A Distribution-Based Approach to Tracking Points in Velocity Vector Fields

Liefei Xu, H. Quynh Dinh
Dept. of Computer Science
Stevens Inst. of Technology
lxu1,quynh@stevens.edu

Eugene Zhang, Zhongzang Lin
School of Electrical Engineering
and Computer Science
Oregon State University
zhange,lin@eecs.oregonstate.edu

Robert S. Laramee
Dept. of Computer Science
Swansea University
R.S.Laramee@swansea.ac.uk

Abstract

We address the problem of tracking points in dense vector fields. Such vector fields may come from computational fluid dynamics simulations, environmental monitoring sensors, or dense point tracking of video data. To track points in vector fields, we capture the distribution of higher-order properties (e.g., properties derived from the gradient of the velocity vector field) in a novel local descriptor called a vector spin-image. Our distribution-based approach has a number of advantages over methods that use topology analysis to track points in vector fields. The local distributions are robust to noise, adaptable to changes in the feature, and can be used to extrapolate the location of features after they have disappeared. We describe the vector spin-image data structure, the higher-order properties we record to track vector field points, and show results of tracking points in the simulated flow through a diesel engine cylinder.

1. Introduction

Vector field analysis has become increasingly crucial as computational methods for simulating fluid dynamics, sensor technology for environmental monitoring, and video surveillance increase in accuracy and ubiquity. The dynamic information captured by these technologies are often in the form of dense vector fields (*e.g.*, optical flow of video data or wind and water velocity from environmental monitors). A fundamental task in flow analysis is the ability to extract features and track their movements in the flow. Example scenarios include an aircraft engineer who studies the formation and evolution of vortices to achieve more effective control, an engine designer who examines the mixing of fuel and air to obtain better fuel efficiency, and an environmental scientist who evaluates the environmental hazard due to pollutant leak near the shoreline. While the definition of features is highly application-dependent, it is common to track a single point through the flow field. Examples

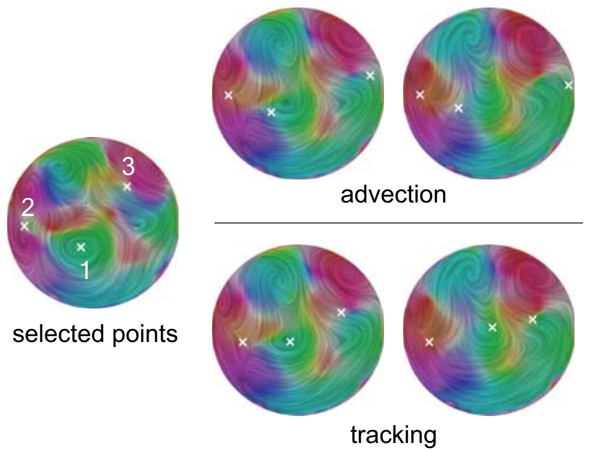


Figure 1. Left: selected points (X) in the 1st slice of a 3D vector field. Right: advection and tracking of these points in later slices.

of flow field feature points include singularities (*i.e.*, fixed points with zero-valued vectors), vortex cores, and distinctive regular points (*e.g.*, points on a ridge).

We address the specific problem of tracking points within 2D planar vector fields. Given a time-varying vector field $V(\mathbf{x}, t)$ ($\mathbf{x}$ and t represent the position and time respectively), two time instances $t_1 < t_2$ and a point $\mathbf{x}_1$, we find a point $\mathbf{x}_2$ such that $V(\mathbf{x}_1, t_1)$ and $V(\mathbf{x}_2, t_2)$ have matching properties. The examples in this paper are steady-state 3D flows that do not vary in time. In this case, t becomes the third spatial dimension rather than time. We treat the 3D flow as a sequence of 2D slices and track points from one slice to the next.

Note that the problem of tracking points is *different* from the *pathline computation* (i.e., advecting a particle along the flow). Particle systems for visualizing flow fields use advection. Particles are placed at seed points in the vector field and moved forward. The movement of the particles trace out pathlines, enabling the vector field to be visualized. In contrast, tracking points for analyzing the evolution of turbulent flow is achieved by finding the corresponding point in the next time or spatial slice. For example, if a point near a center is selected, advecting the point forward in the direction given by the vector field moves it towards the center. Tracking the point tells us how it moves relative to the movement of the center and can indicate how the larger feature (vortex) in which it is embedded evolves. Figure 1 illustrates the difference between advection and tracking for three selected points. When a selected point in a center (1) is advected (top row), it moves away from the vortex core. To adhere to the core, it must be tracked (bottom row). Figure 1 also shows that we can use tracking to extrapolate the position of a fixed point (in this case, a center) immediately after it disappears. Tracking backwards from a fixed point may reveal where turbulence begins and help engineers modify their design to reduce it.

Standard techniques for tracking points in vector fields use topology analysis to locate singular points and track them as they undergo *bifurcation* (topology change) [5, 12, 13, 17, 18]. Unfortunately, topology analysis is sensitive to noise in the vector field, is not easily scalable, and cannot be used to track regular points or non-singular features (such as ridges). Topological features such as separatrix surfaces, periodic surfaces and their connectivity which are needed in tracking the evolution of fixed points are difficult to assess in 3D. A solution often used in practice is to study the projection of a 3D flow onto a 2D submanifold – for example, treating the 3D flow as a sequence of 2D slices.

In this paper, we also consider the 3D flow as a sequence of 2D slices. However, our approach is completely scalable to 3D time-varying flows because it does not require topology analysis. Instead, we capture the geometric statistics of the vector field around each point in a distribution, and track a selected point through the volume by looking for the same statistical signature when going from one slice to the next. Because our approach is statistical in nature, it is robust to noise and can adapt as the selected feature changes.

Our contributions are as follows: (1) our algorithm is the first to track points through dense vector fields using geometric distributions instead of topology information; (2) because we do not require topology information, we are able to track regular points and non-singular feature points such as points on ridges; (3) we are able to extrapolate the position of fixed points after they have disappeared; and (4) we record in a local descriptor higher-order information such as those derived from the gradient of the velocity vector field to find corresponding points in tracking.

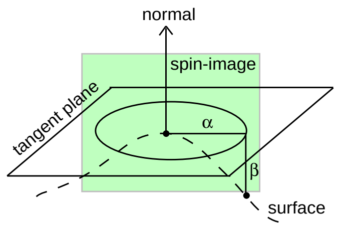
We now review related work in vector field topology analysis and tracking and geometric distributions for shape matching. In Section 3, we describe a local descriptor for vector field points, discuss zeroth and first-order properties of vector fields in Section 4 and how to apply them to tracking in Section 5. We show results of tracking points in the simulated flow through a diesel engine cylinder in Section 6.

2. Related Work

Vector field analysis identifies fixed points (with a zero vector value), separatrices (trajectories emanating from a saddle), and periodic orbits. Fixed points are further classified into sinks, sources, saddles, focus, and center points using the Jacobian matrix. A significant amount of research has been published in the area of vector field analysis and fixed point classification, including [3, 6, 14, 16, 17, 21]. Tricoche *et al.* track topology changes in time-varying 2D and 3D vector fields by locating bifurcations that transforms sinks to sources (and visa versa) and annihilates sink-saddle and source-saddle pairs [5, 17, 18]. The result is a topology skeleton spanning space and time that allows for changes in the number and type of fixed points. Reinders *et al.* track fixed points by finding corresponding features in later time slices based on the position, volume, and orientation of the fixed points [12, 13]. They also track fixed points through topology changes. An overview of different approaches to flow visualization and topology-based feature tracking is provided by Post *et al.* [11]. We focus on tracking points without extracting topology information. Given a selected point in a vector field, we track it to the next vector field by finding the most similar point in the next field. As we will describe in the next section, we identify similar points based on a local descriptor.

Previous work in comparing vector field points has focused on defining a metric to compute the distance between fixed points using alternate phase plane coordinate systems [9, 15]. However, many naturally occuring vector fields do not contain fixed points, and non-singular points can exhibit interesting features. Instead of relying on fixed points, we compute a local descriptor that stores local statistics on the vectors around each point in the form of discrete distributions, or histograms.

Geometric, or shape, distributions provide a way of comparing shapes based on statistical properties [1, 10]. These approaches record the distribution of a selected feature in a histogram for efficient storage, indexing, and comparison. A single global histogram may be generated for the entire shape, or one local histogram for each surface point. Local histograms store information about the neighborhood surrounding a central point and are used to find point correspondences. Points are compared by computing a difference, or norm, between their local histograms. Standard norms include the Minkowski L_N norms, the χ^2, Battacharyya, and Earth Mover's distance, and the correlation coefficient. Geometric distributions are invariant to rotations and translations of the underlying domain (*e.g.*, surface) and robust to noise due to their statistical nature. In shape and contour matching, successful approaches to generating local distributions include spin-images [7] and shape contexts [2, 4]. We now describe how we record the geometric properties of vector fields in a spin-image data struc-



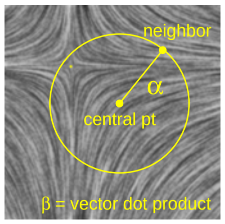


Figure 2. Left: Spin-image for surface points in [7]. Right: Our new vector spin-image for planar domains bins neighboring points based on distance α from the central point and dot-product β between the neighbor's vector and the central point's vector. All points on the yellow circle have a common α but β may differ.

ture for use as a local descriptor for tracking points.

3. A Local Descriptor for Vector Fields

In [19], we introduced a local descriptor for vector fields based on the spin-image data structure. We call it a *vector spin-image*. In Figure 2, we compare the spin-image for surface data (left) to the vector spin-image (right). Spin-images for an oriented surface point are computed by spinning the plane containing the normal vector about the normal axis while binning all surface points as they intersect the plane (making it invariant to rotations) [7]. The spin-image is indexed by the distance α from the central point and the depth β from the central point's tangent plane.

For planar vector fields, the indices for the vector spin-image are the distance α from the central point and the dot-product β between the 2D vector at the central point and that of the neighbor. We then bin (tally) the number of neighbors (within a given radius around the central point) based on their α and β values, resulting in a 2D histogram. Because areas far from the central point are deemed less important in a local descriptor, we use a log scale for radius α to achieve lower resolution in these areas as prescribed in shape contexts [4]:

$$r_i = \exp\left\{\ln(r_{min}) + \frac{i}{I}\ln\left(\frac{r_{max}}{r_{min}}\right)\right\} \tag{1}$$

In the above equation, r_i is the max radius of bin i; r_{min} is the max radius of the bin containing the central point; r_{max} is the maximum support of the spin-image; and i goes from 0 to I where $I+1$ is the total number of radial bins. Bin i covers the radial range from r_{i-1} to r_i. Bins far from the center will have high counts, but because each bin contributes equally to the difference (Equation 4), a point in one such bin actually has less effect on the difference.

4. Vector Field Properties for Tracking Points

As described in the previous section, our initial vector spin-image is two-dimensional and records for each neighboring point the distance from the center point and relative orientation with respect to the vector at the center point via the dot-product. Unfortunately, vector orientation alone often cannot capture essential differences between vector fields such as the strength of the flow feature of a vortex or fixed point. Given two sinks of different strengths, our initial vector spin-image would not be able to distinguish between them. By storing vector magnitude in addition to orientation, we can discriminate between points in the two fields. But because both vector orientation and magnitude are zeroth-order properties, they fail for tracking fixed points (see Figure 3). This is due to the fact that the vector field vanishes at fixed points, and the dot-product between a zero-valued vector and any other vector will be the same regardless of the type of fixed point (*i.e.*, sink, source, saddle, etc.). We now describe first-order properties based on the *velocity gradient tensor* that enable us to overcome this limitation and distinguish between different fixed points.

4.1. First-Order Information

The gradient of a velocity vector field is an asymmetric tensor field. In [20], Zhang *et al.* describe the structures in the eigenvalue and eigenvector fields of the gradient tensor (extending the theoretical results of Zheng and Pang [22]). They then relate tensor analysis to physical quantities such as rotation, angular deformation, and dilation. To more precisely track points through a vector field, we record these first-order properties in the vector spin-image. We now review the properties derived by Zhang *et al.*

The gradient of a 2D planar vector field, $V(x,y) = (F(x,y), G(x,y))$, is an asymmetric tensor field T:

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} \end{pmatrix} \tag{2}$$

T can be decomposed into its symmetric and anti-symmetric components, which measure the scaling and rotation caused by the tensor, respectively. The symmetric component can be further decomposed into isotropic and anisotropic constituents. In [20], the components are combined in the following unified parameterization of the space of 2×2 tensors:

$$\begin{gathered} T = \gamma_d \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \gamma_r \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \gamma_s \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \\ \gamma_d = \frac{T_{11}+T_{22}}{2} \qquad \gamma_r = \frac{T_{21}-T_{12}}{2} \\ \gamma_s = \frac{\sqrt{(T_{11}-T_{22})^2+(T_{21}+T_{12})^2}}{2} \end{gathered} \tag{3}$$

γ_d, γ_r, and γ_s are the strengths of isotropic scaling (dilation), rotation, and anisotropic stretching, respectively. The

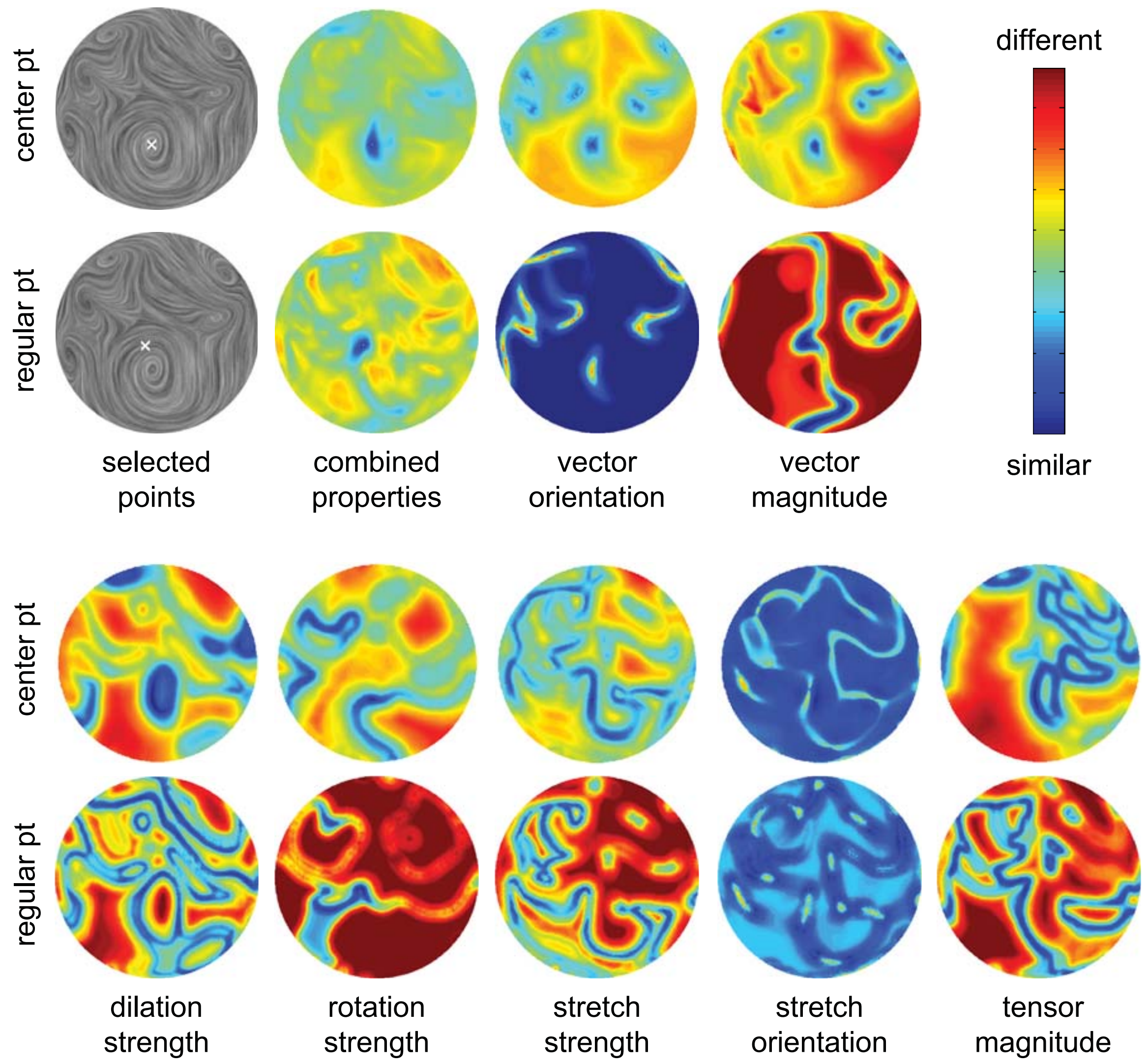


Figure 3. χ^2 difference of a fixed point (1st row) and a regular point (2nd row) to all other points in the same vector field (self-similarity). Top-left: vector field with selected points (white X). Top-2nd from left: χ^2_{total} difference based on combining χ^2 differences of all properties. Remaining columns: χ^2 difference based on individual properties. No single property distinguishes the selected point from all other points, but the combination of properties does as evident by the single small blue cluster in the combined χ^2_{total} difference plot.

orientation of stretch is given by $\begin{pmatrix} T_{11} - T_{22} \\ T_{21} + T_{12} \end{pmatrix}$. In addition to these four first-order properties of the vector field, we also compute the tensor magnitude: $\gamma_d^2 + \gamma_r^2 + \gamma_s^2$. In total, we capture seven properties (two zeroth-order and five first-order properties) in vector spin-images and use them to compare points from one 2D slice of the vector field to the next to track a selected point through a 3D vector field.

5. Vector Spin-Image Parameters

To incorporate the seven properties described in the previous section into an algorithm for tracking points through dense vector fields, we must address the following issues: (1) selecting the appropriate range and resolution for vector spin-images to sufficiently capture the properties, (2) combining the different properties into a unified approach, and (3) updating and adapting vector spin-images as we track changing points through the vector field. We address each issue below.

5.1. Range and Resolution

The selection of the vector spin-image range and resolution for each property depends on the range of values for each property. For example, vector and stretch orientations take on values only between [-1,1], whereas other property values can vary widely in going from the top to the bottom of the diesel engine cylinder as shown in Table 1. In particular, dilation, rotation, and stretch strengths and tensor

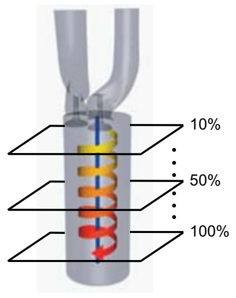


Figure 4. Diesel engine cylinder. 2D planar vector fields are extracted by slicing through the cylinder along its length (*e.g.*, at 10%, 20%, etc.).

magnitude change dramatically from the top to the bottom.

Each bin in the vector spin-image tallies the number of points falling into the bin's range of values for each property. As a result, the overall range of property values in a dataset and the resolution of the spin-image determines the total number of bins and the range covered by each bin. The higher the resolution, the more precise the local descriptor, but also the more expensive to compute and compare. With larger ranges, more bins are required to achieve the same resolution.

To maintain a reasonable number of bins while achieving descriptive distributions, we apply the following strategies: (1) each spin-image in a 2D slice covers only the range of values within the slice rather than defining a global range that covers the entire range of values throughout the dataset; (2) we normalize the range of values to be between 0 and 1 for all positive-valued properties (*e.g.*, magnitude) and between -1 and 1 for all others; and (3) we use the log scale given by Equation 1 to bin vector field properties. In general, we have found a bin resolution of 10 for distance and 20 for all other properties to be sufficient (total of 200 bins). We are able to compare the vector spin-images of one slice to those of the subsequent slice despite (1) and (2) above because of coherence in the vector field. In going from one slice to the next, the range of property values have not significantly changed. Hence comparing their spin-images remains valid. We apply a log scale to bin vector field properties because distribution plots of the properties show a Normal distribution centered around zero. In other words, most points in the dataset have values clustered around zero (with different standard deviations for different properties), and few points with extreme values (at the far ends of the range). As with radius, a log scale ensures that higher resolution is achieved where there is more data and lower resolution in areas with less data.

5.2. Combining Zeroth and First-Order Properties

The seven properties described in Section 4 can be combined into a single vector spin-image of seven dimensions discretized into 20 bins per property (10 for distance). With 7 properties, such a distribution would contain $20^7 * 10$ bins for a total of 12.8 billion bins. Computing, storing, and comparing these high-dimensional vector spin-images would be considerably expensive and wasteful in that there will be many bins with zero or near zero values. Instead, we leverage the additive nature of the difference function as described below.

Because the vector spin-image is a distribution, standard statistical techniques can be used to quantitatively compare them. We use the χ^2 distance (or difference), computed as follows between N-bin normalized histograms, f and g:

$$\chi^2 : D(f,g) = \frac{1}{2}\sum_{i=1}^{N}\frac{(f[i]-g[i])^2}{f[i]+g[i]} \tag{4}$$

In the above equation, each bin contributes equally to the difference. This additive effect enables us to treat each property separately by computing a 2D distribution for each (indexed by distance and one of the seven properties). When comparing points in the vector field, we compute the χ^2 differences between the distributions for each property and combine them into one total difference value:

$$\begin{gathered}\chi^2_{total} = \textstyle\sum_{j=1}^{k} w_j \chi^2_j \\ \textstyle\sum_{j=1}^{k} w_j = 1.0\end{gathered} \tag{5}$$

For seven properties, $k = 7$. By treating the vector field properties separately, we can weigh some properties more than others via w_j in the equation above. Through empirical analysis, we have found that vector orientation as captured by the dot-product is critical to locating fixed points. This is intuitive since the vector at a fixed point is zero-valued unlike all other points. On the other hand, simply capturing vector orientation is not discriminating enough because all fixed points will have the same local distribution as described in Section 4 and shown in Figure 3 where we also show that vector orientation alone is not sufficient to distinguish between regular points. We have found that weighing vector orientation by 0.7 and the remaining properties equally by 0.05 effectively tracks both fixed and regular points. We have used this weighting in all the examples of this paper and the accompanying video.

5.3. Adaptive Distributions

A key advantage of distribution-based tracking is the ability to adapt as the feature changes. For example, if a selected fixed point disappears, we can extrapolate its position afterwards. To ensure that the vector spin-image for a selected point adapts as the vector field evolves from one slice to the next, we update the range of vector field properties and average the vector spin-images that have a difference of 1% or less with respect to the query spin-image.

Position of slice along engine cyl.	vector orientation	vector magnitude	dilation strength	rotation strength	stretch strength	stretch orientation	tensor magnitude
10%	[-1,1]	[0, 5.9]	[-31.8, 60.2]	[-39.6, 47.2]	[0, 62.6]	[-1,1]	[0, 87.3]
20%	[-1,1]	[0, 3.6]	[-20.1, 38.5]	[-26.0, 26.9]	[0, 42.5]	[-1,1]	[0, 59.3]
30%	[-1,1]	[0, 2.8]	[-12.9, 35.7]	[-25.1, 24.6]	[0, 37.3]	[-1,1]	[0, 51.7]
40%	[-1,1]	[0, 2.1]	[-15.2, 28.8]	[-22.5, 22.7]	[0, 31.2]	[-1,1]	[0, 43.0]
50%	[-1,1]	[0, 1.6]	[-13.9, 22.3]	[-18.0, 18.0]	[0, 24.3]	[-1,1]	[0, 33.4]
60%	[-1,1]	[0, 1.2]	[-10.7, 16.5]	[-13.5, 13.5]	[0, 18.1]	[-1,1]	[0, 25.0]
70%	[-1,1]	[0, 0.9]	[-6.1, 12.3]	[-9.9, 9.9]	[0, 13.4]	[-1,1]	[0, 18.5]
80%	[-1,1]	[0, 0.7]	[-3.9, 8.8]	[-7.0, 7.0]	[0, 9.7]	[-1,1]	[0, 13.3]
90%	[-1,1]	[0, 0.5]	[-1.9, 6.1]	[- 4.7, 4.9]	[0, 6.7]	[-1,1]	[0, 9.2]
100%	[-1,1]	[0, 0.4]	[-1.3, 4.1]	[- 3.0, 3.3]	[0, 4.5]	[-1,1]	[0, 6.1]

Table 1. Range of Zeroth and First-Order Property Values

As described in Section 5.1, the values of vector field properties are coherent in space and time, enabling us to update the range of values incrementally as we track. At each slice, we do not compare the vector spin-images of the slice points to the vector spin-image of the selected point in the first slice. Instead, we compare them to a spin-image that is the average of the most similar points from the previous slice. By averaging vector spin-images, we are adapting the query spin-image as we track and accounting for the fact that there may not be a single corresponding point. In Figure 5, the white clusters are points with a difference of less than 1% within the slice ($\chi^2 \leq 0.01 * (\chi^2_{max} - \chi^2_{min})$), where χ^2_{max} and χ^2_{min} are the maximum and minimum differences within the slice. The average position of this cluster is marked in tracking.

6. Tracking Results and Discussion

We tested our distribution-based approach to tracking points on the simulated steady-state flow within the combustion chamber of a diesel engine. Although the data is simulated, it is what engineers use to analyze engine design since there is no way to physically capture flow fields inside an engine cylinder. Engineers typically slice through the cylinder along its length (Figure 4) to analyze the simulation, resulting in the circular flow patterns of Figures 1, 3, and 5. The ideal flow pattern is a helix extending the length of the cylinder. This motion optimally mixes air and fuel leading to a more efficient combustion process [8].

In Figure 5, we select two fixed points (a center and a saddle point) and a regular point in a slice near the intake valves. In the top row, we show the vector field color-coded using eigenvalue visualization based on all properties as described in [20] and track the selected point (white X's) through several slices (frames) down the length of the engine cylinder. We have omitted several slices in between due to space limitations. The 2nd, 3rd, and 4th rows in the figure show the χ^2_{total} difference for each tracked point at each frame using the average vector spin-image as we move from slice to slice. The white region within each plot are points with a difference of less than 1%. These points contribute to the average vector spin-image used as the query spin-image in the next slice. Overall, fixed points are tracked consistently with slight deviations in some frames.

The center point (#1) is tracked as it moves upwards and then eventually disappears by the 8th frame shown. By examining the eigenvalue color-coding, we see that the center point starts out in the center of a green region. As the region moves and transforms in shape, the tracked point moves along with it while remaining in its center. The χ^2 difference plots for the center point (2nd row) show that it becomes increasingly similar to other points in the vector field as it disappears.

A similar result is seen in the χ^2 difference plots for the saddle point (3rd row). The saddle point (#2) begins in a small green region that disappears, subsumed by its neighboring red and orange regions. The tracked point then remains on this boundary between the red and orange region and eventually merges with a nearby ridge. The regular point (#3) is similar to many points in the first slice as seen by the large blue regions in the χ^2 difference plots (4th row). Intuitively, this makes sense since there are many more regular points in the data. By accounting for all seven vector field properties, however, we distinguish the selected point from other regular points as evident in the small white cluster of the χ^2 difference plots. The regular point begins at the intersection of blue, red, and orange regions in the eigenvalue color-coding. The blue region eventually disappears, but the tracked point consistently moves with these regions and remains at the boundary between the red and orange regions even after the blue region disappears.

Our results are in contrast to what happens when advection is used to move particles forward from one slice to the next (Figure 1). There, the selected points do not stay fixed relative to the eigenvalue color-coded regions, and instead, move along pathlines.

We have tested many other fixed and regular points within the vector field with similar results. In the accompanying video, we show tracking of the center and regular points and include additional frames beyond those shown in the figure. The movement of the tracked points is more evident in the video. The video also includes tracking of a point on a ridge which eventually disappears (example not shown in Figure 5).

In terms of complexity, computing and comparing vector spin-images is linearly dependent on the number of bins (determined by the range and resolution of each property). We have found a bin resolution of 10 for distance and 20 for all other properties to be sufficient. For each slice, the 20 bins span the full range of values of each property using a log scale as described in Section 5.1. The vector spin-image radius ranges from 0 to 60 pixels for the engine cylinder dataset (approximately 15% of the cylinder width). The average time to compute the vector spin-images of all seven properties for a single point is 1.6 microseconds on an Intel T2050 1.6 GHz processor (total of about 15 seconds for each slice containing 9,659 points). The average time to compare the vector spin-images of a single point to all other vector spin-images in a slice (9,659 pairs of points compared) is 0.5 secs.

7. Future Work

We plan to explore ways to reduce the memory footprint of vector spin-images to make it possible to combine all seven properties into a single high-dimensional distribution rather than combine the χ^2 differences of each property as we now do. This may be possible by examining the distribution of each property across the entire dataset and eliminating bins that contain few points.

References

[1] M. Ankerst, G. Kastenmuller, H. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. *Proc. of 6th International Symposium on Spatial Databases*, 1999. 2

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 24(24):509–522, 2002. 2

[3] J. Ebling and G. Scheuermann. Clifford convolution and pattern matching on vector fields. *Proc. of IEEE Visualization*, pages 193–200, 2003. 2

[4] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. *Proc. of European Conf. on Computer Vision (ECCV)*, 3:224–237, 2004. 2, 3

[5] C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3D time-dependent datasets. *Proc. of IEEE Visualization*, pages 329–336, 2004. 2

[6] J. Helman and L. Hesselink. Surface representations of two- and three- dimensional fluid flow topology. *Proc. of IEEE Visualization*, pages 6–13, 1990. 2

[7] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(5):433–449, 1999. 2, 3

[8] R. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating swirl and tumble flow with a comparison of visualization techniques. *Proc. of IEEE Visualization*, pages 51–58, 2004. 6

[9] Y. Lavin, R. Batra, and L. Hesselink. Feature comparisons of vector fields using earth mover's distance. *Proc. of IEEE Visualization*, pages 103–109, 1998. 2

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. on Graphics*, 21(4):807–832, 2002. 2

[11] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. The state of the art in flow visualisation: feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003. 2

[12] F. Reinders, F. Post, and H. Spoelder. Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer*, 17(1):55–71, 2001. 2

[13] F. Reinders, A. Sadarjoen, B. Vrolijk, and F. Post. Vortex tracking and visualization in a flow past a tapered cylinder. *Computer Graphics Forum*, 21(4):675–682, 2002. 2

[14] G. Scheuermann, H. Krüger, M.Menzel, and A. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 4(2):109–116, 1998. 2

[15] H. Theisel and T. Weinkauf. Vector field metrics based on distance measures of first order critical points. *Journal of Winter School of Computer Graphics (WSCG)*, 10:121–128, 2002. 2

[16] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. *Proc. of IEEE Visualization*, pages 159–166, 2001. 2

[17] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-based visualization of time-dependent 2D vector fields. *Data Visualization, Proc. of IEEE TVCG Symposium on Visualization*, pages 117–126, 2001. 2

[18] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers and Graphics*, 26:249–257, 2002. 2

[19] L. Xu and H. Dinh. A local descriptor for finding corresponding points in vector fields. *IAPR International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 3

[20] E. Zhang, H. Y. Z. Lin, and R. Laramee. Asymmetric tensor analysis for flow visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(1):106–122, 2009. 3, 6, 8

[21] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics (TOG)*, 25(4):1294–1326, 2006. 2

[22] X. Zheng and A. Pang. 2D asymmetric tensor fields. *IEEE Visualization*, pages 3–10, 2005. 3

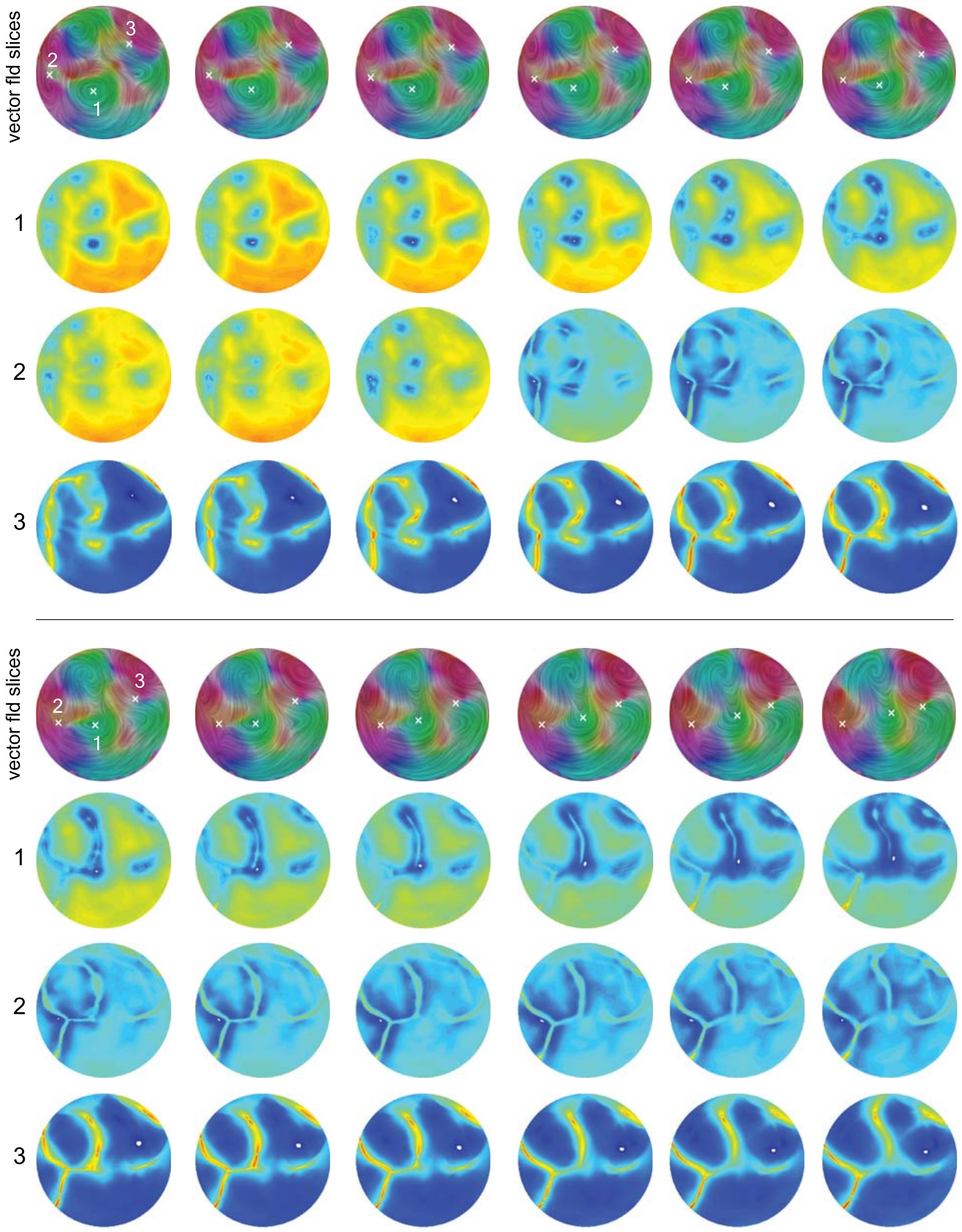


Figure 5. Tracking center (1), saddle (2), and regular (3) points through the engine cylinder flow field. Top rows: vector field slices are ordered from left to right, top to bottom and color-coded based on eigenvalue visualization [20]. Other rows: χ^2_{total} difference between points on each slice and the average vector spin-images from the previous slice for center (2nd row), saddle (3rd row), and regular (4th row) points.