Eurographics Conference on Visualization (EuroVis) 2013
B. Preim, P. Rheingans, and H. Theisel
(Guest Editors)

Volume 32 (2013), Number 3

Boosting Techniques for Physics-Based Vortex Detection

L. Zhang[1], Q. Deng[2], R. Machiraju[3], A. Rangarajan[2], D. Thompson[4], D. K. Walters[5], and H.-W. Shen[3]

[1] School of Computer Science and Technology, Shandong University, China
[2] Department of Computer Science and Engineering, The University of Florida, USA
[3] Department of Computer Science and Engineering, The Ohio State University, USA
[4] Department of Aerospace Engineering, Center for Advanced Vehicular Systems, Mississippi State University, USA
[5] Department of Mechanical Engineering, Center for Advanced Vehicular Systems, Mississippi State University, USA

Abstract

Robust automated vortex detection algorithms are needed to facilitate the exploration of large-scale turbulent fluid flow simulations. Unfortunately, robust non-local vortex detection algorithms are computationally intractable for such large data sets and local algorithms, while computationally tractable, lack robustness. We argue that the deficiencies inherent to the local definitions occur because of two fundamental issues: the lack of a rigorous definition of a vortex and the fact that a vortex is an intrinsically non-local phenomenon. As a first step toward addressing this problem, we demonstrate the use of machine learning techniques to enhance the robustness of local vortex detection algorithms. We motivate the presence of an expert-in-the-loop using empirical results based on machine learning techniques. We employ adaptive boosting (AdaBoost) to combine a suite of widely-used, local vortex detection algorithms, which we term weak classifiers, into a robust compound classifier. Fundamentally, the training phase of the algorithm, in which an expert manually labels small, spatially contiguous regions of the data, incorporates non-local information into the resulting compound classifier. We demonstrate the efficacy of our approach by applying the compound classifier to two computational fluid dynamics data sets. Our results demonstrate that the compound classifier has a reduced misclassification rate relative to the component classifiers.

Categories and Subject Descriptors (according to ACM CCS): I.2.6 [Artificial intelligence]: Learning—Parameter learning; I.4.6 [Artificial intelligence]: Segmentation—Edge and feature detection

1. Introduction

As computer power continues to increase, the complexity of simulations, both in terms of the physics modeled and the simulation size, also increases. Future exascale computing systems will generate increasingly larger simulation datasets [Rep09]. Even now, data is being produced at a rate that far exceeds the ability of application scientists to analyze it. What is lacking are the tools needed to facilitate data analysis and visualization of the resulting massive quantities of data.

One potential toolset, feature detection, is already an important strategy for domain experts who deal with terascale and petascale data. Fundamentally, feature detection operates as a data compression technique by reducing the amount of data that needs to be analyzed to a set of feature descriptors or a feature catalog. There are two distinct paradigms that can be employed to identify a feature [TMJ*02]: local and global (non-local). The local approach, or point classification, operates on a small neighborhood of the data and performs a binary classification as to whether a discrete point belongs to a feature (e.g., shocks in flow data). The collection of identified data points can then be aggregated to form the feature. In contrast, a global approach identifies a feature by an aggregate classification strategy and requires information from nonlocal regions of the dataset (e.g., streamlines in flow data). For certain feature types, the global approach can be more discriminating; however, this increased discrimination comes with an increased cost and, for exascale data, the resulting cost may be prohibitive. To reiterate, local, field-type methods are appropriate for use with exascale data because only local data is needed thus resulting in superior performance and a lower memory footprint.

We conceptualize the underlying problem in feature detection as a lack of *robustness* and attempt to address it via

the application of machine learning methodologies. Our hypothesis is that the performance of these local methods can be improved via the inclusion of domain expertise, i.e., an expert-in-the-loop. Our goal is to to produce a robust feature detection strategy by judiciously combining a selection of local feature detection algorithms into a single compound classifier using the method of adaptive boosting (AdaBoost) [FS97]. The resulting compound classifier remains a local field-type classifier, thereby making it appropriate for exascale data applications. Ideally, it combines the best of all local classifiers and converges asymptotically to the ideal classifier. The key aspect of our method is the use of an expert-in-the-loop to capture ground truth; an endeavor easier said than accomplished. Our approach is to exploit domain expertise to create training data for the boosting algorithm.

Methods from machine learning have not been widely adopted by the visualization community and it is our hypothesis that their utilization will suggest new directions of research in feature detection. Our paper is structured as follows. We first describe (Section 2) methods of vortex detection that rely on the underlying physics. We also describe how vortices are detected manually and explain the difficulties that beset automatic detection. In Section 3, we make a case for expert-in-the-loop and argue for the creation of a compound classifier and describe the AdaBoost algorithm adapted to detecting vortical features. Section 4 discuss results of detecting vortices in a specific data set (the tapered cylinder flow). We compare the performance of the compound classifier against that of other classifiers and show that it performs as well as the best realizable support vector machine, while being more computationally attractive. Finally, in Section 5 we provide closing summary and point to the future.

2. Vortices in Fluid Flow: Physics-Based Feature Detection

In many fluid dynamics applications, vortices are the features of interest. This is particularly true for computational fluid dynamics (CFD) simulations of turbulent flow, which produce copious quantities of data [WM09]. Unfortunately, there is no consensus on a formal, rigorous definition of vortex [CBA05, SBV*11]. In fact, as noted by Cucitore *et al.* [CQB99], the simpler techniques are based on quantities that are "intuitively related to a vortical motion." One highly intuitive description of a vortex is based on the notion of *swirling* fluid motion. Robinson [Rob91] describes a vortex in terms of its instantaneous streamlines as:

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

However appealing this description may be, it is self referential. That is, to find a vortex, you must first know where it is located, how it is oriented, and how fast and what direction it is moving. Nevertheless, this definition does illustrate the inherently global nature of a vortex.

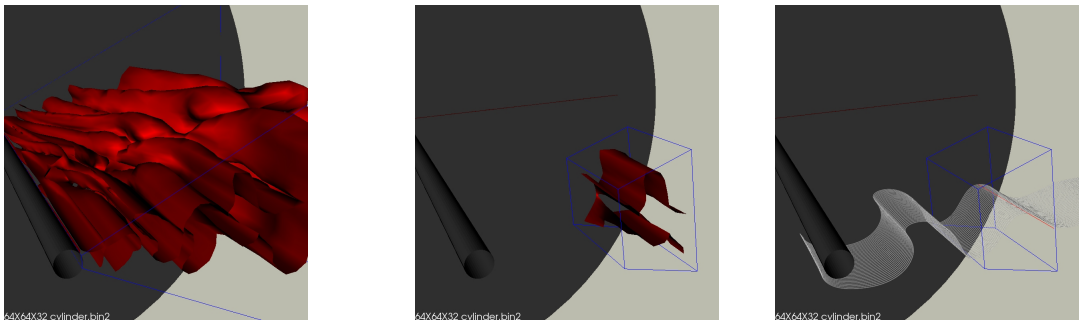
In spite of the lack of a formal definition, numerous vortex detection algorithms, both local and global in nature, have been proposed in the literature. Each of the various algorithms has an associated implicit vortex definition that forms the basis for the algorithm. Global methods are generally based on some notion of *coherence* of particle trajectories. Local algorithms are typically based on some measure, either direct or indirect, of local rotation in the fluid. In practice, local algorithms typically require thresholding of some sort to produce useful results [DD00, KM98, CBA05]. False positives, in which the presence of a vortex is indicated where none exists, are fairly common [HK99, JMT02b, CBA05, Kol07], as well as situations in which different detectors fail to agree on the region that constitutes given vortex [CBA05].

2.1. Manual Vortex Detection

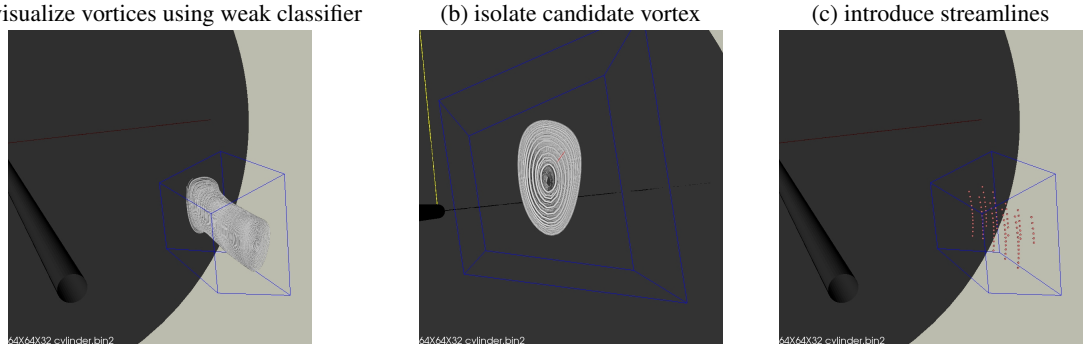
We now describe a process, illustrated in Figure 1, by which an expert can manually extract a vortex. This inherently global approach is based on the definition of a vortex given by Robinson [Rob91], i.e., we look for *coherence* exhibited by closed or spiraling streamlines in a reference frame moving with the vortex. Given a candidate region that contains a vortex, the expert *iteratively* adjusts seed locations for streamlines and the translational velocity of the reference frame to generate streamlines that exhibit coherence.

The first step is to isolate a region or block of the flow field in which a vortex can be easily located. In the example shown here, we employ an isosurface of λ_2 to identify a region of the domain containing a vortex. The next step is to introduce a seed line and iteratively adjust its position while, at the same time, imposing a uniform translational velocity until a set of closed or spiraling streamlines are obtained. It should be noted that, since a vortex may have a variable translational velocity along its axis. Therefore, a single velocity may not accurately capture the motion of the selected region. Here, we limit the length of the selected region parallel to the vortex axis in order to ensure the appropriateness of the assumption of a uniform translation velocity. In practice, this is how the vast majority of interactive visualization is done. The expert adjusts threshold values to produce an isosurface that "looks reasonable" or places seed points to produce streamlines that pass through a region in which a vortex is expected to reside.

Because our point-picking interface is restricted to selecting points in a plane, we employ a simple geometric primitive to select three-dimensional regions. We project the three-dimensional points onto the view plane of the screen and mark those points that are contained within an elliptical region inscribed in a rectangle. For this approach to be effective, the axis of the vortex must be more-or-less aligned



(a) visualize vortices using weak classifier (b) isolate candidate vortex (c) introduce streamlines



(d) iteratively adjust seed line and velocity of reference frame (e) align vortex axis perpendicular to viewplane and select points (f) export label set

Figure 1: Schematic of expert-user process for extracting vortices using streamlines in reference frame moving with the vortex. The key is the selection of the seed line orientation and the reference frame velocity. Unfortunately, this approach requires significant user input and is not appropriate in an automated context.

with the view plane normal. This places a restriction on the length of the vortex can be used as training data. The next step is to mark the points that are contained with the vortex region using selection tool with binary flag (0 – no, 1 – yes). We also have the capability to change the view and select or de-select individual nodes for fine tuning of the labels.

2.2. Automated Vortex Detection

Due to difficulties associated with the selection of appropriate seed points and a translational velocity, the streamline tracing described above is not useful in an automated vortex detection context. Several other global techniques have been reported in the literature. Cucitore *et al.* [CQB99] described a Galilean-invariant method based on the observation that two particles inside a vortical structure maintain the same relative separation even as they follow different trajectories. More recently, several researchers have exploited finite-time Lyapunov exponents (FTLE) [Hal01] to characterize Lagrangian coherent structures, again relying on the notion of coherence to define a flow structure. Examples include the work of Garth *et al.* [GGT07], who employ direct volume rendering to visualize the field, and Sadlo and Peikert [SP07], who utilize hierarchical mesh refinement to more efficiently compute ridges in the FTLE field. Among the most sophisticated vortex detection algorithms is the one described by Haller [Hal05] that provides an objective definition of a vortex based on the stability of fluid trajectories in unsteady, incompressible flows based on the M_Z criterion. In this context, objective means that the definition is independent of the reference frame, which can be rotating, e.g., such as turbomachinery applications. However, as noted earlier, these global techniques are computationally expensive and not a viable approach for routine use with large data.

Local techniques based on a identifying a local minimum in pressure [BS94] or a local maximum in vorticity [SKA99] along with techniques based on normalized helicity [LDS90], which is a measure of the alignment of the vorticity and velocity vectors, have been used with only limited success. Topology-based methods [BT93, SH95, RP98, PR99, JMT02a, RSVP02, WSTH07] seek to exploit the fact that there is a critical point in the velocity field at the vortex core in the plane containing the swirling motion. By their very nature, these local methods provide a description of a vortex in terms of its core line or core region [JMT02a]. However, the utility of these critical-point-based techniques is somewhat limited since they are also not Galilean invariant.

An important class of vortex detection methods is based

on the velocity gradient tensor J. It should be noted that several of the topology-based methods could also be classified in this manner; however, they also depend on the velocity field and are not Galilean invariant while methods that depend solely on the velocity gradient tensor are. In the discussion that follows, we make reference to the rate of strain tensor S and the rate of rotation tensor Ω, which are defined in terms of the velocity gradient tensor as:

$$S = \frac{J+J^T}{2}, \Omega = \frac{J-J^T}{2}$$

In some cases, e.g. two-dimensional steady flow, several of these methods reduce to the same approach. This is not the case, however, in more complex three-dimensional flows. Unfortunately, as reported in the literature [HK99, CQB99, DD00, JMT02b, CBA05], none of these vortex detection schemes is foolproof.

The Q-criterion [HWM88] is based on the observation that, in regions where $Q = \frac{\|\Omega\|^2 - \|S\|^2}{2} > 0$, rotation exceeds strain and, in conjunction with a pressure minimum, indicates the presence of a vortex. The Δ-criterion [CPC90] assumes that a vortex occurs in a region in which the eigenvalues of J include a complex conjugate pair. Here $\Delta = (\frac{R}{3})^3 + (\frac{detJ}{2})^2 > 0$, where $R = \frac{\Omega_{ij}\Omega_{ji} + S_{ij}S_{ji}}{2}$, indicates the presence of complex eigenvalues. However, relatively large regions of the flow can satisfy this criterion. As noted by Chakraborty *et al.* [CBA05], the $Q > 0$ criterion is more restrictive than $\Delta > 0$. A related approach developed by Zhou *et al.* [ZABK99] utilized the imaginary part of the complex conjugate pair of eigenvalues when $\Delta > 0$ to estimate the "swirling strength" of the vortex.

The λ_2-method [JH95] defines a vortex to be a connected region in which $\lambda_2 < 0$, where $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the eigenvalues of $S^2 + \Omega^2$. The key relationship is this approach, i.e., the equality of the Hessian of the pressure and the quantity $S^2 + \Omega^2$, is derived from the incompressible Navier-Stokes equations under the assumption that the unsteady straining is negligible. Although rigorously derived, the λ_2-method is based on the assumption that a vortex is located in a region in which a rotation-induced pressure minimum occurs. Cucitore *et al.* [CQB99] demonstrate that the λ_2-method is strongly related to the Q-criterion because it can be recast in terms of local straining and rotation. They also provide a derivation of the Hessian of the pressure that is appropriate for compressible flows.

Practical implementations of these methods typically include some type of thresholding [KM98]. Both Dubief and Delcayre [DD00] and Chakraborty *et al.* [CBA05] consider the problem of determining the thresholds for the different methods. Chakraborty *et al.* provide guidelines for determining equivalent thresholds based on dimensional consistency of the various terms appearing in expressions Δ, Q, and λ_2. However, they do not address the issue of what constitutes an effective threshold.

Another related vortex identification algorithm has been proposed by Graftieaux *et al.* [GMG01]. Let P be a fixed point in the measurement domain. A dimensionless scalar function Γ_1 at P is defined as

$$\Gamma_1(P) = \frac{1}{S}\int_{M \in S} \frac{(PM \wedge U_M)\cdot z}{\|PM\| \cdot \|U_m\|} dS = \frac{1}{S}\int_S sin(\theta_M) dS$$

where S is a two-dimensional plane containing P, M lies in the plane S and z is the unit-vector normal to the measurement plane. Further, θ_M represents the angle between the velocity vector U_M and the radius vector PM. It can be then shown that $|\Gamma_1|$is unity at the location of vortex center. Also proposed is a way to determine the vortex boundary through the use of another measure, Γ_2, which is essentially a locally computed function depending only on Ω and μ.

$$\Gamma_2(P) = \frac{1}{S}\int_{M \in S} \frac{[PM \wedge (U_M - \tilde{U}_P)]\cdot z}{\|PM\| \cdot \|U_M - \tilde{U}_P\|} dS$$

where $\tilde{U}_P = (1/S)\int_S U dS$. Further, the region with $|\Omega/\mu| \geq 1$ is identified as the vortical region where Ω is the rotation rate corresponding to the anti-symmetric part of the velocity gradient tensor J at point P and μ is the eigenvalue of the symmetrical part of this tensor.

2.3. The Case for the Expert-in-the-Loop

As discussed above, local vortex detection techniques are needed for application to exascale data. However, in the context of an automated vortex detection framework, the fundamental limitations associated with these techniques negatively impact their *robustness*. We now make two assertions regarding their lack of robustness:

- The ambiguities discussed above occur because the local detectors cannot capture the inherently global nature of a vortex [CQB99].
- In most cases, the false positives are manifestations of the fact that the underlying vortex definitions for the local detectors represent *necessary* rather than *sufficient* conditions for the existence of a vortex. The existence of false negatives can be attributed to improper threshold selection.

This set of circumstances presents a conundrum for vortex detection in exascale data – global techniques are prohibitively expensive and local techniques are unreliable.

One approach that has been employed in an attempt to mitigate this problem is to use combinations of local detection methods [BS95, TGK*04, SRE05, BMI*07] to exploit the favorable characteristics of each technique. For example, Burger *et al.* [BMI*07] express local binary feature detectors as fuzzy-sets that can be combined using linking and brushing in an interactive visual framework. Although the underlying strategy of this class of approaches is to combine the results of different detection algorithms, they do not rely on a rigorous framework that produces a systematic combination of these local classifiers.

The motivation for our work stems from the conviction that machine learning methods provide such a framework. Machine learning uses an expert-in-the-loop to produce a global vortex detector by augmenting the information provided by the local feature detectors. The underlying hypothesis is that an expert in fluid dynamics can—by labeling a limited subset of the data (as vortex or not-a-vortex)—help produce a reliable vortex detector. In our model, the expert-in-the-loop enhances the robustness that is lacking in the local vortex definitions *by providing non-local information*. The ability of the machine learning method to predict vortices in new regions crucially depends on both expert information and local feature detectors. The global vortex feature detector obtained via this process is also likely to be more robust than merely reliance on local detectors. To include an expert-in-the-loop, we need to define a set of expert labels as training data as illustrated in Figure 1. The resulting label set is then used as ground truth for the machine learning algorithm.

We have repeatedly emphasized the expected improved robustnesses of the compound feature detector produced by our approach. In an attempt to characterize robustness, we have elected to study the Type 1 and Type 2 errors of the local, physics-based feature detectors and the machine learning-driven global feature detector. Using an expert to label regions in addition to those provided for training allows us to *evaluate* the performance of the compound classifier. As we shall show, the compound feature detector exhibits improved performance in terms of *a lower error rate with a smaller number of false positives for an approximately equivalent number of true positives on two datasets*. Furthermore, it also illustrates the ability of the machine learning-driven method to generalize based on expert labels—another important feature of a robust feature detector.

3. Machine Learning for Vortex Detection

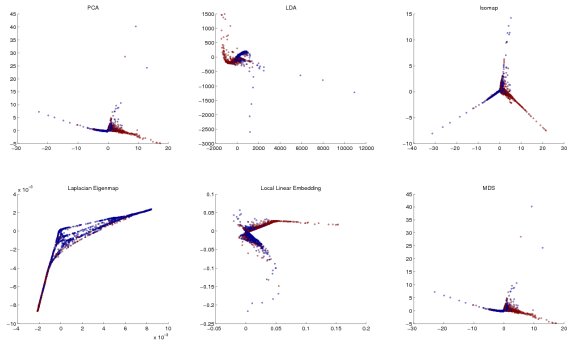
Before embarking on machine learning approaches to construct a compound or integrated feature detector, it is instructive to examine the scatter plot of the individual feature detectors with expert-in-the-loop labels added in order to get a clear picture of the intrinsic discrimination or classification problems ahead of us. A visualization of the intrinsic patterns of the data is of great help in this setting because the added expert labels clearly depict which subsets of patterns are difficult to separate. Since there are four individual feature detectors and additional measures (pressure, density, velocity, energy and cell edge), we resort to linear and nonlinear dimensionality reduction methods to display the data in a lower dimensional space.

3.1. Nonlinear Dimensionality Reduction - Making a Case for Combining Weak Clasifers

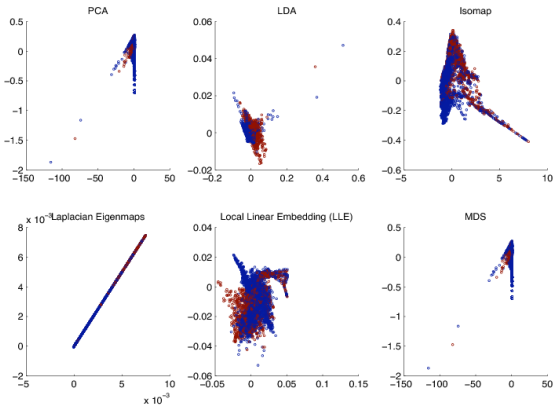
Principal component analysis (PCA) is the leading approach to global and linear dimensionality reduction. We add linear discriminant analysis (LDA) and multidimensional scaling (MDS) [TdL00] as well since they are widely used. In contrast to PCA and LDA are the nonlinear dimensionality reduction methods using manifold learning. The most popular manifold learning methods are local linear embedding (LLE) [SR00], ISOMAP [TdL00] and Laplacian Eigenmaps [BN03]. All three methods get going by first constructing a graph of nearest neighbors in feature space. Regardless of whether we only use the four physics-based feature detectors or augment them with measures such as energy, velocity etc., the approach remains the same. The methods construct a nearest neighbor graph using Euclidean distances between the feature vectors and then determine lower dimensional coordinates which are consistent with the nearest neighbor distances. Since the methods produce very different visualizations (as seen in Figure 2), a few comments on the need to use all methods is in order.

Dimensionality reduction, in our context, is being used to visualize the feature scatter plot. Over-reliance on a single dimensionality reduction method may lull us into complacency w.r.t. the discriminative ability of the feature set. However, if the lower dimensional scatter plots obtained from different methods provide a similar perspective, we would be somewhat justified in reaching a conclusion on the limitations of the features. On the other hand given the lack of a clear discriminating boundary between the true positives and negatives in any of the plots, one is lead to believe that a competent vortex detector can only be found through a careful and larger choice of new features.

We executed all six dimensionality reduction methods twice. First we used only the four physical feature detectors and next we added a five-dimensional measurement vector (three momentum components energy, and minimum edge length of a cell). The main reason for this seeming duplication is to check the overlay of the mesh points (vortex and non-vortex) with and without augmentation of other measures. These measures indicate the physical state and also provide a measure of the sample density pertaining to the sampling (edge length). From Figure 2, the overall picture emerging from dimensionality reduction is quite clear and pinpoints the intrinsic difficulties faced by any method (physics, machine learning or otherwise). While there is separation between a reasonable fraction of vortex and non-vortex nodes, there are dominant clusters where the two sets appear overlaid. This is highly suggestive that we have not yet discovered robust features capable of systematically separating vortices from non-vortices. Despite the fact that the expert-in-the-loop (presumably) has little difficulty labeling nodes as belonging to either of the two categories, the available feature detectors clearly fall somewhat short. Since we cannot reverse engineer (at the present time) the processes in the expert's brain, this work clearly shows that there is room for more, different (and discriminative) features in this domain. *A difference in performance between physics-based features and the expert indicates room for improve-*



(a) four physical detectors



(b) four detectors and five-dimensional measurement vector (three momentum components energy, and minimum edge length of a cell) at each mesh point of the flow volume.

Figure 2: Dimensionality reduction on four and nine features (including four physical feature detectors λ_2, Q, Γ_2 and Δ. Blue marks correct identification of a vortex by all four classifiers, while red marks false identification by any one of them. There is no clear demarcation between these two classes.

ment. If we cannot directly translate what the expert is doing into new physics-based features, there is room for machine learning to help improve the correlations between expert labels and the underlying physics. This work *therefore* shows that an integration of the different features (using supervised learning) should be beneficial since the added expert-in-the-loop information helps us *construct* a compound feature detector that outperforms the individual *physics-based feature detectors*. Note that the benefit of the expert information when applied separately to each feature detector is rather low (since there's just a single threshold to be tuned) whereas an integrated feature detector is likely to benefit more due to our ability to tune the weighted combination. Consequently, while the holy grail of the best feature remains in our future, *leveraging expert-physics correlations* to produce an integrated feature detector is likely to reap immediate dividends.

It should also be noted that the manual approach outlined in Section 2.1 is not appropriate for very large-scale data sets.

3.2. Vortex Classification via Machine Learning

A compound classifier as a composite detector: Once we possess a set of expert-in-the-loop labeled data with a subset of mesh points classified as either vortex or non-vortex, machine learning methods can be pressed into service. Instead of the previously existing situation of competing physics-based feature detectors (Q, λ_2, Δ and Γ), we can now contemplate their *integration* into a composite feature detector which has the potential to retain all of the advantages of the individual detectors. *This is the principal advantage of using machine learning methods in general since they afford the possibility of combining expert- and physics-based information to create a composite feature detector. From a larger perspective, our focus on boosting strategies in this work should be seen as a successful deployment of one machine learning approach. The potential for a larger machine learning study in this domain still remains.*

We use a boosting framework—widely used in recent years in machine learning applications—as it combines efficient scaling with adaptive tuning of feature weights and parameters. These aspects are expected to become more important as we scale to larger datasets and more physics-based feature detectors. The physics-based detectors are now considered to be weak classifiers in our boosting framework. Boosting leads to an integration which is facilitated by the additional information provided by the expert-in-the-loop who serves as a calibration target for the compound classifier. Furthermore, the expert labels provided allow for the composite feature detector (and each individual feature detector) to be evaluated in terms of their statistical performance—actual vortices detected (true positives) and non-vortex mesh points accurately labeled (true negatives). Without expert-in-the-loop labels, not only is there no information available to construct a composite feature detector (compound classifier), there is also no information regarding the *generalization* performance of the feature detectors (individual or composite).

A generalized composite detector: The ability of the composite feature detector to generalize well from a given set of expert-in-the-loop labeled data is one of the cornerstones of a successful machine learning approach. *That is, we seek a composite feature detector which leverages the correlations between expert labels and physics-based feature detectors to obtain improved performance on unseen data.* In the worst case, the integrated classifier merely regurgitates the training set labels while performing no better than chance on an unseen test set. This is the well known phenomenon of *overfitting*—the situation in which the many parameters of the integrated classifier have been too finely tuned to the expert-in-the-loop labels without really coming to grips with the regularities of the individual feature

detectors. At the other end of the spectrum, we expect the compound classifier to outperform the individual feature detectors. Based on these considerations, we require the composite feature detector to use the expert-in-the-loop labels to provide adequate performance (in terms of true positives and true negatives) especially when compared to the individual feature detectors. The overall statistical performance can be summarized using standard *sensitivity* and *specificity* measures (which codify the rates at which true vortices are detected and non-vortices labeled as such).

Boosting vs. SVM: Machine learning methods are similar to standard regression based learning approaches in that we seek a weighted combination of feature detectors ($\mathbf{w}^T\mathbf{f}$ where $\mathbf{w}$ is a weight vector and $\mathbf{f}$ the vectorized set of feature detectors) which maximizes performance relative to a fixed (training set) target. In a classification task, the targets are binary labels (vortex or non-vortex) and for this reason, machine learning methods typically eschew minimizing the ℓ_2 norm of the error (the mismatch between the predicted and true labels). Once the optimum function has been learned based on the training set, it is put to task to predict the labels of the incoming data. Regardless of whether boosting or support vector machines (SVMs)—to pick two recent popular methodologies—are used, training is accomplished by minimizing a suitable convex objective function. The objective function itself is chosen from foundational principles. For example, the SVM attempts to maximize the margin (the shortest distance in feature space) between the two classes whereas logistic regression replaces the ℓ_2 norm $(t-\mathbf{w}^T\mathbf{f})^2$ (where t is the expert-in-the-loop label) of standard regression with an objective tailored to binary labels. Boosting methods use a different objective function geared toward adaptation to incoming features using tuned thresholds that maximize performance.

When we seek to build a compound classifier using machine learning, the plethora of available methods can be bewildering. From our perspective (which takes scalability into account), boosting methods (such as AdaBoost [FS97]) afford the best tradeoff between simplicity and performance. While support vector machines have and continue to be highly successful, they do not scale as well as boosting methods. The SVM is a maximum margin classifier which uses convex quadratic programming to determine the optimum composite feature detector. Given the large mesh sizes in the simulation data, an approach (like the SVM) which scales quadratically in terms of training set sample size is usually not preferred. While more efficient primal domain SVM optimization algorithms are now available, they are difficult to implement and have not yet seen wide deployment. In contrast, AdaBoost which also minimizes a convex objective function is much more efficient in this setting, is easy to implement and has seen wide deployment.

3.3. AdaBoost – Creating a Compound Classifier

In this section, we describe the AdaBoost algorithm which results in a compound classifier. In this framework, each of the physics-based feature detectors (Q, λ_2, Δ and Γ) are reconceptualized as weak binary classifiers. The term "weak" denotes the inability of the physics-based feature detectors (which work point-wise) to benefit from incorporating information from a large subset of the data. The performance of the compound classifier on unseen data is expected to exceed or match at least any of the point-wise weak classifiers. AdaBoost [FS97], short for adaptive boosting, is a meta-algorithm that can be used in conjunction with other machine learning algorithms to improve their performance.

As illustrated in Algorithm 1, AdaBoost repeatedly uses different classifiers in an iteration sequence $t = 1,\ldots,T$. At each time step, a distribution of weights w_t is updated that essentially indicates the importance of chosen examples in the data towards the classification of a specific feature. In each iteration, the weights for each incorrectly classified example are increased (or alternatively, the weights for each correctly classified example are decreased), so that the new classifier focuses more on those examples. AdaBoost forms a conglomerate *hypothesis* (H_t), by adding up the learners trained in each step. In the t-th iteration, it reevaluates the penalty on the data samples according to the extent to which they are "wrongly" classified by the aggregated hypothesis H_{t-1} so far. The new hypothesis is selected to have a better accuracy on the "wrongly classified" samples.

The algorithm receives pairs of data samples and labels as inputs: $(x_1,y_1),\ldots,(x_n,y_n)$ where $y_i = \pm 1$ is the label of data. In our experiments, the feature vectors $\{x_i\}$ are encoded in $\mathbf{R}^4$, of which the dimensions are $Q,\lambda_2,\Delta,\gamma$ respectively. The version of AdaBoost used in this work aggregates weak learners by minimizing an exponential loss penalty function. Other versions of AdaBoost use logistic or L_p regression and we plan to examine the (minor) differences between these method in future work. Each physics-based feature detector is turned into a weak classifier via a one-level decision tree (also called decision stump) since more sophisticated decision trees don't offer significant advantages. In the t-th iteration, a physics-based feature detector $i(k) \in \{1,2,3,4\}$ is randomly chosen, based on which the weak learner h_t is expressed as

$$h_t(x) = l_t \mathrm{sign}(x^{i(k)} - b_t)$$

where b_t is the threshold value of the decision stump. The parameter $l_t = \pm 1$ controls the sign of the weak learner h_t.

3.4. Benefits and Challenges

The paucity of individual feature detectors combined by AdaBoost into an integrated composite is a concern at the present time. AdaBoost (and other machine learning methods) tend to perform better when there is a healthy diversity

Algorithm 1 AdaBoost algorithm

1. Obtain samples $(x_1,y_1),\ldots,(x_n,y_n)$ where $y_i=-1,1$ for non-vortex and vortex samples respectively
2. Initialize weights $w_{1,i}=\frac{1}{n}$ for $i=1,\ldots,n$
3. For $t=1,\ldots,T$:
   a. Get weak hypothesis $h_t:\{x_i\}\rightarrow\{-1,1\}$
   b. Select h_t that minimizes the weighted error $\varepsilon_t=\sum_i^n w_{t,i}\mathbf{1}_{h_t(x_i)\neq y_i}$
   c. Choose $\alpha_t=\frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$
   d. Update the weights $w_{t+1,i}=\frac{w_{t,i}\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ where Z_t normalizes $w_{t+1,\cdot}$
4. Final hypothesis: $H_T(x)=\text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

of individual feature detectors to be pooled. In our case, we have four feature detectors (Q, λ_2, Δ and Γ) which are all based on an understanding of the underlying physical mechanisms of vortex formation. Furthermore, these feature detectors are computed at each voxel (i) considering only the local neighborhood of each voxel, (ii) the relative location of the voxel and its neighborhood in the flow field (e.g., boundary layer, downstream), and (iii) ignoring any considerations of feature scale.

However, once we have established the effectiveness and relative improvement of a suitable machine learning-based composite feature detector, it is quite straightforward to include new feature detectors in future work. We would merely turn the new feature detectors into weak learners as explained above and then insert the new weak learners into AdaBoost. Especially since there is a paucity at the present time of machine learning-based feature integration, we feel that a straight-up integrated feature detector (using the most popular four physics-based features) is the necessary first step toward building more sophisticated integrators in the future. Finally, AdaBoost can be used in conjunction with other classifiers including decision trees where the weights are continuously changed.

4. Results

The goal of our effort is to demonstrate that robust classification of the domain into vortical and non-vortical regions is facilitated by the combination of several localized weak classifiers using the methods of Section **??**. We now describe the performance of the compound classifier relative to that of the weak physics-based classifiers for two different CFD data sets. The importance and role of collecting and using training data will also be emphasized.

4.1. Tapered Cylinder

We first apply our method to the tapered cylinder data set [DCJ91], which describes an unsteady, three-dimensional, incompressible, laminar, viscous flow around a cylinder that is perpendicular to the primary flow direction. Because of the tapering of the cylinder, the vortex shedding frequency varies along the length of the cylinder. Therefore, the vortices are inclined with respect to the axis of the cylinder. This data set was chosen, in part, because the orientation of the vortices would simplify the labeling process and demonstrate the efficacy of the compound classifier which was created by combining physics-based weak classifiers as described in Section **??**.

We now compare the performance of a compound classifier obtained by integrating four calibrated feature detectors, λ_2, Q, Δ, and Γ_2, relative to the individual detectors. First, our domain expert (co-author Thompson) labeled vortices in several spatio-temporal regions, i.e., for different spatial blocks in several time steps. Each individual detector was then calibrated by varying its threshold and selecting the value that produces the "best match" with the expert labels (high sensitivity and specificity; see below). *We chose 1000 samples for training (chosen such that the number of vortex and non-vortex samples were roughly balanced) and 3200 samples for validation from this collection of expert-labeled data and generated the compound classifier.* In the discussion below, we describe the performance of the compound classifier and the four physics-based classifiers using the following error measures:

ER	=	$(P_F+N_F)/(P+N)$	error rate
TP	=	$P_T/(P+N)$	true positive rate
FN	=	$N_F/(P+N)$	false negative rate
FP	=	$P_F/(P+N)$	false positive rate
TN	=	$N_T/(P+N)$	true negative rate
SPC	=	N_T/N	specificity
SEN	=	P_T/P	sensitivity

where P and N are the number of positive (vortex) and negative (non-vortex) expert labels, respectively, P_T and N_T are the number of correctly labeled positive and negative nodes, respectively, and P_F and N_F are the number of incorrectly labeled positive and negative nodes, respectively. It should be noted that these measures are defined here somewhat differently from other popularly used definitions. Typically, the denominator is taken to be the total number of true or negative samples; our definitions employ the total number of samples both positive and negative. Further, there is a degree of redundancy among the various measures.

Figure 3 illustrates the performance of the different classifiers in terms of these error measures. It is significant that the compound classifier shows improvement relative to the physics-based classifiers in almost every category and performs no worse in the remaining categories. In particular, the error rate decreases by approximately 50% while both the specificity and sensitivity are increased. The compound classifier is successful because it reduces both the false positive rate and the false negative rate.

We next investigated the relative performance of the classifiers on a block of data containing a single vortex by com-

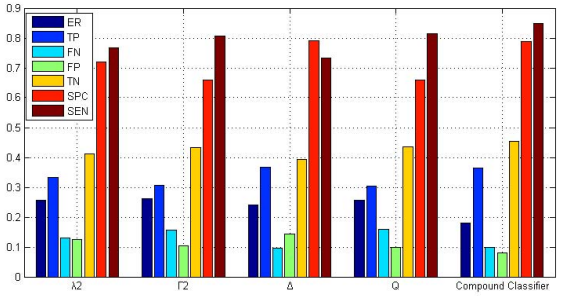


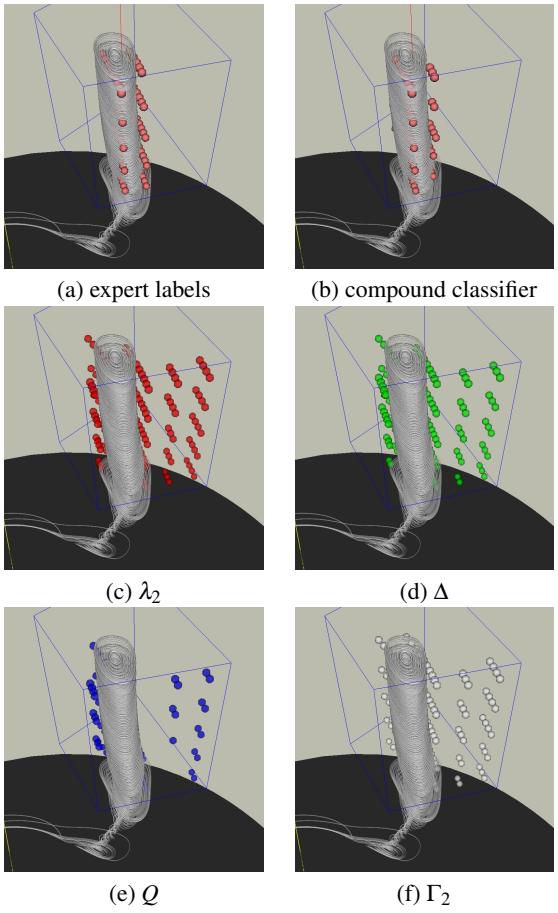
Figure 3: Tapered cylinder - *Multi-block comparison:* The results clearly show that integration of the feature detectors leads to a lower error rate and better classification performance.

paring the nodes selected by the various classifiers with the expert labels and streamlines generated using the procedure described in Section 2.1. In Figure 4, the spheres represent nodes that were marked as being contained within a vortex. Compared to the four weak, physics-based learners, the compound classifier does a much better job reproducing the expert labels. These results also illustrate how the compound classifier does a good job of reducing the instances of false positives. The false positives occur for the weak classifiers because of thresholding. Recall that a calibrated global threshold is employed for each physics-based weak classifier. In this particular block, the global threshold clearly overestimates the extent of the vortex. Figure 5 quantitatively illustrates this behavior. The error rate is decreased by approximately 40% primarily due to a reduction in the false positive rate.

These results demonstrate conclusively that the machine learning-based compound classifier reduced the misclassification of nodes in comparison to four of the more popular physics-based vortex detectors. Further, they demonstrate that the *accuracy* of the visualization, as measured by agreement with set of expert labels that is taken as ground truth, is also enhanced.

4.2. Rearward Facing Step

The second case considered is the unsteady, incompressible, turbulent flow over a rearward facing step, which is shown schematically in Figure 6. The flow enters the domain from the lower left in the positive x direction, encounters the step, and separates. The flow conditions were chosen to match the experimental data obtained by Driver and Seegmiller [DS85]. Details concerning the numerical simulation are reported in Alam, *et al.* [AWT]; two of the co-authors of this paper are also co-authors on the cited paper. Expert labels were again generated using the technique described in Section 2.1. This case was chosen because, unlike the ta-



(a) expert labels (b) compound classifier

(c) λ_2 (d) Δ

(e) Q (f) Γ_2

Figure 4: Tapered cylinder: The nodes marked by the compound classifier show good agreement with the expert labels relative to the calibrated physics-based classifiers.

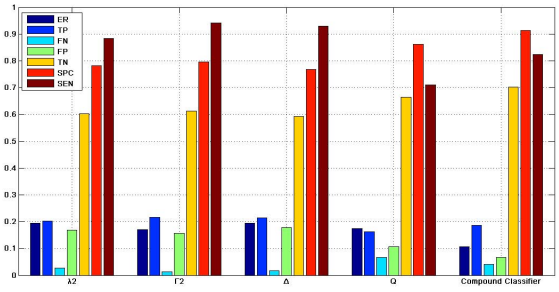


Figure 5: Tapered cylinder - *Single block comparison:* The results clearly show that integration of the feature detectors leads to lower error rates *with a reduced number of false positives.*

pered cylinder, the vortices do not have easily determined orientations and present more of a challenge to the labeling procedure.

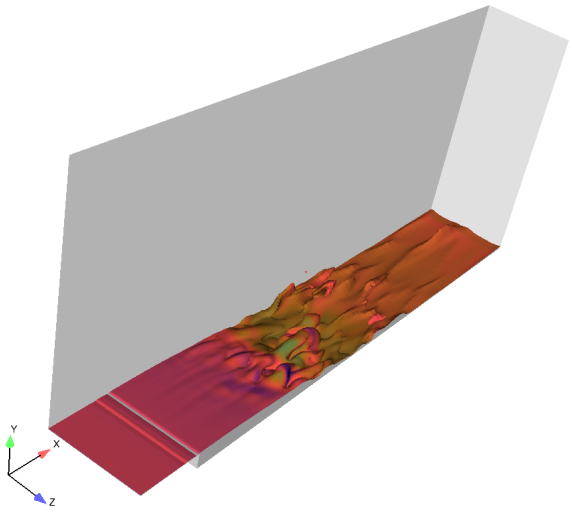


Figure 6: Schematic of computational domain for rearward facing step with an isosurface of instantaneous velocity shaded by instantaneous pressure.

Figure 7 illustrates the performance of the resulting compound classifier versus the physics-based classifiers across multiple blocks. There were 5000 training samples and 5000 validation samples. As before, the error rate was significantly decreased, again because of a reduction in the rate of false positives and the corresponding increase the number of true negatives. There is a slight reduction in the sensitivity in comparison with the results of λ_2; although it is mitigated by an increase in the specificity. We attribute this behavior to the choice of training data. Based on our experiences, we have concluded that the choice of training data plays a significant role in determining the performance of the algorithm. We plan to investigate this topic in more detail in our future efforts.

The efficacy of the compound classifier is demonstrated for a block containing a single vortex 8. Again, the nodes marked by the compound classifier show very good agreement with the expert labels. Figure 9 shows a comparison between classifiers for the single block data. The compound classifier shows a decrease in the false positive rate relative to the physics-based classifiers at the expense of a decrease in the true positive rate. This results in a decrease in the sensitivity, which is mitigated by an increase in the specificity because of the increase in the identification of true negatives. In general, these results exhibit the same trends as the multi-block evaluation shown in Figure 7 especially pertaining to the reduction of false positives.

Although the rearward facing step flow field is considerably more complex than the flow around the tapered cylin-

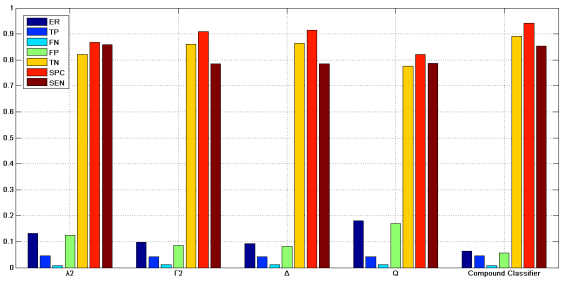
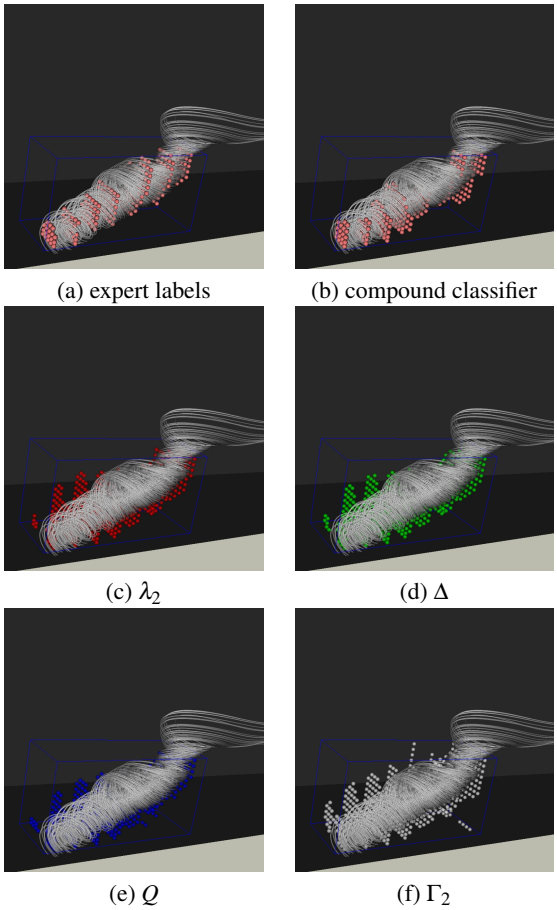


Figure 7: Rearward facing step: The results clearly show that integration of the feature detectors leads to a lower error rate and better classification performance (multi-block comparison).



(a) expert labels (b) compound classifier

(c) λ_2 (d) Δ

(e) Q (f) Γ_2

Figure 8: Rearward facing step: The nodes marked by the compound classifier show good agreement with the expert labels relative to the calibrated physics-based classifiers.

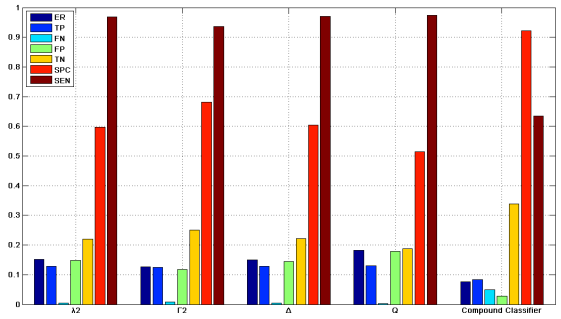


Figure 9: Rearward facing step: The results clearly show that integration of the feature detectors leads to a lower error rate and better classification performance (single-block comparison).

der, the results were qualitatively similar. In general, the compound classifier produces a lower error rate by reducing the number of false positives and false negatives; however, the number of true positives is also reduced.

4.3. Discussion

Given our experience with these two data sets, we assert that a statistical approach to feature detection will bear fruit. Although, the λ_2 classifier is often preferred, its performance is very dependent on the choice of threshold, which in fact should vary from block to block throughout the entire domain. The boosting framework that we espouse seeks to find a globally optimal operating point and chooses from a suite of weak physics-based classifiers. This classifier is conservative in nature in that it seeks to reduce the false positive rate at the expense of true positive rate thus often producing a lower sensitivity. False positives are often the bane of the weak classifiers as shear layers, etc., can also be identified as vortices. A working antidote for the poorer true positive rate of the compound classifier to seek a better balance in the training data between the number of positive and negative labels. We do not include a comprehensive study pertaining to the choice of this ratio in this work. However, we do duly note its influence on the performance of the compound classifier. A final point for discussion is that, since the compound classifier operated from a global perspective, some blocks will fare more poorly than others in terms of the true positive rates. However, on the average, the compound classifier delivers more *robust* performance by reducing the error rate across all blocks as indicated in Figures 3 and 7.

5. Conclusion

We presented a machine-learning based enhancement to vortex visualization techniques for complex flow fields. To our knowledge, this represents the first application of machine learning to feature detection in flow visualization. This algorithm combines several different vortex detection algorithms, which we term weak classifiers, using a semi-supervised, adaptive boosting algorithm (AdaBoost). Then, based on expert labeling, we computed a set of weights to be applied to each of the weak classifiers in order to produce a compound classifier. We used two computational fluid dynamics data sets, the tapered cylinder flow [DCJ91], which is characterized by vortices that are perpendicular to the primary flow, and the flow over a reward-facing step [AWT], which is characterized by a lack of preferential alignment of the vortices, for training and validating the compound classifier. In both cases, the compound classifier showed a quantitatively more accurate classification with with fewer misclassifications. In general, the compound classifier tends to be more conservative than the calibrated physics-based detectors. In most cases, the compound classifier produces fewer false positives and false negatives; however, there is generally a decrease in the number of true positives.

Not surprisingly, there are several opportunities for further research.

- **Using additional feature types**: We are currently using only four classifiers. We specifically remarked about the difficulty this issue poses to the construction of a very successful compound classifier. It is our hypothesis that the use of additional physically derived quantities will boost the performance of our classifier.
- **Selection of training data**: One issue we did not address extensively here is the composition of the training/testing data. There is no doubt that the manner in which this data is selected will influence the results as we remarked earlier. These effects need to quantified so that the performance of the compound classifier will be optimized.
- **Label variability due to multiple experts**: In the results reported here, a single expert (co-author Thompson) performed all of the vortex labeling. In the future, we want to investigate the performance of the compound classifier with respect to the labeling strategies of different experts.

References

[AWT] ALAM M., WALTERS D., THOMPSON D.: Evaluation of a dynamic hybird rans/les modeling methodology for attached and separated flows. *ASME Journal for Fluids Engineering (submitted)*. 9, 11

[BMI*07] BURGER R., MUIGG P., ILCIK M., DOLEISCH H., HAUSER H.: Integrating Local Feature Detectors in the Interactive Visual Analysis of Flow Simulation Data. In *Joint Eurographics–IEEE TCVG Sym. Visualization* (2007), pp. 171–178. 4

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 6 (2003), 1373–1396. 5

[BS94] BANKS D. C., SINGER B. A.: Vortex Tubes in Turbulent Flows: Identification, Representation and Reconstruction. In *IEEE Visualization '94* (1994), pp. 132–139. 3

[BS95] BANKS D. C., SINGER B. A.: A Predictor-Corrector

Technique for Visualizing Unsteady Flow. *IEEE Trans. Visualization and Computer Graphics 1*, 2 (1995), 151–163. 4

[BT93] BERDAHL C. H., THOMPSON D. S.: Eduction of Swirling Structure using the Velocity Gradient Tensor. *AIAA J. 31*, 1 (1993), 97–103. 3

[CBA05] CHAKRABORTY P., BALACHANDARAND S., ADRIAN R. J.: On the Relationships between Local Vortex Identification Schemes. *J. Fluid Mechanics 535* (2005), 189–214. 2, 4

[CPC90] CHONG M. S., PERRY A. E., CANTWELL B. J.: A General Classification of Three-Dimensional Flow Fields. *Physics of Fluids, A 2*, 5 (1990), 765–777. 4

[CQB99] CUCITORE R., QUADRIO M., BARON A.: On the Effectiveness and Limitations of Local Criteria for the Identification of a Vortex. *European J. Mechanics B/Fluids 18*, 2 (1999), 262–282. 2, 3, 4

[DCJ91] DENNIS C. JESPERSEN C. L.: Numerical Simulation of Flow Past a Tapered Cylinder. In *29th Aerospace Sciences Meeting* (1991). AIAA Paper 91-0751. 8, 11

[DD00] DUBIEF Y., DELCAYRE F.: On Coherent Vortex Identification in Turbulence. *J. Turbulence 1*, 1 (2000), 1–22. 2, 4

[DS85] DRIVER D., SEEGMILLER H.: Features of a reattaching turbulent shear layer in divergent channel flow. *AIAA journal 23* (1985), 163–171. 9

[FS97] FREUND Y., SCHAPIRE R.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 1 (1997), 119–139. 2, 7

[GGT07] GARTH C., GERHARDT F., TRICOCHE X.: Efficient Computation and Visualization of Coherent Structures in Fluid Flow Applications. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1464–1471. 3

[GMG01] GRAFTIEAUX L., MICHARD M., GROSJEAN N.: Combining PIV, POD and Vortex Identification Algorithms for the Study of Unsteady Turbulent Swirling Flows. *Measurement Science and Technology 12* (2001), 1422–1429. 4

[Hal01] HALLER G.: Distinguished Material Surface and Coherent Structures in Three-Dimensional Flows. *Physica D 149* (2001), 248–277. 3

[Hal05] HALLER G.: An Objective Definition of a Vortex. *J. Fluid Mech. 525* (2005), 1–26. 3

[HK99] HAIMES R., KENWRIGHT D. N.: On the Velocity Gradient Tensor and Fluid Feature Extraction. In *AIAA 14th Computational Fluid Dynamics Conf., Paper 99-3288* (1999). 2, 4

[HWM88] HUNT J., WRAY A., MOIN P.: *Eddies, Stream, and Convergence Zones in Turbulent Flows*. Tech. Rep. CTR-S88, Center for Turbulence Research, Stanford University, 1988. 4

[JH95] JEONG J., HUSSAIN F.: On the Identification of a Vortex. *J. Fluid Mechanics 285* (1995), 69–94. 4

[JMT02a] JIANG M., MACHIRAJU R., THOMPSON D. S.: A Novel Approach to Vortex Core Region Detection. In *Joint Eurographics–IEEE TCVG Sym. Visualization* (2002), pp. 217–225. 3

[JMT02b] JIANG M., MACHIRAJU R., THOMPSON D. S.: Geometric Verification of Swirling Features in Flow Fields. In *IEEE Visualization '02* (2002), pp. 307–314. 2, 4

[KM98] KIDA S., MIURA H.: Identification and Analysis of Vortical Structures. *European J. Mechanics B/Fluids 17*, 4 (1998), 471–488. 2, 4

[Kol07] KOLÁR V.: Vortex identification: New requirements and limitations. *International journal of heat and fluid flow 28*, 4 (2007), 638–652. 2

[LDS90] LEVY Y., DEGANI D., SEGINER A.: Graphical Visualization of Vortical Flows by Means of Helicity. *AIAA J. 28*, 8 (1990), 1347–1352. 3

[PR99] PEIKERT R., ROTH M.: The Parallel Vectors" Operator - A Vector Field Visualization Primitive. In *Proceedings of the 10th IEEE Visualization Conference (VIS '99)* (Washington, DC, USA, 1999), IEEE Computer Society, pp. 263–270. 3

[Rep09] *ExaScale Software Study: Software Challenges in Extreme Scale Systems*. Tech. rep., DARPA, 2009. 1

[Rob91] ROBINSON S. K.: Coherent Motions in the Turbulent Boundary Layer. *Ann. Rev. Fluid Mechanics 23* (1991), 601–639. 2

[RP98] ROTH M., PEIKERT R.: A Higher-Order Method for Finding Vortex Core Lines. In *IEEE Visualization '98* (1998), pp. 143–150. 3

[RSVP02] REINDERS F., SADARJOEN I. A., VROLIJK B., POST F. H.: Vortex Tracking and Visualization in a Flow Past a Tapered Cylinder. *Computer Graphics Forum 21* (2002), 675–682. 3

[SBV*11] SCHAFHITZEL T., BAYSAL K., VAARANIEMI M., RIST U., WEISKOPF D.: Visualizing the evolution and interaction of vortices and shear layers in time-dependent 3d flow. *Visualization and Computer Graphics, IEEE Transactions on 17*, 4 (2011), 412–425. 2

[SH95] SUJUDI D., HAIMES R.: Identification of Swirling Flow in 3D Vector Fields. In *AIAA 12th Computational Fluid Dynamics Conf., Paper 95-1715* (1995). 3

[SKA99] STRAWN R. C., KENWRIGHT D. N., AHMAD J.: Computer Visualization of Vortex Wake Systems. *AIAA J. 37*, 4 (1999), 511–512. 3

[SP07] SADLO F., PEIKERT R.: Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1456–1463. 3

[SR00] SAUL L., ROWEIS S.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science 290* (2000), 2323–2326. 5

[SRE05] STEGMAIER S., RIST U., ERTL T.: Opening the Can of Worms: An Exploration Tool for Vortical Flows. In *IEEE Visualization '05* (2005), pp. 463–470. 4

[TdL00] TENENBAUM J., DE SILVA V., LANGFORD J.: A global geometric framework for nonlinear dimensionality reduction. *Science 290* (2000), 2319–2323. 5

[TGK*04] TRICOCHE X., GARTH C., KINDLMANN G., DEINES E., SCHEUERMANN G., RUETTEN M., HANSEN C.: Visualization of Intricate Flow Structures for Vortex Breakdown Analysis. In *IEEE Visualization '04* (2004), pp. 187–194. 4

[TMJ*02] THOMPSON D. S., MACHIRAJU R., JIANG M., NAIR J., CRACIUN G., VENKATA S.: Physics-Based Feature Mining for Large Data Exploration. *IEEE Computing in Science & Engineering 4*, 4 (2002), 22–30. 1

[WM09] WU X., MOIN P.: Direct numerical simulation of turbulence in a nominally zero-presure-gradient flat-plate boundary layer. *J. Fluid Mechanics 630*, 1 (2009), 5–41. 2

[WSTH07] WEINKAUF T., SAHNER J., THEISEL H., HEGE H.-C.: Cores of Swirling Particle Motion in Unsteady Flows. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1759–1766. 3

[ZABK99] ZHOU J., ADRIAN R., BALACHANDAR S., KENDALL R.: Mechanisms for generating coherent packets of hairpin vortices in channel flow. *J. Fluid Mechanics 387* (1999), 363–396. 4