Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002)
D. Ebert, P. Brunet, I. Navazo (Editors)

A Novel Approach To Vortex Core Region Detection

Ming Jiang† The Ohio State University
Raghu Machiraju† The Ohio State University
David Thompson‡ Mississippi State University

Abstract

In this paper we present a simple and efficient vortex core region detection algorithm based on ideas derived from combinatorial topology. These ideas originated from Sperner's lemma, which by itself is of little value to detecting vortex cores. However, we take these ideas from the lemma and apply them in a point-based fashion to detecting vortex core regions. The resulting algorithms for both 2D and 3D are quite simple and very efficient compared to existing ones. We applied our algorithms to both numerically simulated and procedurally generated datasets to illustrate the efficacy of our approach.

1. Introduction

Large-scale computational fluid dynamics simulations of physical phenomena produce data of unprecedented size (terabyte and petabyte range). Unfortunately, development of appropriate data management and visualization techniques has not kept pace with the growth in size and complexity of such datasets. One paradigm of large-scale visualization is to browse regions containing significant features of the dataset while accessing only the data needed to reconstruct these regions. The cornerstone of an approach of this type is a representational scheme that facilitates ranked access to macroscopic features in the dataset [12]. In this approach, a feature-detection algorithm is used to identify and rank contextually significant features directly in the wavelet domain. The algorithm requires a scalar field that indicates the location of features of interest.

For this approach to be successful, it is necessary for the algorithm to be able to automatically detect and identify regions containing application-specific features of interest. In [12], velocity data from the equatorial region of the Pacific Ocean [25] was employed as the test case. The primary features of interest in that data set were swirling regions. These swirling regions were identified using a pointwise vortex detection algorithm based on the eigenvalues and eigenvectors of the velocity gradient tensor [3]. The work reported here grew out of our efforts to improve the feature detection algorithm for swirling regions.

† {jiang, raghu}@cis.ohio-state.edu
‡ dst@erc.msstate.edu

Detecting vortices in complex flow fields is a problem of interest in both research and practical applications. Complicating the issue is the fact that there is no formal definition of a vortex. Everyone is familiar with images of swirling flows such as tornadoes, water flowing down a sink, etc. Through these images we have the notion that a vortex entails a bulk swirling motion of the fluid around a central region.

In this paper, we develop a simple and efficient method for detecting the central core region of a vortex. The method is based on an idea derived from a lemma in combinatorial topology: Sperner's lemma, which states that one can deduce the properties of a triangulation based solely on the information given at the boundary vertices. Analogously, our approach deduces the behavior of a vector field based on the information provided by the boundary vectors. In particular, we take a local approach to vortex core detection that involves evaluating whether or not each grid cell belongs the a core region by examining its neighboring vectors. Around core regions, velocity vectors exhibit certain flow patterns that are unique to vortices, and it is precisely these patterns that our algorithms search for in a computational grid. Checking for these patterns is simple and efficient, and as our results clearly show, it is quite effective.

Our paper is structured as follows. We first provide a brief review of existing vortex detection algorithms and discuss some of the issues involved in vortex definition. We next provide some mathematical preliminaries that our algorithm is based upon. Then, we provide results that demonstrate the efficacy of our approach. Finally, we draw conclusions as to the relative merits of our vortex core region detection algorithm.

2. Previous Work

In this section, we briefly review several vortex detection algorithms in the literature. Though these reviews are not meant to be exhaustive, they provide a fairly good overview of the state of the art in vortex detection. It should be noted that each of these algorithms may generate false positives and miss otherwise obvious vortices.

The first group of methods is based on isosurfaces of a scalar field. Levy et al. [10] developed a method on the assumption that a vortex core is located in a region where the normalized helicity $\frac{\mathbf{V}\cdot(\nabla\mathbf{V})}{|\mathbf{V}||\nabla\mathbf{V}|}$ approaches ± 1. In Berdahl and Thompson's [3] method, the assumption is that two of the eigenvalues of the velocity gradient tensor are a complex conjugate pair in regions of swirling flow. A parameter termed the "swirl" is defined at each point in the domain using the magnitude of the imaginary part of the conjugate pair and the velocity in the plane perpendicular to the eigenvector real eigenvalue. According to [3], the swirl is nonzero in regions containing vortices and attains a local maximum in the core region.

Jeong and Hussian [8] defined a vortex based on the symmetric deformation tensor S and the antisymmetric spin tensor Ω. According to [8], if the second largest eigenvalue of $S^2+\Omega^2$ is negative at a point, that point is contained within a vortex. Additionally, if the second invariant of the velocity gradient tensor $\frac{1}{2}\left(|\Omega|^2-|S|^2\right)$ is positive at a point, the point is contained within a vortex. The main disadvantage with these methods is their difficulty in automatically distinguishing the individual vortices.

The second group of methods is based on the extraction of vortex core lines. Banks and Singer [1, 2] developed a predictor-corrector algorithm based on the assumptions that the vortex core is a vorticity line (a streamline in the vorticity field) and that pressure should be a minimum in the core. Sujudi and Haimes [23] described a line-based method that extracts the vortex core by locating points that satisfy the following two conditions: 1) the velocity gradient tensor has complex eigenvalues and 2) the velocity in the plane perpendicular to the real eigenvector is zero. By connecting these points, a line segment representing the vortex core is constructed, though it is not always possible to form a contiguous line. To address this problem, Haimes and Kenwright [6] recast the algorithm to be face-based rather than cell-based.

Roth and Peikert [18, 19, 13] proposed a different approach for detecting core lines using the parallel vector operator. Rather than performing an eigen-analysis on the velocity gradient tensor, their algorithm detects for parallel alignment of the velocity vector with the acceleration vector. Their approach was especially especially designed for turbomachinery datasets, which often contain weakly rotating vortices with nonnegligible curvature. Whereas Sujudi and Haimes' [23] method has difficulty with curved vortices, theirs perform a correction for the curvature by taking second-order derivatives into account. [19] The main disadvantages with these methods are their computational complexity and the fact that more than one core line is often produced within the same core region.

The third group of methods is based on the geometric properties of streamlines. Portela [15] developed a collection of mathematically rigorous definitions for a vortex, using set theory and differential geometry. Essentially, his definitions are based on the idea that a vortex is comprised of a central core region surrounded by swirling streamlines. His 2D method detects vortices by verifying whether or not the winding angle of streamlines around a grid point is a scalar multiple of 2π. Sadarjoen et al. [21, 20] proposed a simplification to the 2D winding-angle method, by using the summation of signed angles along a streamline instead. The main disadvantage with these methods is that they lack a viable 3D counterpart to their 2D approach – winding angles are only meaningful in 2D.

3. Vortex Definition

There have been several vortex definitions proposed by the fluid dynamics community, though none of them are completely satisfactory. The difficulty lies in the generality of such definitions. One of the first definitions was proposed by Lugt [11]:

> *A vortex is the rotating motion of a multitude of material particles around a common center.*

The problem with this definition is that it is vague and does not lend itself to motivating any practical algorithms. In light of this, Robinson [16] attempted to provide a more concrete definition of a vortex:

> *A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex core.*

The primary shortcoming of this definition is that it is self referential. Additionally, no one has been able to utilize it to develop an effective algorithm [2]. In general, it is hard to detect the correct reference frames for all types of flows.

Portela [15] developed a collection of mathematically rigorous definitions for a vortex, using set theory and differential geometry. Though his 2D definitions are sound, his 3D definitions resemble that of Robinson's [16], and he only provided a viable 2D algorithm based on his definitions. The intuition behind his definitions, however, are quite insightful and general. Essentially, a vortex is comprised on a central core region surrounded by swirling streamlines. There are two integral aspects behind this intuition: the central core region and the swirling streamlines surrounding it. Both aspects are of equal importance and must be adequately addressed by any

algorithm endeavoring completeness. In this paper, we address the former aspect: detecting the central core region of a vortex.

4. Sperner's Lemma

Our simple and efficient algorithm for detecting the central core region of a vortex is based on Sperner's lemma. The lemma is a result from combinatorial topology. It's fame originated from having provided a simple and elegant proof for Brouwer's Fixed Point theorem. [5, 7, 17]

Theorem 4.1 *(Brouwer's Fixed Point Theorem)*
Every continuous mapping $f : \sigma \rightarrow \sigma$ where σ is a $p-$simplex has a fixed point.

In other words, Brouwer's theorem says that if we were to stir a cup of coffee, then after the stirring there must be some point in the coffee which is in the exact same position that it was in before the stirring. Moreover, if we try to stir that point out of its position, another point would be inadvertently stirred back into its original position.

Lemma 4.2 *(Sperner's Lemma)*
Every properly labeled subdivision of a simplex σ has an odd number of distinguished simplices.

The lemma says the following. Given a convex set in n-dimensions, triangulate it into subtriangles and assign to each vertex of the subtriangles a label from $1, 2, ..., n+1$. If the initial vertices of the convex set is fully labeled, then there exist an odd number of fully labeled subtriangles inside the convex set. A subtriangle is fully labeled if it receives all $n+1$ labels. [17]

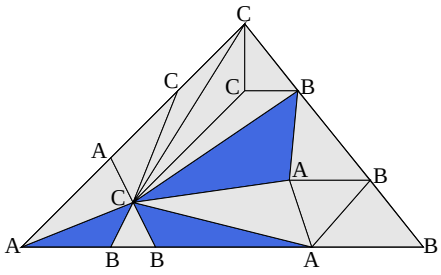

Figure 1: *Sperner labeling and simplicial subdivision of the original fully labeled triangle ABC result in three fully labeled subtriangles, colored dark blue*

Figure 1 is an example of Sperner's lemma in 2D which illustrates the simple idea of Sperner labeling. Given a 2D triangle, we can assign to each vertex of the triangle a label: $\{A, B, C\}$. If we begin with a fully labeled triangle, meaning each vertex of the triangle is labeled with a unique label, then Sperner's lemma guarantees us that any subdivison of the triangle would result in a fully labeled subtriangle, provided that Sperner labeling is used. In Figure 1, the original fully labeled triangle *ABC* is arbitrarily subdivided, resulting in three fully labeled subtriangles that are colored dark blue.

Sperner labeling is a very simple procedure: any vertex along the edge of a fully labeled triangle can only choose from the labels at the two vertices of that edge. So new vertices along an edge, introduced after a subdivision, can only choose from two labels. In Figure 1, the two vertices along the left edge of the outer triangle can only be labeled with either *A* or *C*. Similarly, vertices along the right edge can only be labeled with either *C* or *B*, and along the bottom edge, *A* or *B*. There are no restrictions for vertices introduced inside a fully labeled triangle. [5, 7]

Clearly, Sperner's lemma can be applied in any dimension. For instance, given a fully labeled tetrahedron, there would be at least one fully labeled subtetrahedron after a subdivision. [17]

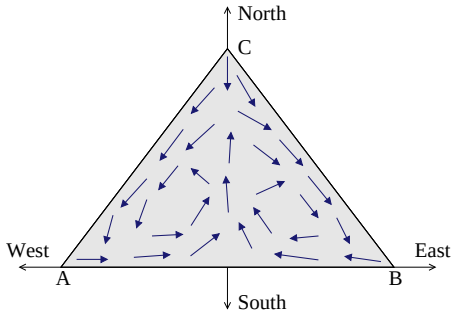

Figure 2: *A 2D vector field that satisfies Sperner's lemma. Note the two swirling centers on each side of the triangular domain and the switching saddle region near the top*

4.1. Labeling Duality

Sperner's lemma can be applied not only to triangulations but also to vector fields as well. Henle [7] pointed out the duality between labeling a triangulation and labeling a vector field. Essentially, this duality leads to the correspondence between fully labeled subtriangles and the fixed points, (i.e., critical points), of a vector field. From the perspective of critical point theory, Sperner's lemma guarantees the existence of at least one critical point under the condition of a fully labeled vector field domain.

Figure 2 is an example of a vector field that satisfies Sperner's lemma. (The domain is triangular purely for the sake of illustrating the similarities between triangulation and vector field.) By labeling a vector according to the direction range in which it point, we can label the vector field in the same fashion as we label a triangulation. This particular example satisfies Sperner's lemma because all the vectors point within the domain, including the ones at the three vertices. Therefore, since the three vertex vectors must necessarily point in different directions, then three unique labels

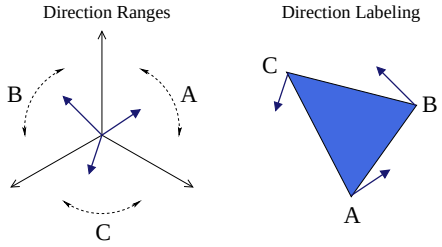

Figure 3: *The three equally spaced direction ranges correspond to direction labeling of the triangular cell*

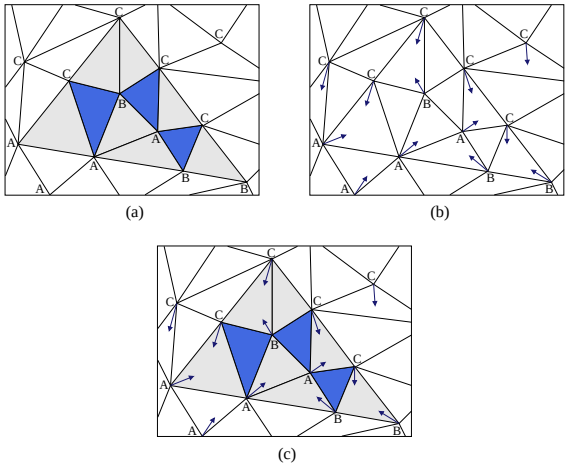

Figure 4: *(a) Sperner labeling a triangulation, (b) direction labeling a vector field defined on an unstructured grid, and (c) superimposing the two to show their labeling duality*

$\{A,B,C\}$ can be assigned to them, forming the fully labeled "triangular" vector field.

As alluded to above, direction labeling corresponds to assigning labels to vectors according to the direction ranges in which they point. In this case, we have three equally spaced direction ranges, as illustrated in Figure 3, each with a corresponding label chosen from $\{A,B,C\}$. A fully labeled triangular cell, in this case, corresponds to a property which we call direction-spanning. Direction-spanning occurs when each vector at the vertex of a cell point in a unique direction range. The resemblance between the two labeling techniques is most striking when we think of the vector field as defined on an unstructured grid. Figure 4 illustrates the dual nature between the two labeling techniques.

4.2. Subdivision and Interpolation

The process of subdividing a fully labeled triangle and applying Sperner labeling to its subtriangles is the dual of interpolating a continuous vector field and direction labeling its interpolated vectors. Figure 5 illustrates this process.

Let Ψ_i denote the process of subdividing a fully labeled triangle and applying Sperner labeling to subtriangles after the i_{th} iteration. With the exception of being a triangle subdivision, Sperner's lemma does not place any restrictions on the kind of subdivisions allowed. Starting with a fully labeled triangle $A_iB_iC_i$, the triangle subdivision produces a fully labeled subtriangle $A_{i+1}B_{i+1}C_{i+1}$, whose vertices are labeled according to Sperner labeling.

In terms of a vector field, the process Ψ_i involves interpolating between vectors at A_i, B_i, and C_i. Since the vector field is assumed to be continuous, the interpolation can be carried out along the three edges of triangle $A_iB_iC_i$. Because interpolation along a straight edge is always linear, the direction range of the interpolated vector at A_{i+1} must be equal to either A_i's or C_i's. In this case, the direction range is that of A_i's. Similarly, the vector at B_{i+1} is interpolated from A_i and B_i, and at C_{i+1}, from B_i and C_i.

As Henle [7] pointed out, the duality between Sperner labeling and direction labeling relies on the subdivision process of the former and the continuity and compactness assumptions of the latter. Theoretically, the process of subdivision can be continued *ad infinitum*, until triangles of arbitrarily small size are reached. Then arbitrarily small fully labeled triangles would contain all three labels that are, in a topological sense, near to each other. On a vector field, this would mean that all three vectors, pointing in different directions, would be near to each other as well. By assuming that the vector field is compact, there exist a vector that would be near to all three vectors, which means it must point in all three directions. The only vector that can point in all three directions at once is the zero vector (i.e., a critical point).

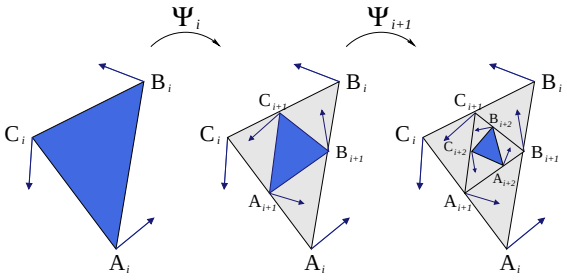

Figure 5: *The duality between subdivision with Sperner labeling and interpolation with direction labeling*

5. Algorithm

The novelty of our approach lies in its sheer simplicity. Our goal was to design an effective vortex core region detection algorithm that is both simple and efficient. What makes our approach novel are the concepts from Sperner's lemma that

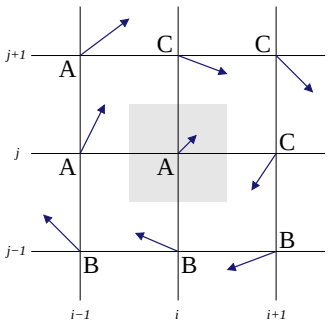

Figure 6: *2D vortex core region detection. Grid point* (i, j) *is detected because its immediate neighbors satisfy the direction-spanning property*

can lend themselves conveniently to designing an effective algorithm. (Note that this does not mean we use or apply Sperner's lemma, in any form, in our algorithm.)

We designed a local, point-based algorithm which, unlike the scalar methods, can extract the vortex core regions individually, unlike the line methods, is computationally inexpensive, and unlike the geometry methods, has a viable 3D algorithm. The algorithm was designed for for both structured and unstructured grids. For each grid point, our algorithm examines the velocity vectors of its immediate neighbors to see if they satisfy the direction-spanning property. If so, then that grid point is extracted as part of a vortex core region. The extraction procedure proceeds like a region growth algorithm, where all vertex-, edge-, and face-connected neighbors are recursively visited. Note that our algorithm is not restricted to grid points alone, using the centers of grid cells or any other interpolated grid positions would also suffice. However, these other grid positions come at the extra cost of interpolation without providing any advantages to the algorithm. A similar performance enhancement was noted in [6].

The reasoning behind this simple approach is straightforward and is based on the concepts from Sperner's lemma. In order for the immediate neighbors of a grid point to satisfy the direction-spanning property, it must be the case that each direction range has a velocity vector pointing in it. This indicates that the region delimited by the immediate neighbors contains either switching flows or swirling flows, since other laminar flows generally do not exhibit this property in a local neighborhood. Thus implying there is a high probability the grid point belongs to a vortex core region.

5.1. 2D Algorithm

The 2D algorithm is quite simple and efficient. It involves two passes over the grid. The first pass applies direction labeling to all the velocity vectors, and the second pass checks for the direction spanning property among the immediate neighbors of the grid points. The size of the neighborhood is not restricted by the algorithm; we chose immediate neighbors for reasons of efficiency. The algorithm can accommodate larger neighborhoods at an increased cost during the second pass, but we did not find that to be necessary for all the example datasets that we've tested. This 2D algorithm is illustrated in Figure 6 for a structured grid. The gray cell surrounding grid point (i, j) corresponds to the disjoint region delimited by (i, j)'s immediate neighbors. From examining the vectors and the directions in which they point, it is not too difficult to discern where the core region lies: near grid point (i, j).

Since the direction-spanning property can imply either swirling flows or switching flows, topological cleanup using eigen-analysis is performed as a post-processing step to eliminate the switching saddle regions from the set of detected vortex core regions. As it has been noted in [14, 3, 23], swirling flows imply conjugate pair eigenvalues in the velocity gradient tensor, so the topological cleanup eliminates detected grid points that do not meet this condition.

The computational complexity is linear. Each pass at most examines the local neighborhood of a grid point. The first pass, which the velocity vectors are direction labeled, requires only checking the signs of the x and y velocity components of each grid point. The second pass, in which the algorithm checks for the direction-spanning property, requires examining the direction label of the the immediate neighbors of each grid point. Direction-spanning can be implemented quite efficiently by using bits to represent the each direction range. The direction-spanning property can be represented using bit masks, which reduces the problem of checking for direction-spanning to comparing against bit masks.

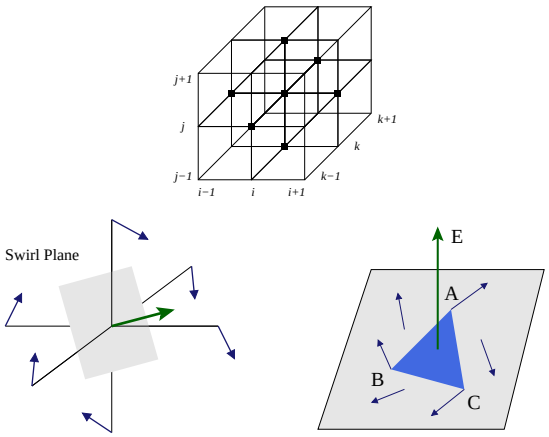

Figure 7: *3D vortex core region detection. Grid point* (i, j, k) *is detected because its immediate neighbors satisfy the direction-spanning property on the swirl plane*

5.2. 3D Algorithm

3D vortex core regions are much more difficult to detect than their 2D counterparts. The difficulties lie in the fact that 3D core regions are, in general, cylindrical shapes that can bend or twist in various different ways. Detecting such a region requires computing the core direction, which may or may not correspond to the actual direction of the vortex core region. Our approach involves computing the core direction and applying our 2D algorithm to the neighboring velocity vectors projected onto that plane. We refer to this plane as the swirl plane, because as [16, 15] pointed out, instantaneous streamlines projected onto this plane exhibit a swirling pattern.

Figure 7 illustrates the steps involved in our 3D algorithm on a structured grid. For each grid point (i, j, k), compute the core direction and the corresponding swirl plane. There are several existing methods to compute the core direction, all of which involves the velocity gradient tensor. They range from the least computationally intensive, the vorticity vector [2, 22], to the most computationally intensive, the real eigenvector [23, 6]. It should be noted that [4, 23] have shown the latter to be the correct core direction, and the others are only approximates to the core direction. Then project the neighboring velocity vectors onto the swirl plane and apply the 2D algorithm on the projected vectors. If the 2D algorithm detects the direction spanning property among the projected vectors, then there is a high probability that grid point (i, j, k) belongs to 3D vortex core region.

To complete the Sperner's lemma analogy for the 3D case, note that if our 2D algorithm detected the direction-spanning property among the projected vectors, then there exists a fully labeled triangle on the swirl plane whose three vertices are located at three of the projected neighboring positions. In Figure 7 this the the dark blue triangle labeled *ABC*. As we have alluded to earlier, in 3D we would expect a fully labeled tetrahedron [17], which contains four vertices with distinct labels each. The vector at the fourth vertex must point in a direction outside the swirl plane. In this case, the fourth direction can be none other than the core direction, denoted by the green arrow labeled *S*. Thus the fully labeled tetrahedron is precisely the tetrahedron labeled *ABCS*.

The computational complexity of our 3D algorithm is also linear. Unlike the 2D algorithm, there is only one pass over the entire dataset, since direction labeling has to be applied on the swirl plane at each grid point. Computing the core direction takes constant time. Projecting the neighboring velocity vectors onto the swirl plane and checking for direction-spanning among the projected vectors depend on the size of the neighborhood in which these operations are applied. Given that each grid point has a constant number of immediate neighbors, six for structured grids, the computation time at each grid point is constant.

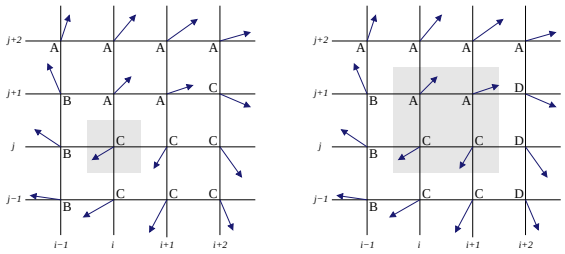

Figure 8: *The left grid was labeled with three direction ranges, and the right grid was labeled with four direction ranges. The four grid points that satisfied the direction-spanning property in the right grid more faithfully captured the swirling region than the single grid point that satisfied the direction-spanning property in the left grid*

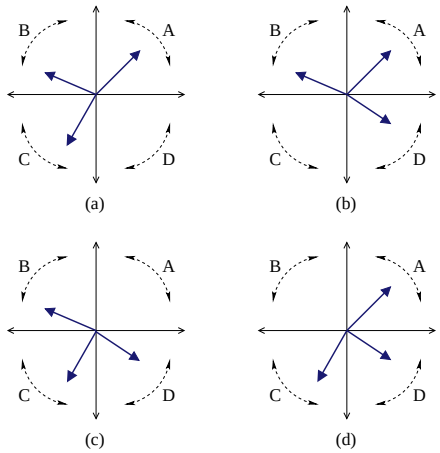

Figure 9: *The four possible types of direction-spanning with three vectors and four direction ranges: (a) ABC, (b) ABD, (c) BCD, and (d) ACD*

5.3. Direction Quantization

So far, both the 2D and 3D algorithms use three equally spaced direction ranges for direction labeling and checking the direction-spanning property. This is not always sufficient, as Figure 8 illustrates. The vector field in Figure 8 was designed so that the vortex core region would lie within the region delimited by the grid points $(i, j), (i+1, j), (i, j+1), (i+1, j+1)$. This is fairly obvious from examining the velocity vectors and the directions in which they point.

Figure 8(left) was labeled with three direction ranges. Unfortunately, only grid point (i, j) was detected, since its immediate neighbors satisfy the direction-spanning property. This is a side effect from how the direction ranges are quantized. Direction quantization refers to the number of possible direction ranges in which a vector can point. This problem

can be alleviated by introducing a finer quantization, (e.g., from three direction ranges to four). Figure 8(right) was labeled with four direction ranges. The four grid points that satisfy the direction-spanning property more faithfully captured the swirling region than the single grid point in Figure 8(left).

Figure 9 illustrates the four possible types of direction-spanning with three vectors and four direction ranges. In general, for n direction ranges, there are $\binom{n}{k}$ possible types of direction-spanning, where k is the number of vectors: $0 \leq k \leq n$. The quantization level can be set as high as desired, though the usefulness of such high level of quantization becomes suspect, not to mention the exponential increase in the search complexity when checking for direction-spanning property. For all the examples that we've tested, a level four quantization was sufficient.

Besides the efficiency tradeoff, increasing the level of quantization also affects the sensitivity of the 3D algorithm. What makes our 3D algorithm unique is its insensitivity to core direction variations, meaning approximate core directions can be just as effective as exact core directions. The coarser the level of quantization, the less sensitive our 3D algorithm is to core direction variations, thus we can use the inexpensive vorticity vector, rather than the expensive real eigenvector, for core direction. The reason is decreasing quantization levels increases the width of the direction ranges, and it is precisely these wider direction ranges that afford us the extra room to be less sensitive to variations in the core direction.

6. Results and Discussion

We have tested our algorithms on both numerically simulated and procedurally generated datasets. In every case, the vortex core regions detected by our algorithms were either analytically correct or concurred with similar results from other research groups. In order to visualize the extracted vortex core regions, we rendered them using both isosurfaces and actual grid cells. Although isosurfaces can appear at times smoother than the actual grid cells, they also introduce surface fitting errors, whereas the actual grid cells can accurately represent the vortex core regions. We also traced streamlines near the detected core regions in order to present a visual proof of these vortex cores through their swirling streamlines.

6.1. 2D Results

We tested our 2D algorithm using various datasets. Figure 10 shows the results from a pair of Rankine vortices. The detected core regions are colored black and the swirling streamlines surrounding the core regions are colored white. Figure 10(left) shows the output without topological cleanup. Notice the switching saddle point inbetween the two vortices was also detected. Figure 10(right) shows the output after a simple topological cleanup, where the switching point was eliminated completely, leaving behind only the pair of vortices.

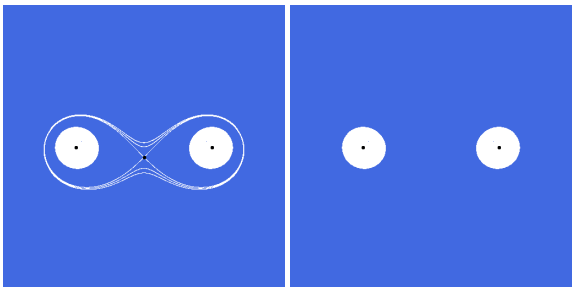

Figure 10: *2D Rankine vortices*

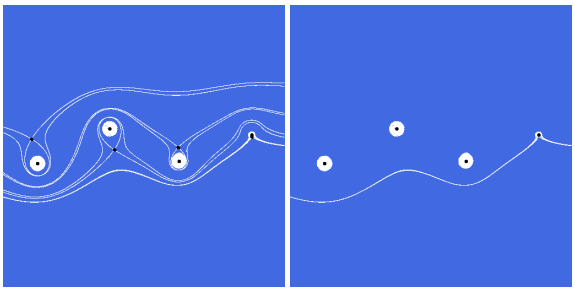

Figure 11: *Wake simulation using Rankine vortices*

Figure 11 shows the results from a wake simulation conducted using Rankine vortices. The difficulty with this dataset is that it contains a weak vortex at the right end of the flow field. Figure 11(left) shows the output without topological cleanup. Not surprisingly, the switching points were detected along with the Rankine vortices. Figure 11(right) shows the results after topological cleanup, where all the switching points were eliminated. The extra streamline that traced out from the weak vortex is due to the close spatial proximity between the weak vortex and a switching point.

Finally, Figure 12 shows the pre-cleanup and post-clean results from our algorithm on a LIC dataset from NASA Ames [24]. Figure 12(left) shows three vortices switched by the two saddle points, and Figure 12(right) shows the output after topological cleanup. Our results are very similiar to the LIC results presented in [24].

6.2. 3D Results

For the testing of our 3D algorithm, we used a variety of datasets, both numberically simulated and procedurally generated. The bent-helical vortex, first presented in [18] was a challenge for many existing algorithms because of its non-trivial curvature. Figure 13 shows the results from our 3D

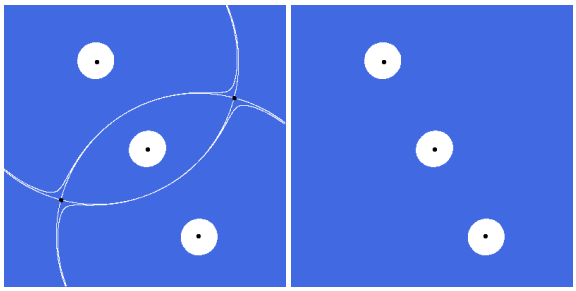

Figure 12: *LIC dataset from NASA Ames* [24]

algorithm for both slow and fast varying bent-helical vortices <see color plate>. The yellow region is the detected core region and the blue lines are the swirling streamlines. Our results match the ones presented in [18] [19].

We also tested our 3D algorithm on two numerically simulated datasets, both of which are standard benchmarks for demonstrating the efficiacy of a vortex detection or flow visualization algorithm. The first dataset is the blunt fin. Figure 14 shows the results of our algorithm. Although all three vortices match the vortex core lines detected in [19], they are not exactly correct. The two smaller vortices should have been detected as a single vortex. Like the line algorithms, this is one of the shortcomings of our algorithm: it doesn't always extract a contiguous vortex.

The other benchmark dataset is the delta wing. Figure 15 shows the results from our algorithm, which concur with the results presented in [9]. Notice how the two core regions starting from the nose of the delta wing suddenly disperses over the wings' middle section. This phenomenon is called a spiral vortex breakdown [9], and our algorithm was able to extracted them automatically. These positive results, both in 2D and 3D, demonstrate the efficacy of our algorithm.

7. Conclusion

We have presented a novel approach to vortex core region detection that is both simple and efficient. Our approach is based on an idea derived from a lemma in combinatorial topology: Sperner's lemma, which states that one can deduce the properties of a triangulation based solely on the information given at the boundary vertices. In a dual fashion, our approach deduces the behavior of a vector field based on the information provided by the boundary vectors. With respect to vortex core region detection, this means that we can tell with some certainty whether or not a grid point belongs to a vortex core region by examining its immediate neighbors for the direction-spanning property.

We have tested our algorithms on various datasets, both numerically simulated and procedurally generated. In every case, the vortex core regions detected by our algorithms were either analytically correct or concurred with similar results from other research groups. Of course our approach is not without its shortcomings, most noticeable of which are the false positives produced for complex flow datasets. One of our goals for future works is to develop feature verification algorithms that can verify the detected core regions. We would also like to develop feature tracking algorithms for time-dependent flow using the detected vortex core regions.

8. Acknowledgments

We would like to thank Prof. Tamal K. Dey from The Ohio State University for introducing us to the ideas of Sperner's lemma and how it relates to vector fields. This work is partially funded by the NSF under the Large Data and Scientific Software Visualization Program (ACI-9982344), and an NSF Early Career Award (ACI-9734483).

References

1. D. C. Banks and B. A. Singer. Vortex Tubes in Turbulent Flows: Identification, Representation and Reconstruction. In *Proc. Visualization '94*, pages 132–139, 1994.
2. D. C. Banks and B. A. Singer. A Predictor-corrector Technique for Visualizing Unsteady Flow. *IEEE Trans. on Visualization and Computer Graphics*, 1(2):151–163, 1995.
3. C. H. Berdahl and D. S. Thompson. Eduction of Swirling Structure Using the Velocity Gradient Tensor. *AIAA J.*, 31(1):97–103, January 1993.
4. M. S. Chong, A. E. Perry, and B. J. Cantwell. A General Classification of Three-Dimensional Flow Fields. *Phys. Fluids, A*, 2(5):765–777, 1990.
5. D. I. A. Cohen. On the Sperner Lemma. *J. Combinatorial Theory*, 2:585–587, 1967.
6. R. Haimes and D. Kenwright. On the Velocity Gradient Tensor and Fluid Feature Extraction. In *AIAA 14th Computational Fluid Dynamics Conference, Paper 99-3288*, June 1999.
7. M. Henle. *A Combinatorial Introduction to Topology*. Dover, 1979.
8. J. Jeong and F. Hussain. On the Identification of a Vortex. *J. Fluid Mechanics*, 285:69–94, 1995.
9. D. Kenwright and R. Haimes. Vortex Identification - Applications in Aerodynamics. In *Proc. Visualization '97*, pages 413–416, 1997.
10. Y. Levy, D. Degani, and A. Seginer. Graphical Visualization of Vortical Flows by Means of Helicity. *AIAA J.*, 28(8):1347–1352, August 1990.
11. H. Lugt. *Vortex Flow in Nature and Technology*. Wiley, 1972.

12. R. Machiraju, J. Fowler, D. S. Thompson, W. Schroeder, and B. K. Soni. EVITA - Efficient Visualization and Interrogation of Tera-scale Data. In *Data Mining for Scientific and Engineering Applications*, pages 257–279, 2001.

13. R. Peikert and M. Roth. The "Parallel Vectors" Operator - a Vector Field Visualization Primitive. In *Proc. Visualization '99*, pages 263–270, 1999.

14. A. E. Perry and M. S. Chong. A Description of Eddying Motions and Flow Patterns Using Critical Point Concepts. *Ann. Rev. Fluid Mechanics*, 19:125–155, 1987.

15. L. M. Portela. *Identification and Characterization of Vortices in the Turbulent Boundary Layer*. PhD thesis, Stanford University, 1997.

16. S. K. Robinson. Coherent Motions in the Turbulent Boundary Layer. *Ann. Rev. Fluid Mechanics*, 23:601–639, 1991.

17. K. H. Rosen, J. L. Michaels, J. L. Gross, J. W. Grossman, and D. R. Shier. *Handbook of Discrete and Combinatorial Mathematics*. CRC Press, 2000.

18. M. Roth and R. Peikert. Flow Visualization for Turbomachineray Design. In *Proc. Visualization '96*, pages 381–384, 1996.

19. M. Roth and R. Peikert. A Higher-order Method for Finding Vortex Core Lines. In *Proc. Visualization '98*, pages 143–150, 1998.

20. I. A. Sadarjoen. *Extraction and Visualization of Geometries in Fluid Flow Fields*. PhD thesis, Delft University of Technology, 1999.

21. I. A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, and H.-G. Pagendarm. Selective Visualization of Vortices in Hydrodynamic Flows. In *Proc. Visualization '98*, pages 419–422,558, 1998.

22. R. C. Strawn, D. Kenwright, and R. Haimes. Computer Visualization of Vortex Wake Systems. *AIAA J.*, 37(4):511–512, April 1999.

23. D. Sujudi and R. Haimes. Identification of Swirling Flow in 3-D Vector Fields. In *AIAA 12th Computational Fluid Dynamics Conference, Paper 95-1715*, June 1995.

24. V. Verma, D. Kao, and A. Pang. PLIC: Bridging the Gap Between Streamlines and LIC. In *Proc. Visualization '99*, pages 341–351, 1999.

25. A. J. Wallcraft. The Naval Layered Ocean Model Users Guide. Technical Report 35, Naval Oceanographic and Atmospheric Research Laboratory, December 1999.

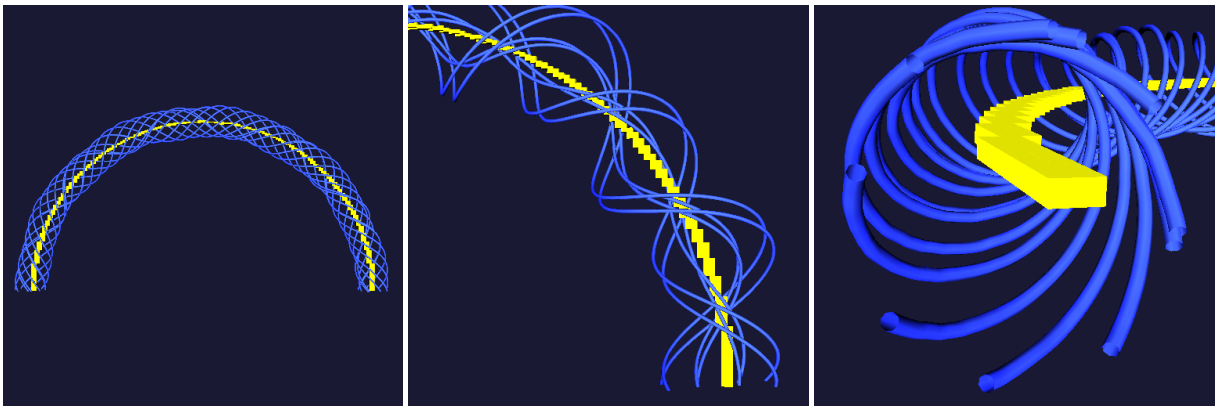

Figure 13: *Bent-helical vortex*

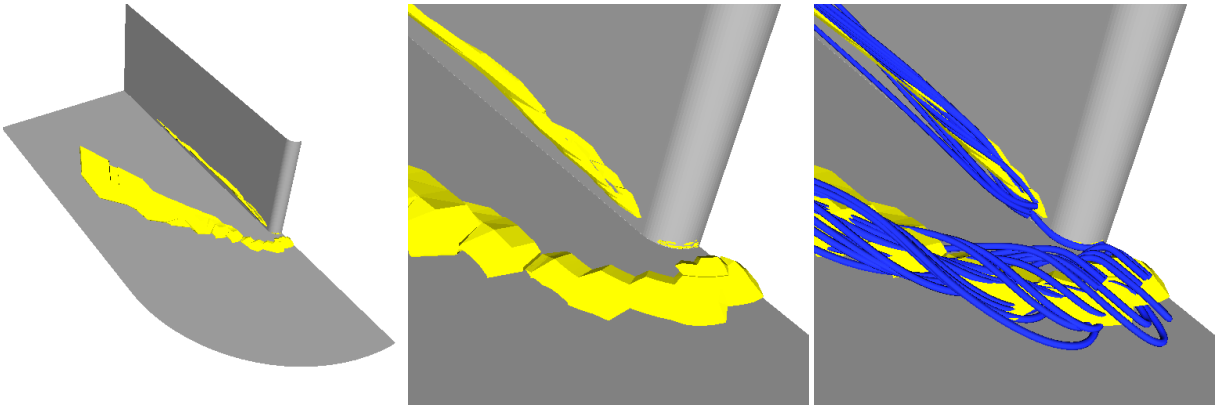

Figure 14: *Blunt fin vortices*

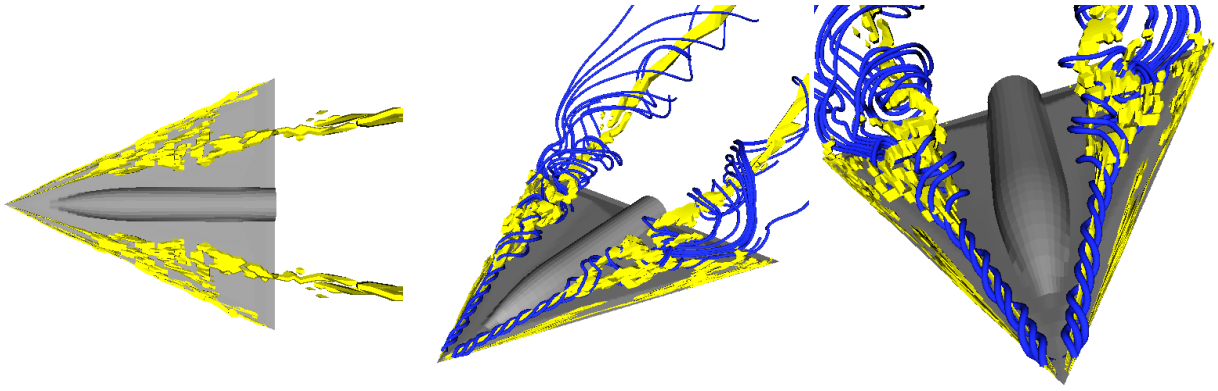

Figure 15: *Delta wing vortices*