geometry-based metric which reflects whenever the streamline enters an interesting region of the flow field can be helpful for this purpose. However, such a metric is not readily available for use. Some metrics such as average curvature and winding angle have been used with limited success. Hence, the search for a suitable metric (or a suite of metrics) which can differentiate between streamlines representing different flow features is still active.

In this paper, we introduce a metric called box counting ratio for effective geometric analysis of streamlines. This metric is related to the Kolmogorov capacity or box counting dimension, which was introduced by A. Kolmogorov for analyzing dynamical systems in general [1]. The box counting dimension is one way of defining a “fractal dimension” of a set of points. It can be applied to streamlines because streamlines are known to have some self-similarity near vortex-like points [2]. Another study has shown that the self-similarity of streamlines persist over a considerable scale space [3]. Vassilicos and Hunt in [2] explain how the box counting dimension measures vortices in turbulent flows. Numerous other papers apply the box counting dimension to measure and analyze turbulent flow, e.g. [4], [5], [6], [7].

The box counting dimension can be interpreted as a measure of how densely a fractal fills up its embedding metric space. In our studies, we observed that some flow features such as vortex and turbulence are represented by streamlines moving along complex
curvilinear paths confined in a region, leading to a dense structure. On the other hand, fairly straight, or winding but sparse streamlines may represent different types of features such as laminar flow. This variation of compactness or space-filling capacity among different features is well captured by the box counting ratio. To be more specific, a streamline embedded in a 3D space can have a box counting ratio between 0 and 3. Intuitively, each small segment of a streamline is always a one-dimensional linear object and should have box counting ratio close to 1. But depending on the local field properties, a larger segment may fill up the space in a more complex manner. As a result, its shape can approximate a higher dimensional object such as a surface or a volume.

Box counting ratio offers a number of advantages in analyzing and visualizing streamlines:

- It measures the compactness of the geometry without having to know its exact shape. Any structure which is geometrically complex is captured by this. So, our method is not limited to detection of any particular type of feature and can be used in applications where exact definition of a feature is not available.
- Streamlines often pass through multiple regions of interest and contain more than one features, possibly of different sizes. Box counting ratio can be employed to detect features of varying sizes.
- The value of the box counting ratio has a geometric interpretation. A value close to 1 indicates that the geometry of the streamline is close to being linear. Similarly, box counting ratio close to 2 means the streamline densely fills up a 2D subspace of the 3D space and resembles a surface. Or, it may represent a relatively sparse 3D space. Finally, a value close to 3 indicates that the streamline forms a really dense 3D structure. So, the value of our proposed metric not only quantifies the importance of a geometric feature, it also speaks of the nature of the feature.

Besides using box counting ratio for measuring complexity of streamlines, we also present a visual analytic framework for feature based exploration of flow fields, which is another contribution of this paper. In the proposed framework, a collection of streamlines or segments, selected based on their complexity, are represented as high dimensional feature vectors. These feature vectors are projected back to the 2D screen space using suitable dimensionality reduction techniques. The purpose of this projection is to provide an interactive visual space, which meaningfully places features, to the user. Without having to navigate through a 3D vector field which is less friendly and often impossible to navigate, the user can brush regions of potential interest on this 2D space to see the corresponding streamlines in a linked display.

The following sections of the paper are organized as follows: Section 2 summarizes the research in fractal dimension and exploratory vector field visualization. Section 3 introduces box counting ratio and discusses how to compute it for discretized streamlines. Section 4 describes the interactive visual analytic framework for large number of streamlines, including feature localization and streamline segmentation at different scales. We present results from various datasets and report performance in Section 5, followed by limitations and conclusion in Sections 6 and 7.

2 RELATED WORK

2.1 Fractal Dimension

Fractal dimension of streamlines and its relation to various fluid properties have been studied in Physics and Fluid Dynamics area. Vassilicos et al. [2] identify spirals as k-fractals, which are locally self-similar structures in a flow field, and analytically measure their Kolmogorov capacity, which is also known as box counting dimension. In another work, box counting dimension is employed as a finer selection criterion on streamlines already filtered based on persistently strong curvature [7]. Another work shows that permeability of a porous medium is dependent on the fractal dimension of the pore area and the tortuosity dimension of the streamlines, where both are estimated by box counting method [3], [8]. Braun et al. [4] suggest that streamlines develop fractal properties due to constant stretching and folding of the vector field and such properties can be evaluated using box counting algorithm. Recently, the box counting ratio was defined and used by Khoury and Wenger [9] to measure the complexity of isosurfaces.

Unlike the fluid dynamics literature, our main focus is not to explore the relation between the box counting dimension and a particular fluid property. Our method leverages a fractal dimension based metric to enable interactive flow feature exploration. We use it to quantify the geometric complexity of streamlines and take it further to automatically locate features and visually present streamline segments corresponding to the complex regions.

2.2 Complexity Analysis of Flow Fields

Besides fractal dimension, a variety of techniques with different goals has been proposed to evaluate a streamline’s complexity. This includes an information theory based approach [10] for hybrid visualization of scalar and vector data, an image-space method for automatic view selection given a set of streamlines [11] and an automatic method for dynamic streamline selection given a viewpoint [12]. Apart from these, multiple techniques [13], [14], [15] exist for detection of a particular type of feature called vortex core.
Fig. 1: (a) Demonstration of computation of box counting ratio. **Left.** A fairly straight streamline yields ratio 1 or close to 1. **Middle.** Depending on complexity, the ratio may increase. **Right.** In 2D, a complex streamline can at most occupy all the grid squares, leading to box counting ratio 2. (b) Effect of grid size on box counting ratio. The same spiral is placed in four different grids with decreasing resolution from left to right. The box counting ratios by taking consecutive pairs from left to right are 1.28, 2 and 2 respectively.

Fig. 2: Box counting ratio of 3D streamlines. (a) A relatively straight streamline with box counting ratio 1. (b) A streamline with a complex feature and higher box counting ratio. **Inset.** Zoomed-in view of the complex segment.

Streamline predicates are proposed by Salzbrunn et al. [16] to quantify the relation between flow features and streamlines. Shi et al. [17] propose a suite of metrics for pathlines, allowing multi-variate query on the metrics.

After computing the box counting ratio and using it to locate features, we project them to a 2D space linked to the spatial domain. The concept of using linked feature space to interactively explore vector data was introduced by Henze [18]. Doleisch et al. [19] presents a similar approach of using multiple co-ordinated display for complex static and time-varying volume data and flow simulation data. A recent work [21] describes an interactive feature based framework for direct analysis of vector fields. This technique projects the vector directions at spatial grid points onto a high dimensional feature space and then allows visual exploration of interesting regions via brushing and linking on the feature space. Another recent paper [22] computes Hausdorff distance matrix from the streamlines and projects them on a 2D space using multi-dimensional scaling. Our solution is different from these methods because we use a different metric, and we do not apply transformation over all the streamlines. Rather we first reduce the data by identifying the segments of potential importance and then feature-space projection.

Since flow fields can have features of different sizes, multi-scale feature detection techniques are needed to analyze them. This is considered a crucial problem in image processing as well. A classic approach used in image processing is Shift-Invariant Feature Transform (SIFT), which detects the salient points within an image at multiple scales [23]. Our segmentation algorithm is based on this principle. SIFT-based techniques have been adopted to detect features in various types of scientific data. The size-based transfer function by Correa and Ma [24] is one example which allows the user to classify volumetric features according to their sizes. For flow data, Schlemmer et al. [25] employ moment invariants of the local neighborhoods to detect flow features of different sizes.

3 **FRACTAL ANALYSIS OF STREAMLINES**

3.1 **Definitions**

3.1.1 **The Box Counting Dimension**

Length, area and volume are the simplest measures of how much space an object occupies. However, since an object with fractal behavior replicates itself at different scales, its conventional length, area or volume keeps increasing as the scale of measurement is refined. It is observed that, for true fractals, the rate of growth of this measurement converges to a number, when the unit of measurement becomes infinitesimally small. This limiting value is known as fractal dimension. Formally speaking, if a fractal $F$ is measured at a scale of measurement $\delta$, the measured quantity, which is denoted by $N_\delta(F)$, and the unit of measurement are observed to have the following relation: $N_\delta(F) \approx \delta^{-D}$, where $D$ is a constant known as fractal dimension [26]. In other words, fractal di-
Mension is defined as:
\[
D = \lim_{\delta \to 0} \frac{\log(N_\delta(F))}{\log(1/\delta)} \tag{1}
\]

The box counting dimension is one way of defining a fractal dimension of a set of points. It provides a way to quantify the extent of space-filling by the fractal object. The box counting dimension of a set \( F \) is defined by counting the minimum number \( N_\delta(F) \) of boxes of edge length \( \delta \) which cover \( F \) and then taking the limit of \( -\log(N_\delta(F))/\delta \) as \( \delta \) goes to zero.

However, it is fundamentally different than another common measure of fractal dimension, the Hausdorff dimension of the set. The box counting dimension measures self-replicating, fractal behavior around a single point while the Hausdorff dimension is always one. As explained in [2], the box counting dimension measures self-replicating behavior on an integer Hausdorff and box counting dimensions. In contrast, spiral curves can have box counting dimension greater than one (see Section 3.3).

In Figure 1a, a straight line, or a series of line segments (Left figure) along the same direction, is the simplest possible form of a streamline. As we move to more complex streamlines spanning over a 2D rectilinear space, the box count grows more rapidly with decreasing grid size. This results in a box counting ratio somewhere between 1 and 2 (Middle figure). The most complex streamline in 2D intersects with every grid square and hence, has a box counting ratio 2 (Right figure).

An intuitive explanation of the definition can be found in Figure 1a. A straight line, or a series of line segments (Left figure) along the same direction, is the simplest possible form of a streamline. As we move to more complex streamlines spanning over a 2D rectilinear space, the box count grows more rapidly with decreasing grid size. This results in a box counting ratio somewhere between 1 and 2 (Middle figure). The most complex streamline in 2D intersects with every grid square and hence, has a box counting ratio 2 (Right figure).

3.1.2 Box counting ratio

For all but the simplest streamlines, it is impossible to compute the above mentioned limit of \( -\log(N_\delta(F))/\delta \) as \( \delta \) goes to zero. Moreover, physical objects, as opposed to mathematical ones, exhibit fractal behavior only in a range of \( \delta \) values. Thus the limit as \( \delta \) goes to zero does not truly exist.

Instead of computing this limit, it is practical to compute \( \log(N_\delta(F)) \) for a set of \( \delta \) within a given range, fit a line to the points \( (1/\delta, \log(N_\delta(F))) \), and then compute its slope. For mathematical objects, the slope of this line converges to the box counting dimension as \( \delta \) goes to zero. The box counting ratio used in this paper converges to the box counting dimension as \( \delta \) goes to zero. The box counting ratio computes \( N_\delta(F) \) for just two values, \( \delta_1 = \delta \) and \( \delta_2 = 2\delta \), and is based on the ratio of those two values. We call this quantity box counting ratio, denoted by \( B \), to be defined as below:

\[
B = \log\left(\frac{N_{\delta_1}(F)}{N_{\delta_2}(F)}\right) \tag{2}
\]

It is a single, well-defined quantity which depends only on a single value \( \delta \).

3.2 Box Counting Ratio of a Streamline

A streamline \( S \) is an sequence of line segments \( \{s_i, s_{i+1}\} : i \in [1, N] \), obtained by advecting a seed point(s) for a specified number of steps, or until the point hits the field boundary. Computation wise we are interested to know how many grid cubes of a fixed length intersect with the streamline. This number indicates how spread out or irregular the geometry of the streamline is at that scale. This value is first computed in a grid of cubes having length \( \delta \) and then in a grid of cubes with length \( \delta/2 \). The logarithmic rate of these two counts gives the box counting ratio of the streamline.

Two consecutive sample points on a streamline may be in many voxels apart based on the advection parameters. In that case, a suitable voxel traversal algorithm is required to count the number of voxels traversed between a pair of consecutive samples. We have employed Amanatides-Woo algorithm [27] for this purpose.

Another issue arises when a streamline intersects a grid cube more than once. In such cases, multiple intersections between a streamline and a grid cube are counted only once. Only by ignoring the duplicate counts we can obtain a box counting ratio compatible with the streamline’s complexity. Hence, a suitable data structure for efficiently storing the voxels which are already traversed is needed to avoid duplicate counts.

3.2.1 Choice of Box Size

As mentioned in Section 3.1.2, box counting ratio depends on one parameter, the length of the edge \( \delta \) of the box used for counting. This denotes the edge length of the finer of the two grid resolutions used. As \( \delta \) becomes smaller and smaller, box counting ratio of any space curve (which is not a true fractal) should eventually converge to 1. On the other hand, as \( \delta \) increases, the number of boxes in the grid reduces,
Fig. 3: Box counting ratio based range query on streamlines. (a) Distribution of the box counting ratio of 1000 streamlines from Solar Plume. (b) All streamlines with box counting ratio more than 1.6 from this dataset contain complex feature. (c) Streamlines with box counting ratio between 1.35 and 1.6 have complex segments as well. (d) A few streamlines with box counting ratio almost equal to 1.0.

Fig. 4: Effect of grid resolution pair on the computation of the box counting ratio. (a) Box counting ratio of 1000 streamlines computed with seven different grid resolution pairs. Finer to coarser resolutions are plotted from left to right. (b) Two clusters of streamlines which becomes prominent at different box sizes. (c) Two different features corresponding to two clusters.

allowing each box to occupy a higher fraction of the bounding volume of the curve. As a result, the value of the box counting ratio tends to increase in general. However, if box size becomes so large so that the entire curve fits in one box, then it is considered a point-like object at that scale and results in a box counting ratio of zero. So, at a very large box size until there, the streamlines tend to have high values, making distinction difficult. On the other hand, at a very small box size, they tend to have values close to 1. Any box size in between, which produces enough variation among the scores, is suitable.

Figure 1b shows the box count for a spiral at four different box sizes. If the box counting ratios of the spiral are computed using consecutive grid resolution pairs from left to right, the values turn out to be 1.28, 2 and 2 respectively. Figure 4a presents the summary of computing the box counting ratio of 1000 streamlines from Solar Plume dataset at 7 different grid resolution pairs. Clearly, the shrinking height of the blue rectangle (which represents the inter-quartile range) towards the left shows convergence of values to 1 at smaller box size. Also, the upward trend of the blue box and the median from left to right indicates the values generally go up with box size.

Change of the box counting ratio with change of grid resolution pair can be exploited to identify features of different sizes. We observe that depending on size, different features (a streamline or a complex part of it) manifest most prominently at different grid resolution pairs. Figure 4b shows two clusters of streamlines whose values respond differently to varying box sizes and Figure 4c shows the actual features corresponding to them (blue curve corresponds to the top vortex).

3.2.2 Influence of Rotation and Translation

In this section, we examine the robustness of our metric in terms of translation or rotation of the segments.
or the underlying grid. The value of the metric should not differ too much for two similar-shaped streamlines seeded from two very close points. Intuitively, streamlines covering fewer boxes should be more susceptible to change under translation or rotation. A simple test is performed to verify the intuition where a set of streamline segments of different lengths are perturbed several times. The range of the variation of their box counting ratio and the average box counts are recorded. For each segment, Figure 6 plots the range of variation of the box counting ratio against the average box count. The plots support out intuition showing the maximum range of variation to be 1.5, which is quite large given 0-3 range of the metric, for short segments.

To deal with the issue, our implementation discards the segments with too small box count. Not only that, to make the computation of the box counting ratio robust under rotation of the grid, we apply a linear transform, called Hotelling transform [28], to the rest to maximize the box count. Hotelling transform, a popular technique in image processing, is based on the idea that a straight line aligned to one of the axes occupies more boxes than its counterpart which has the same length, but is aligned to the diagonal. Hence, the transform rotates an object such that its principal components match the axes as much as possible. Besides this, one more step is taken to nullify the effect of translation. For any streamline or segment, we first shift it such that its bounding box coincides with the origin of the grid (e.g. grids in Figure 1). We do not use a global grid for all. These two transforms add to the computation time of the metric which will be discussed in detail in Section 5.2.

3.3 Comparison with Other Metrics

It may appear that the box counting ratio measures the curvature of a streamline in a different way. However, a closer look reveals that the two metrics capture different properties. Previous research [7] shows that they can even be used as complementary metrics. The authors of that paper used signed curvature, the scalar product of the normals at two consecutive locations of the streamline, averaged over a certain number of steps. This is clearly different from box counting dimension. For instance, a meandering streamline should have very low value of integrated signed curvature, whereas its box counting ratio can still be quite high depending on how compact it is. Hence, a low value of integrated signed curvature does not necessarily imply low box counting ratio. The contrary is not true either. A high value of integrated signed curvature does not necessarily imply high box counting ratio. Using their metric, the authors of the above mentioned paper first filtered out streamlines with “persistently strong curvature”, primarily to eliminate straight and meandering streamlines and keep the spiral and helical ones. As a next step, they used the box counting dimension or Kolmogorov capacity to achieve a finer level of refinement. It was observed that the streamlines with persistently strong curvature span over a wide range of the box counting dimension.

We perform a similar study using unsigned curvature, computed at each point of a streamline as the derivative of the tangents at two consecutive locations. To compute integrated curvature of a segment, we normalize these values over its length. The following tests on synthesized and real data show that the two metrics serve two different purposes. **Synthesized data:** Four 2D spiral streamlines are generated using the following equation: $r = 1/(ap)$, where $a$ denotes the angular deviation from x-axis and $p$ controls the rate of winding (Figure 7a). A low value of $p$ is used for all of them, so that they wind towards the center at a slow rate forming a dense spiral. The first two are made to stop after just 1 and 4 circles, the last two roll till 8 and 16 respectively. We obtain the following integrated curvature values for the streamlines: 0.018, 0.0176, 0.0175 and 0.0175. Knowing the bounding box of the spiral, we can verify that these values approximately match with $1/R$ formulation of curvature. While the curvature remains more or less same, their box counting ratios are 1.00, 1.28, 1.49 and 1.59 respectively for the streamlines with 1, 4, 8 and 16 coils. **Real data:** For the Solar Plume dataset, we compute both box counting ratio and integrated unsigned curvature for 1000 streamlines and then apply a threshold on curvature to select only top 25% streamlines. The histogram of the box counting ratio of only these streamlines (Figure 7b) shows that majority of them has box counting ratio close to 1, which is on the lower side. Figure 7c displays the streamline with maximum average curvature. Since it is contained within a fairly large bounding box of approximate dimension $64 \times 46 \times 78$, its low space-filling capacity is reflected by its overall low box counting ratio (1.03).

Besides curvature, another important property of a space curve is torsion, which measures how fast a curve is twisting in a 3D space. It is easy to distinguish the nature and purpose of torsion from the box counting ratio. Torsion is meaningful only for a specific type of streamlines, namely, helices or spirals advancing in a 3D space. It cannot distinguish between plane spirals, because a planar curve of non-vanishing curvature has zero torsion everywhere. On the other hand, the box counting ratio is able to measure the complexity of a planar spiral (in 0-2 range) as well as a non-planar helical structure (in 0-3 range). However, our metric is not limited to any particular type of geometry and serves a much wider spectrum of features.

A number of techniques rely on variations and extensions of a metric called winding angle [13], [14] for vortex core detection. We do not provide explicit
comparison with this metric for the aforementioned reason that our objective is not to automatically detect a particular type of feature. We rather aim at presenting all the potentially complex regions, which includes vortices, to the user in an organized manner.

Another metric used to measure the tendency of rotation of a flow field is vorticity or curl. Since it is a property of the field, accurate computation of curl requires the local velocity. Without that, vorticity of streamlines can still be computed less accurately by approximating the values from the available sample points on the streamlines. Box counting ratio, on the other hand, can be computed without any knowledge of the underlying vector field and hence, can be more useful in the large data scenario, where the field may be too large to store locally or access repeatedly during metric computation.

### 3.4 Distribution of Box Counting Ratio of Streamlines

A histogram generated from a set of streamlines and corresponding box counting ratios provides an overview of the range of complexity of the streamlines. It informs the user about the presence of complex region(s), encouraging further analysis and query for precise location of the complex regions. Figure 3a presents a histogram with 100 bins obtained from 1000 streamlines generated randomly in the vector field Solar Plume. As expected, most of the streamlines have box counting ratio close to 1. Only a few are found above 1.2. Based on the histogram, range queries can be performed to single out potentially interesting streamlines. Figure 3b shows 8 streamlines corresponding to the rightmost part of the histogram. We note that all these streamlines, having box counting ratio 1.6 or more, indeed capture a complex region of the dataset. The 14 streamlines in Figure 3c correspond to another range (1.35-1.6) on the higher side. They also contain vortex-like segments. To avoid clutter, we present a few streamlines with value close to 1 (Figure 3d). They are seen to be hardly capturing any complex pattern. We will show later that computing the box counting ratio on segments of streamlines produces better result, since streamlines often comprise segments of varying complexity.

### 3.5 Box Counting Ratio of Streamline Segments

If geometric complexity varies widely along a streamline, box counting ratio computed over the entire streamline may not capture the local variations. To see the local variation, we slide a segment of fixed arc length along a streamline and compute the metric for each segment (considering it to be a shorter standalone streamline) on-the-fly (Figure 5). The particular streamline in the example has a small complex segment (painted in red). The box counting ratios of the sliding segments for different segment lengths (bottom two rows) underline the importance of selecting the right segment length (in terms of number of advection steps or arc length along the streamline). In general, if the segment length is too small compared to the size of a complex part, the segments may never capture the whole of it. Also, any sharp change in the streamline direction may result in a high score, producing a noisy plot with multiple local maxima. On the other hand, if the segment length is too large compared to a feature, the complex segment is never isolated, resulting in a smoothed out plot. Section 4.1 will discuss these issues and proposed solutions at length.

### 4 Feature Based Representation of Streamlines

The objective of our work is to identify, organize and present the complex regions from a large number of streamlines using the box counting ratio. To achieve this goal, we propose a novel visual analytic framework which extracts the potentially interesting segments from individual streamlines and places them on an interactive 2D visual space. The user is able to easily select points or regions from this space and visualize corresponding features on a linked display.

The sequence of steps that leads to such meaningful feature based representation of streamlines is presented in Figure 8. Given a collection or database of streamlines, we first locate the features on each
of them using the box count ratio, and then create a feature-preserving segmentation. Following steps involve construction of feature vectors from a selected subset of segments and finally, projection of the features to a 2D space.

It is worth mentioning that this framework can potentially work with any suitable metric or a combination of metrics. The choice of metric depends on the goal of the analysis. Based on the chosen metric, streamlines will be segmented in different ways and different types of segments will be selected as features. In this paper, we will continue to discuss the individual steps of the framework using the proposed metric.

### 4.1 Feature localization using box counting ratio

To locate the geometrically complex region(s) along a streamline, separate analysis of its different parts is required due to the variation of complexity along its trajectory. Since the specific size of an interesting segment is not known in advance, it is important to study the parts at different length scales as well. However, it is not computationally tractable to create streamline segments of all possible lengths from all possible points along its trajectory. Sliding a fixed length segment along the streamline is not practical either, since the computation has to be repeated for a wide range of length.

In this paper, we propose a hierarchical 2-way subdivision scheme for this where a streamline is recursively partitioned into two smaller halves until a segment of equal or smaller than a minimum length threshold is reached. The box counting ratio of each segment formed in the subdivision process is computed. Hence, we obtain a hierarchy of segments (Figure 9a) along with the scores, with the entire streamline at the root of the hierarchy and the segments of different lengths at the appropriate levels.

We refer to this hierarchy of metric values as feature map of the streamline, due to its ability to indicate presence of features. It can be visualized using a compact space-filling layout (Figure 9a). The horizontal span of the entire rectangle from left to right denotes the streamline’s start to end point. Segments of same level are placed in the same horizontal stretch from left to right based on their start points. The bottom level denotes the entire streamline. The box counting ratios are appropriately color coded so that the segments with high values stand out.

Besides its utility as a standalone visualization, the feature map serves as the prerequisite of feature-aware segmentation of a streamline in our framework.

### 4.2 Feature Map Based Segmentation

The segmentation algorithm analyzes the feature map, or possibly a set of feature maps from different grid resolution pairs used to compute box counting ratio, of each streamline. It generates an array of segments along the trajectory of each streamline. This algorithm is required because a complex part of a streamline may be captured by multiple overlapping segments at different levels of the hierarchical feature map.

Formally, a segment is defined by the following attributes:

1) **Level:** Level in the hierarchy. Level 0 denotes the entire streamline and increasingly higher levels represent shorter segment. Any two consecutive segments in the same level have only one point in common.

2) **Start:** The ID of the first sample point along the streamline to which the segment belongs. It is equivalent to storing the actual spatial location of the starting point.

3) **Length:** The number of sample points along the segment. The actual arc length of the streamline can be computed from start point and length.
Fig. 10: (a) Demonstration of the segmentation of a streamline based on its feature map computed with one or multiple grid resolution pairs. (b) A streamline with multiple complex segments detected using the box counting ratio. (c) Feature maps computed at different grid resolution pairs.

4) **Box counting ratios**: Box counting ratio for a given grid resolution pair.

Input to the segmentation algorithm is the feature map of a streamline. The proposed algorithm first extracts the list of segments from the highest level of the feature map. The array of blank rectangles in Figure 10a stands for the extracted top level segments. The extracted segments are ordered according to their start points. For each extracted segment, the box counting ratios of all other segments from different levels which have an overlap with it are examined. The maximum of all of these scores is taken as the final value of this segment.

As shown in Figure 10a, the algorithm can also process multiple feature maps computed using different grid resolution pairs. For a given streamline, the recursive subdivision produces same set of segments, regardless of the grid resolution pair. So, the different feature maps associated to a streamline contain the same set of segments. When more than one feature maps are available, the algorithm scans through the overlapping segments across all of them to obtain the maximum possible score.

Even though the above step assigns a score to each extracted segment (denoted by the colored array right below the blank one in Figure 10a), these scores are gathered from different levels and possibly from different resolution pairs of the feature map. Hence, these scores are re-computed. Since computation of the metric on a short segment is more susceptible to noisy values, consecutive segments with same or very close scores are first concatenated to form larger segments (bottommost colored array in Figure 10a).

Even though the above step assigns a score to each extracted segment (denoted by the colored array right below the blank one in Figure 10a), these scores are gathered from different levels and possibly from different resolution pairs of the feature map. Hence, these scores are re-computed. Since computation of the metric on a short segment is more susceptible to noisy values, consecutive segments with same or very close scores are first concatenated to form larger segments (bottommost colored array in Figure 10a). (Threshold of closeness can be implementation specific. Since the box counting ratio has a fixed range (0-3), a stable criterion can be set based on some percentage on this range.) Then, the box counting ratios of the concatenated segments are re-computed. Again in this case, the computation can be repeated across all grid resolution pairs, when more than one is being used.

The example in Figure 10 highlights the benefit of calculating the box counting ratio using more than one grid resolution pairs during segmentation. The streamline in Figure 10b has three geometrically complex segments of different sizes. Figure 10c presents the feature maps for this streamline using different resolutions. The feature map at the top left corner uses the finest pair of resolutions and hence, looks smoothed out. The other three feature maps can highlight at least one of the complex segments.

### 4.3 Construction of Feature Vectors

After each streamline is segmented, the high scoring segments should now represent the potential features and hence, they are selected using some reasonable threshold. Due to the standard interpretation available for the range of the box counting ratio, any value above 1 works as a threshold. However, the right threshold depends on the abundance and the value range of the complex segments in the data and to some extent, on the analysis goal. A histogram of the box counting ratios of the segments can guide the user to a quick but informed selection of threshold.

The next step in the framework is to represent the selected segments as feature vectors. A representative feature vector should have a spatial component to ensure that spatially close segments having similar box counting ratio also stay in close proximity on the feature space. However, box counting ratio alone does not contain information about the spatial location and size of a segment. The location information related to a segment can be captured through various quantities, for example bounding box, center of bounding box, etc. Similarly, size of a segment is related to quantities like arc length, cross-diagonal length of its bounding box etc. For our experiments, we have selected the following as components of the feature vector:

- **Center of bounding box**: Center of the bounding box of the discrete sample points constructing the
segment. This encapsulates information about the location of the segment.
- **Diagonal span of bounding box**: Length of the cross-diagonal of the bounding box contains information about the size of the segment.
- **Box counting ratio**: Finally, the box counting ratio of the segment provides information about how much of the bounding box is actually occupied by the segment.

In essence, the feature vector constructed this way describes a segment by its location, shape and space-filling capacity. From now on, we will use the term *feature segment* for an entity in the feature space.

### 4.4 Projection and Visualization of Features

We have employed suitable multi-variate data visualization techniques and proposed novel interactions on top of it to extract the information encapsulated in the high dimensional feature vectors.

![Fig. 11: 2D visual space showing relationship between feature size and box counting ratio in Solar Plume data set.](image)

(a) Feature size increases from left to right. The selected feature has relatively high size and forms the outer region of a vortex. (b) The link guides to another segment which has smaller size, higher box counting ratio and comes from the inner region of the same vortex.

#### 4.4.1 Projection onto a Subspace

One way to visualize the feature vectors is to select a pair of dimensions, which form a subspace of the feature space, and to plot them on 2D. This method works when the analysis goal is well-defined. As an example, Figure 11 plots the box counting ratio against feature size (measured by the diagonal length of the bounding box of the feature segment). Each point on the 2D space denotes a feature segment, color coded by its size (using the same color map as the one in Figure 9b). The user is allowed to select one or more points in the feature space (blue rectangle encloses selected points) and see the actual segment(s) in a linked display. To assist the user in gaining further insight, we have enhanced the 2D space by adding a link between the feature segments that come from the same streamline. This apparently simple interaction can help in different ways. For example, the user may want to see the different segments coming from one streamline, while selecting or discarding streamlines based on their importance. The links give an idea about a streamline’s contribution to the feature space. In other words, this reveals which streamlines are more important either in terms of having many features (equivalent to having high number of links), or in terms of containing features of varying sizes (expressed by links going across the horizontal axis for this particular plot).

In the particular example shown in Figure 11, the selected segment belongs to a streamline which has contributed to the feature space with a relatively large segment (Figure 11a), which covers the outer zone of a vortex, a smaller segment (Figure 11b), which covers the central zone of the same vortex and another segment (not shown). This indicates that this streamline is a good choice for being selected as a representative of this vortex. The actual segments (shown in insets) are colored according to their box counting ratios. The deeper color of the inner vortex indicates that it has higher box counting ratio than the outer part, as expected.

#### 4.4.2 Projection using dimensionality reduction

To understand the relationship among the feature segments in the high dimensional space in a more general way, we employ Principal Component Analysis (PCA), a widely used dimensionality reduction technique, to transform the features and project them on to the two dimensions with maximum variance. Any other dimensionality reduction technique which can preserve the proximity among high dimensional feature segments in the resulting 2D space, such as multi-dimensional scaling, can be used as well with suitable changes in the feature vectors. Since different feature dimensions may have varied range and scale, each dimension of the feature vector is divided by its standard deviation before employing PCA for all the results shown below (unless otherwise specified).

Figure 12 presents the results of PCA on the feature segments created from the Solar Plume dataset. The left column displays the segments selected by the user as points in the 2D space, the right column shows the actual segments. Since all the segments in the feature space have high box counting ratio, the box counting ratio has less variation which makes it a less ideal candidate for color coding. We color the features on the 2D space based on feature size, the other available feature dimension. The segments, to be displayed on demand in the actual 3D space, are colored according to their box counting ratio. A set of relatively large features has been selected in A (Figure 12a) and they are found to be spatially clustered as well. Selection B contains relatively smaller features, which turn out to have very high box counting ratios. Figure 12d demonstrates a different mode of display, where the entire streamline(s) containing...
the selected feature(s) is shown. The actual segment that has caused a streamline to be visible is marked in red (inset). Figure 12d indicates that in this case, the corresponding streamlines behave similarly in the locality of the closely located features. However, as the streamlines move far away from the clustered segments, they tend to follow different paths.

The 2D space may contain cluster(s) depending on the nature of the data and the density of streamlines. But isolated points are also important since they usually represent features which are not captured by many streamlines due to small size and hence, are easy to miss or to get occluded in a standard 3D visualization. Also, depending on the goal, the user may choose to visualize the selected feature segments in different ways, such as segments, or short streamlines containing the segments, or entire streamlines. For instance, if the goal is to select a subset of streamlines to create an uncluttered visualization, a representative streamline can be picked from each cluster, discarding the rest.

5 Results and Analysis

5.1 Case studies

In this section, we demonstrate with a few examples how our framework helps locate and visualize important feature segments as opposed to static visualization comprising many streamlines. To start with, we study the Solar Plume dataset, a $126 \times 126 \times 512$ vector field with many vortices and tortuous regions. But a static visualization with only 200 streamlines (Figure 13 left) fails to convey useful information about the features. Change of viewpoint is not helpful either, especially for dense streamlines. Sparse streamlines can reduce occlusion, but may skip features. On the other hand, our technique is able to detect and display a spectrum of the feature segments, as shown in the top row of Figure 14. It displays all the segments with high box counting ratio (threshold used is 1.8 in this case) in spatial domain. The user can select one or a few of them at a time.

**Visual link between segments:** The visual links between segments help extract additional knowledge from the PCA outputs. For example, the images at the bottom row of Figure 14 indicate that the three selected regions correspond to the three different vortices. Now, the user may follow the links that move out of the box and select the other end of the links. If the other end corresponds to a different vortex, then a flow connection between these two vortices
is revealed. Such flow connections between remote features are often hidden in a cluttered streamlines-based visualization. However, in this particular case, the other ends of the links from the three boxes correspond to the respective vortices, suggesting that these vortices are not connected to any other one.

In addition to Plume, we explore two more datasets with our technique. The first one, called Ocean, is a $3600 \times 2400 \times 40$ vector field (Figure 13 middle) which is the result of an eddy resolving simulation. The vector field contains numerous vortices and complex patterns. So, dense streamline generation is needed to capture all of them. However, visualizing all the features together is not useful due to clutter. Moreover, although it is almost 2D, its large x-y span makes it difficult to interactively explore the small-scale local features from the global view.

We apply our visual analytic framework in two different ways to obtain two different distributions of feature segments, important for separate reasons. In the first case (Figure 15), we apply PCA without prior standardization of the dimensions. The motivation is to retain the spatial distribution of features as much as possible. Since the streamlines for Ocean are embedded in wide 2D planes and the z-span is comparatively small, skipping the standardization makes x and y the two dominant feature dimensions. Hence, the distribution of points on the 2D space resulting from PCA closely preserves the original spatial distribution of the streamline segments. As can be seen in Figure 15, the top region of the PCA output corresponds to streamlines from the upper part of the spatial domain. In other words, horizontal partitioning of the PCA space corresponds to a similar partitioning of the streamlines in the spatial domain. The connectivity pattern of the segments is also interesting, because a dominance of near-horizontal links can be noted in all three regions. Given the already noticed correspondence between the PCA space and the spatial domain, this suggests that, for this dataset, flow features along horizontal directions are more likely to be linked by one or more long streamlines. This also implies that there are hardly any flow feature going from the top region to the bottom region or vice versa for this dataset.

In the second case shown in Figure 16, we employ PCA after standardization of the dimensions and in this case, the feature size appears to be one prominent dimension. The feature segments on 2D space are color coded based on size, so it is apparent that features are organized according to their sizes (diminishing from left to right). Exploring individual links on this space reveals interesting flow patterns. For instance, tracking a long link (Figure 16a), which stands out in the 2D space, we can find two different sized features (Figure 16b) connected through a convoluted streamline. The link shown in Figure 16c reveals connection between two far away vortices (Figure 16d). Without the help of the links, it would have been cumbersome to identify such connections in a standard visualization of the data.

We also experiment with a $512^3$ vector field called Nek, a thermal hydraulics simulation. Due to the nature of the field, any visualization of it (Figure 13 far right) would suffer heavily from occlusion. Our framework helps to unveil many interesting features of different sizes hidden inside this vector field. Figure 17 presents two examples, where the points selected on 2D space correspond to complex features of the field. We also observe that it has relatively small number of features and unlike Plume and Ocean, the feature space hardly contains any long link.
5.2 Performance

We describe two algorithms in this paper: computation of box counting ratio (Section 3.2) and feature map based segmentation of streamlines (Section 4.1 and 4.2). Table 1 provides total computation times of these algorithms on different datasets. All tests are performed on an Intel(R) Core(TM) i5-2410M CPU 2.3GHz processor, 6GHz machine.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Avg. streamline length</th>
<th>Computation w/o HT</th>
<th>with HT</th>
<th>Segmentation w/o HT</th>
<th>with HT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plume</td>
<td>2113</td>
<td>16</td>
<td>28</td>
<td>91</td>
<td>188</td>
</tr>
<tr>
<td>Ocean</td>
<td>3419</td>
<td>27</td>
<td>46</td>
<td>173</td>
<td>337</td>
</tr>
<tr>
<td>Nek</td>
<td>6197</td>
<td>51</td>
<td>86</td>
<td>360</td>
<td>656</td>
</tr>
</tbody>
</table>

TABLE 1: Performance. Timings for computation of box counting ratio and feature based segmentation of 1000 streamlines from different datasets with and without Hotelling Transform (HT).

Computation of the box counting ratio of the entire streamline depends on two factors: (a) the selected pair of resolutions of the grid and (b) the number of discrete points along the streamline. For all the tests shown in the table, a grid pair of respective cube lengths 1 and 2 is used. Each dataset contains 1000 streamlines generated from 1000 seeds randomly placed in the vector field. The streamlines are truncated when they hit the boundary of the domain. This is why the average number of sample points per streamline (column 2) is given to provide an estimate of the size of the data. The timing for segmentation algorithm includes computation of feature map (Section 4.1). The segmentation is basically an iteration of box counting ratio computation. The number of iteration depends on the length of the streamline as well as the threshold of minimum segment length. For all the results shown in the table, the chosen threshold is 100. The reported timing is based on segmentation using one resolution pair. All the streamlines used have equi-spaced sample points (fixed arc length between any two consecutive sample points) so that the number of sample points remains a reasonable estimate of the actual arc length of the streamline.

Section 3.2.2 describes a pair of linear transforms which can be optionally employed to make the metric computation translation and rotation invariant. We provide timings for computation with as well as without these transforms (columns 3 and 4). Since the segmentation algorithm iteratively computes the metric, it is also affected by these transforms (columns 5 and 6). As the computational overhead is not negligible, a GPU-based implementation is in our plan. We also plan to improve on our current CPU-based implementation.

6 LIMITATIONS

The proposed feature exploration framework analyzes flow fields based on the streamlines. It does not directly process the field because the sheer size of the field is often not suitable for interactive analysis. Hence, even though our method is not biased to or dependent on any particular streamline placement strategy, it is favorable to the situation where the flow field has been sampled with a large number of streamlines and all the so, all the flow features have
been captured by at least one of them. While excessive streamlines do not pose a problem to our technique in terms of feature detection, sparsely spaced streamlines are not well-suited for our method.

Our framework has been applied to steady flow so far. Extending it to unsteady flow may not be straightforward. Since the time-varying primitives such as pathline have an implicit temporal dimension, it requires further study to understand the meaning of fractal dimension on them. Also, the velocity magnitude is crucial for unsteady flow. It can be missed if boxes are counted only in the spatial domain.

Selection of representative streamlines from the visual space is a fairly manual process. It can occasionally become tedious based on the density of points. Even though our main focus in this paper is not on the user interface, enhanced supports for the user are needed in the long run such as automatic streamline selection from a selected cluster.

7 Conclusion and Future Work

In this paper, we propose a new method for analyzing vector fields. We introduce a metric called box counting ratio which quantifies the geometric complexity of streamlines. We use this to extract potentially interesting segments from the streamlines and present them on a navigable and interactive 2D space linked to the spatial domain. This streamline based approach is relevant in the context of extreme scale data where direct interaction with the field is practically impossible.

We intend to move forward with our idea along a couple of new directions. We plan to identify other geometry-based metrics, if any, which can be combined with box counting ratio to achieve more meaningful feature space organization of streamlines.

We are trying to extend the idea to time-varying flow fields. We are also interested to study the fractal behavior of other geometric objects widely used to analyze flow field such as stream surfaces.

References


