A Local Descriptor for Finding Corresponding Points in Vector Fields

Liefei Xu and H. Quynh Dinh
Department of Computer Science, Stevens Institute of Technology
lxu1@stevens.edu, quynh@stevens.edu

Abstract

*Vector fields may come from video data (via optical flow and tracking), from weather phenomena (*e.g.*, wind speed and direction), and from medical imaging. An important component in analyzing this data is to be able to quantitatively compare different points within a vector field or across different fields of the same type. We present a novel local descriptor to compare individual points in a vector field and rank them based on which are most similar to a selected point. The descriptor captures the statistics of the neighboring vector field around a central point and is discriminating, robust to noise, and efficient to generate and compare.*

1 Introduction

Vector field data is becoming more prevalent in video analysis (optical flow and tracking), weather monitoring (*e.g.*, wind and water speed and direction), and medical imaging (*e.g.*, phase-contrast magnetic-resonance angiography captures 3D velocity fields of moving tissue [12]). An essential component in analyzing this data is to quantitatively compare different points within a vector field or across different fields of the same type. Such comparisons will enable us to evaluate how the vector field has changed at a point, and to find point correspondences that can be used to register vector fields when the transformation to align them is unknown.

Our contribution is a rotationally invariant, robust local descriptor that captures the distribution of vectors around a central point. In Section 4, we show that our local descriptor is precise through self-similarity tests that compare selected points in planar vector fields to all other points in the domain.

2 Related Work

Vector field analysis identifies singularities (critical points where the magnitude of the vector vanishes), separatrices, and periodic orbits. Singularities are further classified into sinks, sources, saddles, focus, and center points using the Jacobian. There is much research in the areas of vector field analysis and singularity classification, including [5, 10, 13, 14, 3, 15]. Previous work in comparing vector fields have focused on defining a metric to compute the distance between singularities using alternate phase plane coordinate systems [7, 11]. However, many naturally occuring vector fields do not contain singularities, and non-singular points can exhibit interesting features such as the ridge formed where opposing vectors meet in Figure 3. Such ridges often occur in water currents. Instead of relying on singularities, we compute a local descriptor for every point in the field. Although we do not explicitly track singularities, our approach is able to locate them by virtue of finding points with distinctive local descriptors.

Geometric, or shape, distributions provide a way of comparing shapes based on statistical properties [1, 8]. 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.

Statistics on vector fields from optical flow have been explored in [9] where the goal was to obtain a global statistic for use as a prior in optical flow computation, not to define a local descriptor. We focus on local distributions that can be used to compare points within and between vector fields. In shape matching, successful approaches to generating local distributions include spin-images [6] and shape contexts [2, 4] which store the distribution of neighbors surrounding a central point. We now describe how we adapt these concepts to develop a local distribution for planar vector fields.

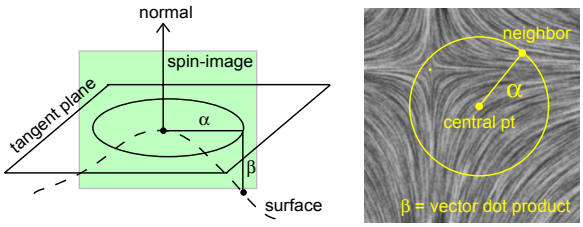

Figure 1. Left: Spin-image for surface points in [6]. Right: Our new vector spin-image for planar domains bins neighboring points based on distance α from the central point and dot-product β between the neighbor's vector and the central point's vector. All points on the yellow circle have a common α but β may differ.

3 A Local Distribution for Vector Fields

In developing a local descriptor for vector fields, our goals are similar to those for surface points. The descriptor must be robust to noise, invariant to transformations, discriminating, and efficient to generate and compare. We apply concepts from shape contexts and spin-images to create a local descriptor for vector fields which we call a *vector spin-image*. In Figure 1, 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) [6]. 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, we define the vector spin-image indices as 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]:

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

In the above equation, r_{min} is the 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. Bins far from the center will have high counts, but because each bin contributes equally to the difference (Equation 2), a point in one such bin actually has less effect on the difference.

Histograms are normalized prior to computing the difference between them. Without normalization, the computed difference between points near the boundary of the field and all other points is incorrect due to missing data (boundary histograms have far fewer data points resulting in lower bin values). Normalization greatly reduces these effects, but even with normalization, there are residual errors at the boundaries as shown in Figure 2 (top-left). Inversely weighting the bins by the bin area (which increases at larger radii) also does not eliminate the boundary errors. We have found two approaches that solve the boundary problems. The first (not shown) normalizes each bin by the total number of sample points gathered within the bin's concentric ring which is effectively an approximation of bin size. In the second approach, we borrow from image processing and reflect the vector field around edges to fill in missing data. By doing so, we are essentially making the assumption that the missing data is statistically similar to areas of the vector field nearby. Figure 2 shows that using reflections completely eliminates boundary errors and does not adversely affect the computed difference anywhere on the interior of the field.

In Section 4, we use reflection to generate spin-images and focus on illustrating the effects of support radius and bin resolution, as well as the precision of our algorithm in determining the similarity of a vector field with respect to a selected point in the field.

4 Results and Conclusions

We generate synthetic vector fields using Zhang *et al.*'s design tool [15]. We tested our algorithm on simple fields with a few different singularities, on fields without singularities but with distinctive features, and on complex fields designed for painterly rendering [15]. Source singularities are green; sinks are red; saddles are yellow; centers are pink (attracting) and cyan (repelling); and regular vectors are cyan arrows. All vector fields are 512x512 in size. For each field, we perform a *self-similarity* test wherein a point in the field is selected (indicated by X or a dot in the images) and compared to all other points in the field using the χ^2 distance. The χ^2 distance between N-bin normalized histograms, f and g, is:

$$\chi_2 : D(f,g) = \frac{1}{2}\sum_{i=1}^{N}\frac{(f[i]-g[i])^2}{f[i]+g[i]} \qquad (2)$$

4.1 Effects of Support Radius and Resolution

Figure 2 shows the effects of reducing the support (maximum) radius of vector spin-images (top row) and of reducing histogram resolution (bottom row). We use three sets of radius parameters ($r_{min} = 5$, $r_{max} = 60$; $r_{min} = 3$, $r_{max} = 40$; and $r_{min} = 2$, $r_{max} = 20$ pixels) and three histogram resolutions (10×10, 8×8, and 5×5). We also tested rectangular spin-images with higher radial resolution than dot-product and vice versa.

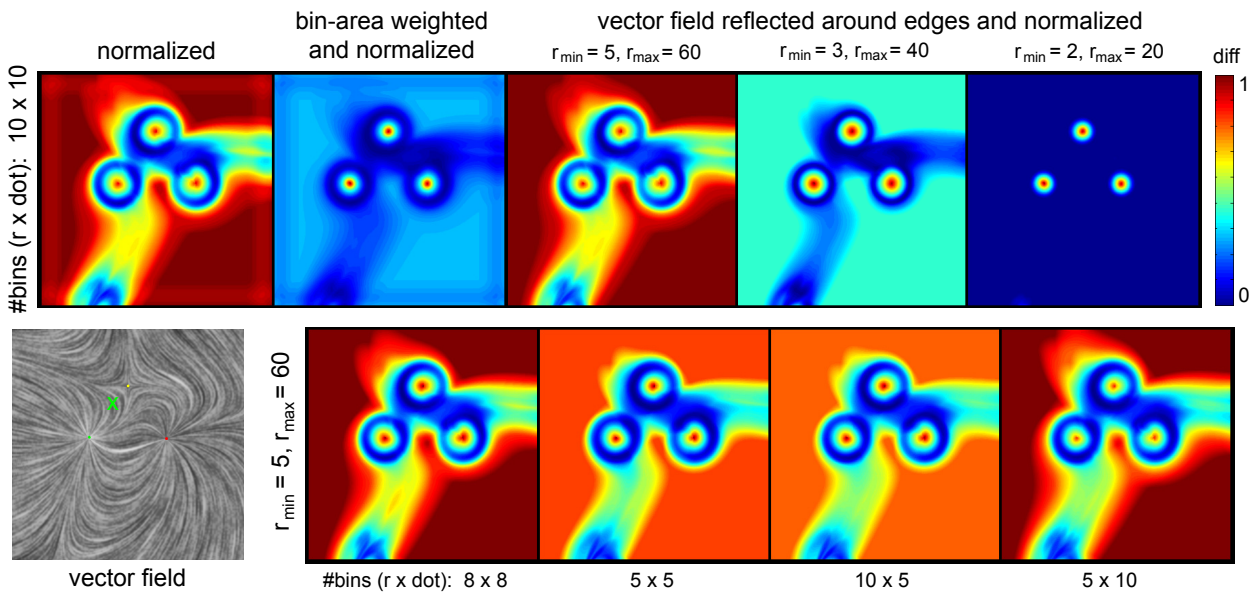

Figure 2. Self-similarity of a test vector field (lower-left) to a selected point (X). Top: effects of bin normalization and decreasing r_{max}. Bottom: effects of bin resolution on a constant support ($r_{max} = 60$). Red to blue = increasing similarity.

The time to generate spin-images which depends on the max radius far surpasses the time to compare them. Computing a spin-image for $r_{max} = 60$ is $4\times$ longer than for $r_{max} = 40$ which is $4\times$ longer than for $r_{max} = 20$. However, with a small support radius, the descriptor becomes less discriminating. In the self-similarity test in Figure 2, many more points are found to be similar to the selected point when $r_{max} = 20$. In general, the measured differences become coarser with fine differences disappearing as the support radius is reduced. Depending on the application, the more efficient small support radius may be sufficient. For example, if the goal is to locate singular points, a small support does just as well as shown in Figure 2 (top row).

Spin-images for all points in a vector field can be pre-computed and stored in a database for comparison to other fields later. The number of comparisons grows significantly when the goal is to register multiple points between different vector fields. Hence, it is important to reduce the time for computing the difference between vector spin-images. The time complexity for the χ^2 distance (and most other distance metrics) is linear with respect to the histogram size. We can reduce the computation time by using lower resolution vector spin-images. The bottom row of Figure 2 shows the effects of doing so. A significant difference in the self-similarity test is seen with 5×5 spin-images when compared with the results using 10×10 spin-images (8×8 gives nearly identical results). By reducing the resolution only in the radial dimension and maintaining high resolution for dot-product, we obtain essentially the same results as with our highest resolution vector spin-images. Note that reducing dot-product resolution while maintaining high radial resolution produces the opposite effect (self-similarity results are similar to using the lowest resolution histograms). This implies that the distribution of vector orientation is a key component in distinguishing between different points in the vector field.

4.2 Identifying Distinct Features

The results in Figure 2 reveal that our local descriptor cannot distinguish between different types of singularities (the test vector field contains a sink, source and saddle) but can precisely locate singularity position. The magnitude of the vector at the singularity vanishes, resulting in zero-valued dot-products and a local distribution that is the same for all types of singularities but differs from all non-singular points.

In Figure 3, we show that our local descriptor is effective in extracting distinct features from fields that contain no singularities. The first example has a ridge formed by two opposing vector fields. By selecting a point on the ridge (1), all other ridge points are highlighted as similar. The second selected point (2) is just off the ridge, and shows the sensitivity of our algorithm because the ridge is now no longer marked as similar. In the second example, we have selected two points – (1) a ridge-like feature and (2) in an area where the vectors are uniformly straight. In both cases, our algorithm is

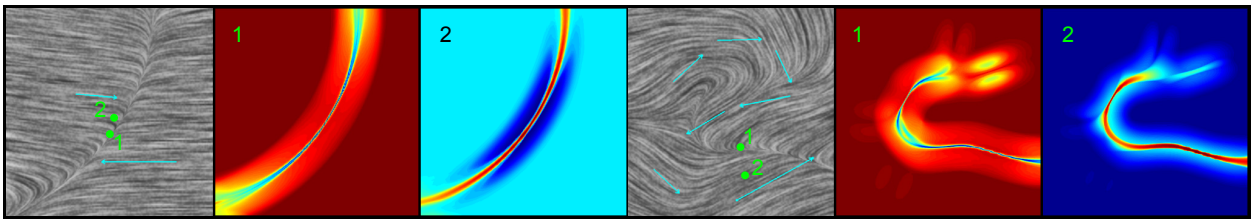

Figure 3. Vector fields with no singularities but with distinctive features. Red to blue = increasing similarity to green dots.

able to segment the field into two distinct regions.

In Figure 4, we apply our algorithm to complex vector fields from painterly rendering [15]. On the left, we choose a distinct point (the tip of a cat's ear in the image data). On the right, we chose a point in a swirl. The self-similarity tests highlight similar areas such as the lower-middle area in the field on the right where the vectors turn. In future work, we will cluster vector spin-images to more efficiently compare whole vector fields.

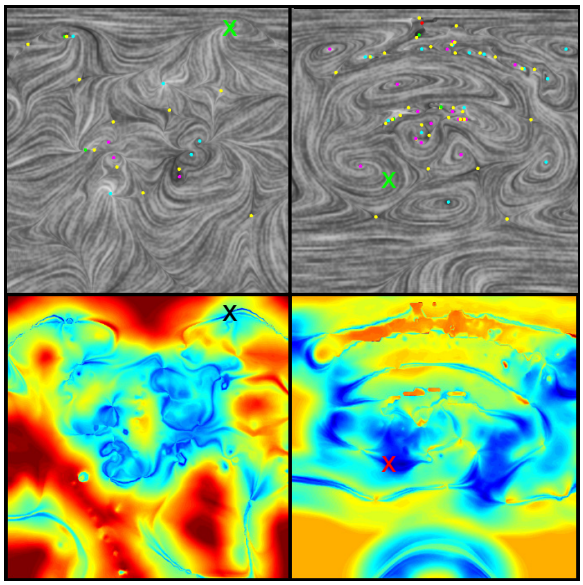

Figure 4. Vector fields from painterly rendering [15]. Bottom: self-similarity to selected points (X).

5 Acknowledgements

We thank Eugene Zhang for the vector field design tool and painterly rendering data. This work was funded in part by NSF CreativeIT Award# IIS-0742440.

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] 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.

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

[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.

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

[6] 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.

[7] 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.

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

[9] S. Roth and M. Black. On the spatial statistics of optical flow. *Int'l J. of Computer Vision*, 74:33–50, 2007.

[10] 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.

[11] H. Theisel and T. Weinkauf. Vector field metrics based on distance measures of first order critical points. *Journal of WSCG*, 10:121–128, 2002.

[12] M. Tovar. Vector-field classification in magnetic-resonance angiography. *AMIA Sym.*, pages 926–930, 1998.

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

[14] 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.

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