EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005), pp. 1–7
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

Vector Field Analysis and Visualization through Variational Clustering

Alexander M^{c}Kenzie[1], Santiago V. Lombeyda[2] and Mathieu Desbrun[2]

[1] University College London
[2] California Institute of Technology

Abstract
Scientific computing is an increasingly crucial component of research in various disciplines. Despite its potential, exploration of the results is an often laborious task, owing to excessively large and verbose datasets output by these simulation runs. Several approaches have been proposed to analyze, classify, and simplify such data to facilitate an informative visualization and deeper understanding of the underlying system. However, traditional methods leave much room for improvement.

In this article we investigate the visualization of large vector fields, departing from accustomed processing algorithms by casting vector field simplification as a variational partitioning problem. Adopting an iterative strategy, we introduce the notion of vector "proxies" to minimize the distortion error of our simplification by clustering the dataset into multiple best-fitting characteristic regions. This error driven approach can be performed with respect to various similarity metrics, offering a convenient set of tools to design clear and succinct representations of high dimensional datasets. We illustrate the benefits of such tools through visualization experiments of three-dimensional vector fields.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: Flow Visualization

1. Introduction

With the continued advance of computer architectures, unprecedented computational processing power is available to any scientist whose research may benefit from computer modeling or simulation. Many disciplines have adopted such methods, motivating sub-branches in science and engineering; molecular modeling and computational fluid dynamics (CFD) are examples of this. Today, computers allow researchers to perform increasingly complex 3D simulations using extremely fine grids to capture even the most subtle of detail. These simulation runs typically generate many gigabytes of data, whereby post processing and visualization become critical steps in the pipeline. The demand for tools to analyze and extract the relevant information from these datasets has been recognized, and many approaches and techniques proposed.

We focus on the representation of 3D vector fields, a challenging topic in scientific visualization for which no natural representation exists. Unlike geometry, color, or texture, vector fields are difficult to depict clearly, and thus warrant special attention to develop an intuitive visual understanding: given a static field, we are confronted with up to six dimensions of data (position and affiliated vector in 3D) that must be projected onto a 2D computer screen. We begin by reviewing the strengths and weaknesses of existing visualization tools as a motivation for our work.

1.1. Brief Review of Visualization Techniques

Despite the fairly recent invent of computer modeling for scientific purposes, the important task of visual depiction of flow fields (to convey, inspect, and analyze their content) has some history, in which a few methods have become popular.

Hedgehog and Glyphs Since an arrow icon is typically used to depict a vector, a natural approach is to define various base points in our field at which to sample the vectors, and display the associated arrow icon. Samples can be taken throughout the field and allows us to present a complete view of both the magnitude and direction in a single image. Unfor-

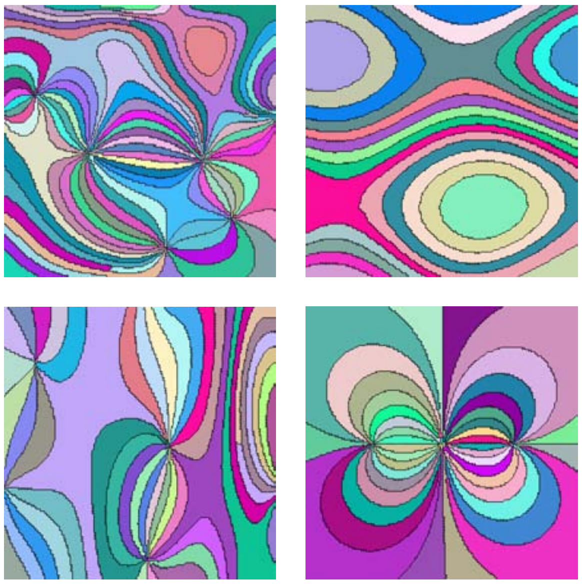

Figure 1: *Variational Segmentation: Error-driven clustering is performed on various flow fields using the $\mathcal{L}^2$ metric for vector comparisons. Each color represents an area in which both direction and magnitude of the field is very similar, providing an otherwise hidden insight into the system.*

tunately, locality is a significant concern with this concept: each sampling only depicts the flow at a specific point in the field; sub sampling the field will miss small yet important details, while increasing this resolution will lead to a very cluttered visualization. In [LAK*98], arrowheads are replaced with more advanced glyphs in attempt to better present multivariate data.

Contraction and Topological Structure Rather than visualizing vector fields directly, many have proposed to "contract" the data into scalar quantities that represent *salient* characteristics of the field. Examples of important physical features that can be presented in this fashion include the magnitude, the divergence, or the vorticity of the flow. In the same vein, the topological structure (critical points, separation lines–see [GL91]) can be depicted as a very coarse, yet highly informative representation of a vector field's content.

Streamlines An alternate approach defines field lines that are always tangential to the flow [TB96, SM02]. Given a vector field that doesn't vary in time, we could imagine dropping a particle into the field at a specific point and tracing its path as the flow field pushes and pulls the particle along the direction of the field. Streamlines can often provide an elegant solution in 2D, but suffer from similar locality problems as the arrow plot - a poor placement of particles could miss important details, for example; yet, too many streamlines will again cause confusion.

LIC Line Integral Convolution [CL93] has become a popular method for flow visualization in 2D. A texture, typically an image of random noise, is applied to the vector field and pixels of this image are advected along the flow and interpolated to generate a new image of the noise after being distorted by the field. LIC can also be used to depict vector fields that are defined on arbitrary surfaces embedded in 3D. In both cases, resulting images capture the details of 2D flows extremely well.

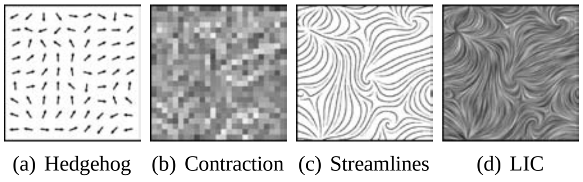

Figure 2: *Popular techniques of vector field visualization applied to a non-trivial 2D dataset.*

Many of these visualization strategies have proven extremely effective on visualization of 2D flow fields, and even time varying 2D vector fields. And in [LKD*01], a mechanism was presented to evaluate the effectiveness of any given strategy. However, none of these basic methods translate well into three-dimensional flow field visualization. Simple translation of space filling methodologies, such as glyphs or LIC, fail because perceptually the end viewer requires opaque 3D structures and reasonable depth cues in order to discern among the layers of data being displayed. Approaches that use volume rendering techniques on 3D LIC [RSHTE99] or advanced stream surfaces with texture hints (based on [GIS03]) and lighting hints [MTHG03] have improved the efficacy of 3D vector visualization, although these techniques quickly produce displays that are too taxing to comprehend for anything beyond the most simple of flows. For further general discussion on recent advances, please consult [PVH*02]

1.2. Vector Field Simplification through Clustering

A remedy for reducing the clutter is to minimize the amount of data presented to the user while converting that which *is* presented into simple yet descriptive iconic elements. A common approach is to employ clustering of similar contiguous vectors in order to represent large regions of the flow by a single "average" vector. This was used in [TV99] and also [GPR*00]: an extension that uses the Cahn Hillard model of physical-based clustering as well as a phase separation model. In more recent work, [DW04] use a Voronoi tessellation based algorithm for cluster creation and meanwhile in [GPR*04] we see the concept of algebraic multigrids to find stable clusters in 2D and 3D. As a significant pitfall, however, these clustering schemes generally fall short by concentrating on results in 2D, where other methods have proven more effective.

1.3. Contributions

We hereby propose a vector clustering technique based on K-means by extending the work of [DW04], which is not only efficient and stable, but can also use different, physically-based metrics in order to provide a meaningful segmentation of the input vector field; we explore distance metrics based on direction, gradient, curl, and divergence to offer a wide range of tools applicable to various vector visualization goals in 2D and 3D. After demonstrating the efficiency of this new technique, we use it as a basis for visualizing turbulent vector fields.

2. Variational Segmentation

To depart from most of the traditional methods, we cast such flow field data mining as a *variational* partitioning problem. Given a large, verbose dataset our approach endeavors to cluster together areas of the field with low 'entropy': ideally "similar" vectors are grouped together because the relative importance that each individual vector carries with respect to the entire field is very low. By simplifying a vector field into several regions, a global distortion error can be defined and an attempt made at minimizing this quantity by applying an iterative strategy that produces locally optimal partitions. The entire segmentation process as we describe it next is directly inspired by [CSAD04] which provides a deeper discussion of all the steps involved, but in the context of geometry instead of vector fields; we strongly recommend referring to this paper for further explanation.

2.1. Input Data and Concepts

Discrete Vector Fields Our algorithm is applicable to discrete vector fields: computers—being machines of finite precision—are inherently more suited to discrete data rather than continuous representations. In the discrete setting, a mesh delimits the surface or volume in which a vector field is defined. For each primal element of this mesh (triangle in 2D, tetrahedron in 3D), there exists an associated vector - this type of data is known as a *piecewise constant* field since the flow field is considered constant within each primal element. It is this geometry (and associated field) that we attempt to simplify through clustering.

K-partitioning The idea of clustering vectors of a flow field into a partition to help approximation has already been used many times in scientific visualization [TV99, GPR*00]. We entertain the idea that an approximating vector is essentially a surrogate linear approximant for a set of originally grouped vectors that share similar characteristics. In this context, clustering a vector field into a partition with k regions appears to be a natural way to efficiently resample our data. Each region $\mathcal{R}_i$ of a partition $\mathcal{R}$ can then be summarized by an "average" vector **proxy** V_i (*average* with respect to a given distortion metric). Traditionally the partitioning is achieved in a greedy fashion, and although we base our approximation on partitioning too, we will see in the following sections that our method iteratively seeks a *partition that best represents the dataset*: this variational nature will make the results more striking due to their near-optimal qualities.

2.2. Defining Local Distortion Measures

Now that we have a representative vector proxy for each region of the mesh, a distortion error is defined that determines how close the simplification is to our original flow field dataset. To find the error of an initial input vector, we simply compare it with its newly associated representative vector proxy V_i and check the local deviation from this average, i.e., the local distortion. Thus by integrating the distortion error between each vector in a region $\mathcal{R}_i$ and its vector proxy V_i (i.e., by summing the difference between a vector and its associated proxy and weighting the error proportionally to the area (in 2D) or the volume (in 3D) of its primal element), we obtain the total error in a given region $\mathcal{R}_i$. We can then compute a global error for the whole vector field by adding together the total distortion of every region $\mathcal{R}_i$. Let us now put it in mathematical terms: given an error metric E, a desired number of proxies k, and an input mesh M, we call the **optimal vector proxies** a set V of proxies V_i associated with the regions $\mathcal{R}_i$ of a partition $\mathcal{R}$ of M that minimizes the total distortion:

$$E(\mathcal{R},V) = \sum_{i=1..k} E(\mathcal{R}_i, V_i) \qquad (1)$$

If we consider using $\mathcal{L}^2$ as our error metric E, the $\mathcal{L}^2$ error of a vector proxy V_i and the associated region $\mathcal{R}_i$ is simply the distance of the vectors and their representative, integrated over the volume (or surface) of the region:

$$E_{\mathcal{L}^2}(\mathcal{R}_i, V_i) = \iiint_{x\in\mathcal{R}_i} \|\mathbf{v}(\mathbf{x}) - \mathbf{V_i}\|^2 dx \qquad (2)$$

In a discrete implementation, we compute the $\mathcal{L}^2$ error of a region as the sum of distortions between all primal elements $P_i \in \mathcal{R}_i$ (with volume $|P_i|$ and vector $\mathbf{v_i}$) and the representing vector proxy:

$$E_{\mathcal{L}^2}(\mathcal{R}_i, V_i) = \sum_{i\in\mathcal{R}_i} \|\mathbf{v_i} - \mathbf{V_i}\|^2 |P_i| \qquad (3)$$

Now for a region $\mathcal{R}_i$, the optimal vector proxy V_i is simply $(\sum_{P_i\in\mathcal{R}_i} |P_i|\mathbf{v_i})/T$ where T is the total 3D volume (or area in 2D) of the region. Using this distance metric to drive the iterative partition optimization (detailed in Section 2.4), we observe physically relevant partitions such as shown in Figure 1.

2.3. Defining Higher-Order Measures

To compute first-order-metric clustering based on divergence, gradient, or curl, we must calculate these quantities for each primal element P. Generalizing the distortion measures specified in Section 2.2, we consider first-order metrics over *piecewise-linear* vector fields (rather than piecewise-constant). In this setting, flow is defined at each vertex in our mesh, and therefore the div, grad and curl will be constant per primal element P. At a given point x, our flow f

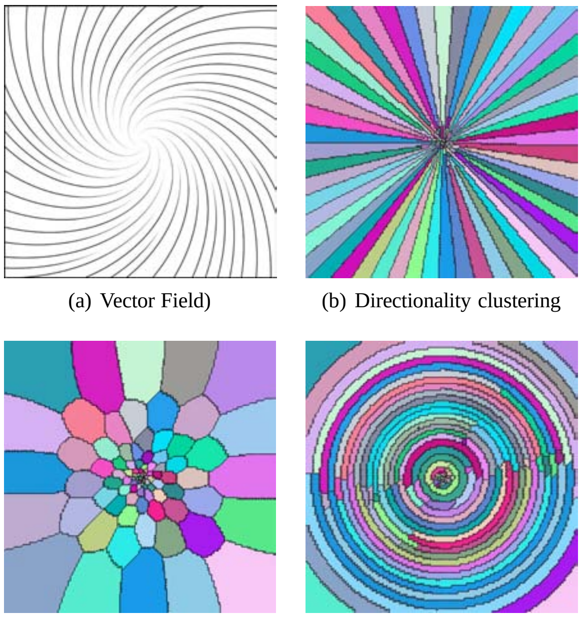

Figure 3: *Application of various distortion metrics, 100 proxies.*

is an interpolation of the field defined by the vertices of the encompassing element:

$$f(x) = \sum_i \phi_i(x) f_i \tag{4}$$

with ϕ_i being the piecewise-linear basis function valued 1 at node x_i, and 0 at all other nodes of P, and f_i being the value of f at node x_i. Due to the local support of the basis functions ϕ_i, the value of f within a tetrahedron which is defined by $(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ is simply: $f = \phi_{i_1} f_{i_1} + \phi_{i_2} f_{i_2} + \phi_{i_3} f_{i_3} + \phi_{i_4} f_{i_4}$.

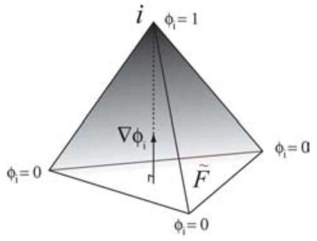

Figure 4: *Basis functions: Value of ϕ_i over a tetrahedron is shown via the color gradient. The vector $\nabla\phi_i$ has length $1/h$ where h is the height of the tetrahedron from base face $\tilde{F}$.*

Notice that $\nabla\phi_i$ is nothing more than the vector orthogonal to the face opposite to i, $\tilde{F}$ in the direction of i, where $|\nabla\phi_i| = (3 \cdot |P|)/|\tilde{F}|$. The direction of $\nabla\phi_i$ is easily found as the cross product of two edges of $\tilde{F}$. In 2D, the same concepts apply: $\tilde{F}$ is replaced with an edge $\tilde{e}$ and $|\nabla\phi_i| = (2 \cdot |P|)/|\tilde{e}|$. Having computed $\nabla\phi_i$, we are now able to calculate the gradient of f (∇f), or its divergence ($\nabla \cdot f$), or its curl ($\nabla \times f$) very easily; please refer to [PP00] and [TLHD03] for details. Finally, we can store the regional (volume weighted) mean values of these quantities in the vector proxy for use in partitioning, the errors being defined in a similar fashion to that of $E_{\mathcal{L}^2}$ in Equation 2. Note that these zeroth and first order metrics for vector comparison in 2D and 3D are a useful extension to the metric defined in [DW04]: not only do they allow physically-relevant clustering, but they also permit higher-order clustering. Clustering together vectors with similar curl components to naturally detect vorticies and eddies is possible, as is comparing divergence which allows us to point out sources and sinks. Figure 3 illustrates how a single dataset may look through these different metrics. The potential for any variety of other metrics exist, where necessary.

2.4. K-Means Algorithm for Discrete Vectors

Distortion-driven Flooding The procedure to build a k-partition—connected and non-overlapping k regions—of our input mesh is straightforwardly achieved through a rapid flooding process that makes locally optimal decisions in an attempt to reduce our global distortion, as defined in [CSAD04]. To bootstrap the process, we pick k random seed elements in our mesh and for each; we assign a vector proxy defined by this seed's vector data. Then, for each seed, we add its neighbors to a priority queue, where priority is given to neighbors that have the lowest distortion with respect to the seed's proxy. As we remove elements from the queue, we assign that element to its vector proxy of low distortion and then continue by adding its neighbors to the queue in a recursive fashion until all elements have been assigned to a region.

Optimizing the Partition The K-means algorithm (or more precisely, the Lloyd's clustering algorithm) can then be applied—a deterministic and fixed point iteration that provides near-optimal clustering for a surface we wish to split into k regions. Conceptually, the idea is simple: after defining k random centers, all the data points on the surface are partitioned into k regions by assigning each point to its nearest center. Then, the algorithm updates the centers to be the centroids of their associated regions before starting a new partition with these new centers. This process is repeated until a stopping criterion is met. It is easy to understand how this generalized framework can be applied to our distortion minimization problem: in this setting, the centers correspond to our vector proxies and the surface we wish to partition corresponds with our mesh. Once our mesh has been partitioned through the flooding mechanism, we update our vector proxies by taking the new "average" in the associated region, and repeat these two steps until our partition converges, see Figure 6. It can be proven [CSAD04] that such an algorithm aims at minimizing our global distortion error E - the flooding stage minimizes E for a fixed set of vector proxies while the proxy fitting stage minimizes E for a fixed set of regions.

Possible Extensions The basic technique presented in this paper is simple to extend in various ways. First, repeated, subclustering can be performed to provide a versatile, hierarchical clustering of arbitrary fields. Second, the metric can

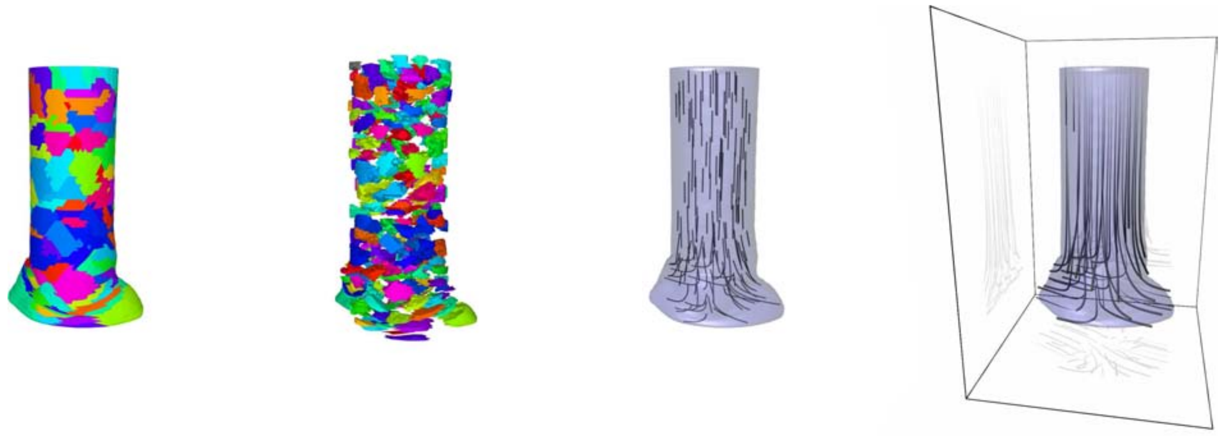

Figure 5: *Pipeline for visualizing a cylinder dataset during impact under non-slip conditions, 100k tetrahedra. Left: Variational segmentation into 200 regions. Middle-Left: Exploded view of the cluster volumes. Middle-Right: Streamlines initiated from the centroid of each cluster, terminated at the region boundary. Right: Thinning tubes drawn through the entire domain while adhering to a distance threshold. We use orthogonal shadows projected on the base planes for added 3D cues.*

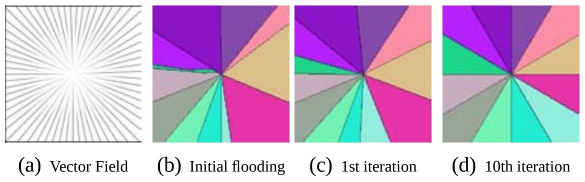

Figure 6: *Lloyd's algorithm is applied to quickly drive down the distortion error. Through iterative partitioning and proxy fitting, the clustering converges in just a few steps.*

be easily altered: one especially useful metric for visualization is a weighted combination between the gradient tensor metric and a spatial position metric. But more physically-driven metric choices can also be made.

3. Results and Discussion

We have tested our variational segmentation technique extensively on vector fields varying from analytic, coarse grid test cases to large, noisy flow fields with tens of thousands of sample points on irregular grids. In all cases we obtain very natural segmentations, especially when incorporating proxy teleportation [CSAD04] to eject proxies from sub-optimal local minima. Applying the variational mechanism to different metrics allows us to extract new and insightful perspectives from any flow field—Figures 5 and 8, 9 show this machinery in action.

The framework presented here gives us much space for further exploration and developments. A significant drawback to discuss is our present inability to extract a meaningful visualization from our segmentation. In other words, now that we have simplified the field into k regions, how can we present this data in a constructive manner that will generate a greater understanding of the underlying flow for scientific researchers?

The naïve approach of showing a hedgehog per region is both cluttered and unintuitive. Previous work on clustering has relied on slightly more advanced representations such as curved arrows, analogous to shaded streamlines with an icon at the front of the curve to represent flow directionality. But arrowheads specifically, are extremely strong visual elements, which garner focus away from the more subtly curved streamlines. This is a very undesirable side effect because the curve is where most of the information about the movement of the flow is presented. Conversely, streamlines alone (without an arrowhead) fail to indicate the direction of the flow, which is rather important when predicting particle movement, as well as to distinguish between singularities such as sources and sinks.

3.1. Visualization through Streamlines

As suggested by Figure 3(c), the first-order gradient tensor metric provides an elegant mechanism to home-in on singularities and active areas in the data, therefore serving as an excellent foundation for visualization purposes. One possible strategy for this (similar to [JL97]) is presented below:

Having obtained the representative clusters, we use these for the placement of streamlines. For each region in the partition, we initiate a streamline from the barycenter of the cluster and integrate backwards and forwards over the entire vector field using a standard Runge-Kutta scheme. To ensure an attractive distribution of streamlines for the visualization, a Euclidean distance threshold t is introduced. We trace the streamlines in ascending order of the associated cluster's volume, and if at any time this trace comes sufficiently close to anything that has previously been drawn (within distance t), the streamline trace is terminated and we proceed to the next seed.

3.2. Thinning Tubes

For an elegant display of the resultant streamlines, we introduce the use of thinning tubes—generalizing from [JL97] and [TB96]. Thinning tubes are tubes of finite volume,

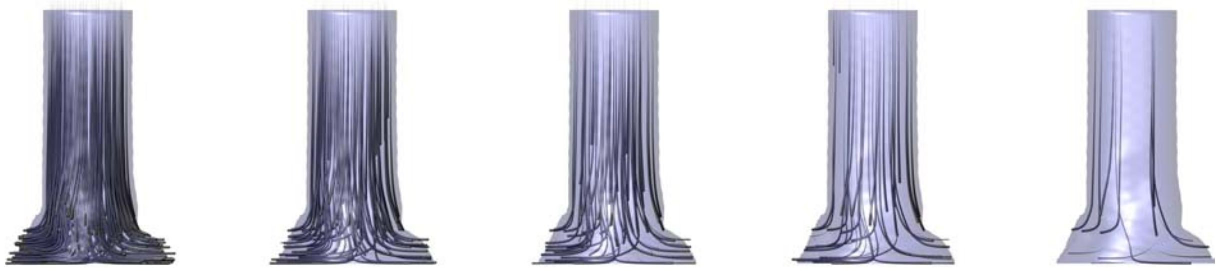

Figure 7: *The effects of an increasing streamline distance threshold.*

whereby the radius of the tube's circular cross section linearly increases in the direction of the vector field. The advantage of thinning tubes over regular streamlines or flow ribbons is that the varying girth indicates the direction of flow; and does so in an unobtrusive manner that doesn't detract visual focus from the subtle paths of the flow. Moreover, as 3D objects, thinning tubes render well under basic specularity supporting light models, thus increasing the 3D information—3D cues—presented to the researcher.

Given a streamline s of length l, for any point on s that has a distance d along the curve to the front of the streamline, we compute the radius $r(d)$ of the thinning tube's cross section at that point to be $r(d) = k * ((l-d)/l))$. The front of the thinning tube will therefore have a radius k, and in our implementation k is also directly correlated with the length of the streamline. This helps to achieve a visual balance that emphasizes longer streamlines.

Note that thinning tubes have the potential for many variations. We can further manipulate them to resemble artistic brushstrokes, or to supply additional information - in particular we could indicate the magnitude of the vector field by varying the girth of the tube, or varying its color as it integrates the field.

3.3. Conclusion and Future Work

Considering the generalized framework of Lloyd's algorithm and its ease of implementation, our method quickly paves the road for numerous enhancements. In particular, consider a vector field that is *not* static, but varies over time. Extending our notion of a vector proxy, a 4D approximation (3D + time) using a space-time metric can be constructed to effectively make the best of both spatial and temporal components: variational motion segmentation could reveal itself a powerful visualization tool.

4. Acknowledgments

We wish to thank Pierre Alliez for providing insight (and examples, as in Figure 3(a)) into new streamline techniques, Peter Schröder, for the generous provision of lab space and Thai food, Céline Loscos for support, Rudiger Westermann for the car dataset, and the Student-Faculty Program at Caltech that made this work possible.

References

[CL93] CABRAL B., LEEDOM L.: Imaging Vector Fields using Line Integral Convolution. In *ACM Trans. on Graphics* (1993), pp. 263–270. 2

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational Shape Approximation. *ACM Trans. on Graphics 23*, 3 (2004), 905–914. 3, 4, 5

[DW04] DU Q., WANG X.: Centroidal voronoi tessellation based algorithms for vector fields visualization and segmentation. In *Proceedings of the IEEE Visualization 2004 (VIS'04)* (2004), pp. 43–50. 2, 4

[GIS03] GORLA G., INTERRANTE V., SAPIRO G.: Texture synthesis for 3d shape representation. *IEEE Transactions on Visualization and Computer Graphics 9*, 4 (2003), 512–524. 2

[GL91] GLOBUS A. C. L., LASINSKI T.: A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings Visualization '91* (91), pp. 33–40. 2

[GPR*00] GARCKE H., PREUSZER T., RUMPF M., TELEA A., WEIKARD U., VAN WIJK J. J.: A Phase Field Model for Continuous Clustering on Vector Fields. *IEEE TVCG 6*, 2 (2000), 139–149. 2, 3

[GPR*04] GRIEBEL M., PREUSSER T., RUMPF M., SCHWEITZER M. A., TELEA A.: Flow field clustering via algebraic multigrid. In *Proceedings of the IEEE Visualization 2004 (VIS'04)* (2004), pp. 35–42. 2

[JL97] JOBARD B., LEFER W.: Creating evenly-spaced streamlines of arbitrary density. In *8th Eurographics workshop on Visualization in Scientific Computing* (1997), pp. 453–460. 5

[LAK*98] LAIDLAW D. H., AHRENS E. T., KREMERS D., AVALOS M. J., JACOBS R. E., READHEAD C.: Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings*

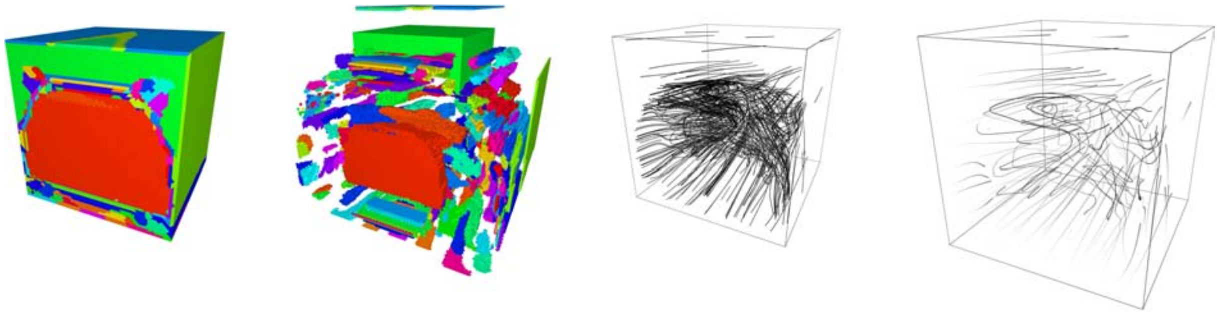

Figure 8: *Visualizing a flow field left in the wake of a moving automobile, 1.25 million tetrahedra. We cluster the dataset into 200 regions (Left; exploded view Middle-Left), initiate streamlines from the cluster centroids (Middle-Right) and ultimately apply thinning tubes and distance threshold for the final result (Right). Notice how a cross section of the station wagon (Left, red cluster) is preserved in the gradient tensor partition, and how turbulent flow is densely clustered leaving the area of sparse activity relatively untouched.*

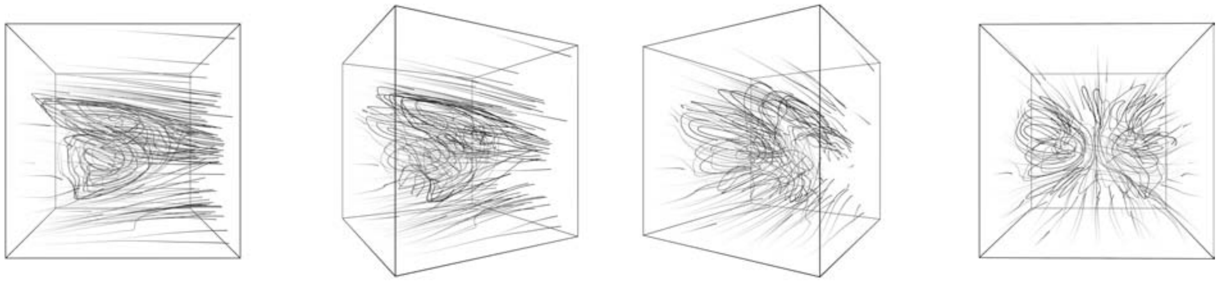

Figure 9: *Perspectives from the resulting visualization of the vehicle flow field in Figure 8, taken 30° apart.*

of the conference on Visualization '98 (1998), pp. 127–134. 2

[LKD*01] LAIDLAW D. H., KIRBY M., DAVIDSON J. S., MILLER T., DASILVA M., WARREN W., TARR M.: Quantitative comparative evaluation of 2D vector field visualization methods. In *Proceedings of IEEE Visualization 2001* (October 2001), IEEE, pp. 143–150. 2

[MTHG03] MATTAUSCH O., THEUL T., HAUSER H., GRÖLLER E.: Strategies for interactive exploration of 3d flow using evenly-spaced illuminated streamlines. In *Proceedings of the 19th Spring Conference on Computer graphics* (2003), ACM Press, pp. 213–222. 2

[PP00] POLTHIER K., PREUSS E.: Variational Approach to Vector Field Decomposition. In *Eurographics Workshop on Visualization (VisSym)* (2000). 4

[PVH*02] POST F. H., VROLIJKA B., HAUSERB H., LARAMEEB R. S., DOLEISCHB H.: Feature Extraction and Visualisation of Flow Fields. In *Eurographics '02* (2002). 2

[RSHTE99] REZK-SALAMA C., HASTREITER P., TEITZEL C., ERTL T.: Interactive exploration of volume line integral convolution based on 3d-texture mapping. In *Proceedings of the conference on Visualization '99* (1999), IEEE Computer Society Press, pp. 233–240. 2

[SM02] SCHUSSMAN G. L., MA K.-L.: Scalable self-orienting surfaces: A compact, texture-enhanced representation for interactive visualization of 3d vector fields. In *Pacific Conference on Computer Graphics and Applications* (2002), pp. 356–365. 2

[TB96] TURK G., BANKS D.: Image-Guided Streamline Placement. In *ACM Trans. on Graphics* (1996), pp. 453–460. 2, 5

[TLHD03] TONG Y., LOMBEYDA S., HIRANI A., DESBRUN M.: Discrete Multiscale Vector Field Decomposition. *ACM Trans. on Graphics 22*, 3 (2003), 445–452. 4

[TV99] TELEA A. C., VAN WIJK J. J.: Simplified representation of vector fields. *IEEE Proc. Visualization '99* (1999), 35–42. 2, 3